tiempo

April 29, 2025

[1]:
```
import requests
import datetime
import time
import json
import json
import csv
import pandas as pd
import numpy as np
import pprint
import seaborn as sns
import torch
import matplotlib.pyplot as plt
import torch.nn as nn

from dateutil.relativedelta import relativedelta
from sklearn.model_selection import train_test_split, TimeSeriesSplit,␣
 ↪RandomizedSearchCV
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from tqdm import tqdm
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, make_scorer
from sklearn.preprocessing import StandardScaler
from torch.utils.data import Dataset, DataLoader
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

[2]:
```
API_KEY = "eyJhbGciOiJIUzI1NiJ9.
 ↪eyJzdWIiOiJvc2N1cm9tZXRlb3JvQGdtYWlsLmNvbSIsImp0aSI6ImQ4YjQ3YmVjLWQ5MWMtNGRhMy1hNjNjLTRlZjI
 ↪w_x6_x8-lfdn9L3Hho-jOFgQBoZnqj4qumE5yBb_FGg"
download_data = False
```

0.1 Obtener los datos de la API de AEMET

[3]:
```
def fetch_interval_data(api_key, station, start_dt, end_dt):
    """
    For the given date range [start_dt, end_dt], build the URL,
    call AEMET's API, then use the "datos" URL to retrieve the actual
    meteorological data.
    """
```

```
    # Format dates as required by the API.
    start_str = start_dt.strftime("%Y-%m-%dT00:00:00UTC")
    end_str = end_dt.strftime("%Y-%m-%dT23:59:59UTC")

    url = (
        f"https://opendata.aemet.es/opendata/api/valores/climatologicos/diarios/
↪datos/"
        f"fechaini/{start_str}/fechafin/{end_str}/estacion/{station}/"
        f"?api_key={api_key}"
    )
    print(f"Requesting data from {start_str} to {end_str} ...")
    response = requests.get(url)
    if response.status_code != 200:
        print(
            f"Initial request FAILED for {start_str} - {end_str}. "
            f"Status code: {response.status_code}"
        )
        return []

    try:
        initial_data = response.json()
    except Exception as e:
        print("Error parsing the initial JSON response:", e)
        return []

    if initial_data.get("estado") != 200 or initial_data.get("descripcion") !=␣
↪"exito":
        print(
            f"API reported an error for {start_str} - {end_str}:",
            initial_data,
        )
        return []

    # The real meteorological data is available at the URL provided
    # in the "datos" key.
    datos_url = initial_data.get("datos")
    if not datos_url:
        print("No 'datos' key found in the API response.")
        return []

    datos_response = requests.get(datos_url)
    if datos_response.status_code != 200:
        print(
            f"Request for the real data FAILED for {start_str} - {end_str}. "
            f"Status code: {datos_response.status_code}"
        )
        return []
```

```
    try:
        real_data = datos_response.json()
    except Exception as e:
        print("Error parsing the real data JSON:", e)
        return []

    return real_data
```

[4]:
```
output_file = "weather-00-25.json"
if download_data:
    # Obten esto del csv "diccionario24.csv"
    station = "3170Y"

    # Define la fecha de inicio y la fecha final de los datos que vamos a␣
 ↪recolectar para entrenar los modelos.
    # Se ha escogido la fecha de 2009 porque es la primera en la que la API␣
 ↪devuelve datos.
    global_start = datetime.datetime(2009, 1, 1)
    global_end = datetime.datetime(2025, 4, 28)
    merged_data = []

    current_date = global_start
    while current_date <= global_end:
        potential_end = current_date + relativedelta(months=+6) - datetime.
 ↪timedelta(days=1)
        segment_end = min(potential_end, global_end)

        interval_data = fetch_interval_data(API_KEY, station, current_date,␣
 ↪segment_end)
        if interval_data:
            if isinstance(interval_data, list):
                merged_data.extend(interval_data)
            else:
                merged_data.append(interval_data)
        else:
            print(
                f"No data retrieved for the period {current_date.date()} to "
                f"{segment_end.date()}."
            )

        current_date = current_date + relativedelta(months=+6)

        # Pausa para eliminar los errores de la API "too many requests"
        time.sleep(1)

    # Escribimos los datos en un archivo JSON
```

```
    try:
        with open(output_file, "w", encoding="utf-8") as f:
            json.dump(merged_data, f, indent=2, ensure_ascii=False)
        print(f"Data merged and saved to {output_file}.")
    except Exception as e:
        print("Error writing merged data to file:", e)
```

0.2 Convertir los datos de JSON a CSV

[5]:
```
# Input JSON file
with open(output_file, 'r', encoding='utf-8') as json_file:
    input_json = json.load(json_file)

# Output CSV file
output_csv = output_file.replace('.json', '.csv')

# Get all possible fieldnames from all dictionaries in the JSON
all_fieldnames = set()
for item in input_json:
    all_fieldnames.update(item.keys())

# Convert JSON to CSV
with open(output_csv, mode='w', newline='', encoding='utf-8') as csv_file:
    writer = csv.DictWriter(
        csv_file,
        fieldnames=list(all_fieldnames),
        restval=-999,  # Default value for missing fields
        extrasaction='ignore',  # Ignore extra fields not in fieldnames
        delimiter=';'  # Use semicolon as separator,
    )
    writer.writeheader()
    writer.writerows(input_json)

print(f"JSON data has been successfully converted to {output_csv}")
```

```
JSON data has been successfully converted to weather-00-25.csv
```

[6]:
```
# Replace all the , with . in the CSV file
with open(output_csv, 'r', encoding='utf-8') as file:
    data = file.read()
data = data.replace(',', '.')
data = data.replace(';', ',')
with open(output_csv, 'w', encoding='utf-8') as file:
    file.write(data)
```

0.3 Comprobar las fechas que faltan en los datos

[7]:
```
# Carga el CSV
data = pd.read_csv(output_csv)

data['fecha'] = pd.to_datetime(data['fecha'], errors='coerce')
data = data.sort_values(by='fecha')
data['gap'] = data['fecha'].diff().dt.days
gaps = data[data['gap'] > 1]

if not gaps.empty:
    print("Los huecis enncontrados en estos datos son:")
    print(f"La suma de los días en los que no hay datos es de: {int(gaps['gap'].
 ↪sum())} días")
    print(gaps[['fecha', 'gap']])
else:
    print("No gaps found in the 'fecha' field.")
```

```
Los huecis enncontrados en estos datos son:
La suma de los días en los que no hay datos es de: 734 días
          fecha    gap
65   2009-04-10    5.0
119  2009-06-04    2.0
224  2009-09-18    2.0
227  2009-09-22    2.0
270  2009-11-05    2.0
273  2009-11-09    2.0
408  2010-04-14   22.0
498  2010-07-15    3.0
532  2010-08-19    2.0
537  2010-08-28    5.0
569  2010-09-30    2.0
673  2011-01-19    8.0
803  2011-06-09   12.0
1375 2013-07-01  182.0
1433 2013-09-03    7.0
1759 2014-07-27    2.0
1761 2014-07-31    3.0
1826 2014-10-05    2.0
1827 2014-10-07    2.0
1855 2014-11-05    2.0
1871 2014-11-22    2.0
1967 2015-02-27    2.0
1999 2015-04-01    2.0
2057 2015-06-11   14.0
2167 2015-09-30    2.0
2172 2015-10-06    2.0
2185 2015-10-20    2.0
```

```
2194 2015-11-03    6.0
2198 2015-11-09    3.0
2200 2015-11-12    2.0
2201 2016-03-30  139.0
2263 2016-06-16   17.0
2319 2016-08-25   15.0
2399 2016-12-15   33.0
2400 2016-12-17    2.0
2401 2016-12-19    2.0
2408 2016-12-28    3.0
2409 2016-12-31    3.0
2420 2017-01-12    2.0
2436 2017-02-10   14.0
2477 2017-05-11   50.0
2488 2017-07-27   67.0
2501 2017-08-25   17.0
2512 2017-10-25   51.0
3203 2019-09-19    4.0
3220 2019-10-07    2.0
3221 2019-10-09    2.0
3304 2020-01-01    2.0
3597 2020-10-22    3.0
```

0.4 Transformar los datos y limpieza de los mismos

[8]:
```
data = pd.read_csv(output_csv)

all_fields = data.columns.tolist()
print(f"Total number of fields: {len(all_fields)}")
pprint.pprint(all_fields, compact=True)

print(f"Número totoal de filas: {len(data)} antes de eliminar las filas en las␣
 ↪que faltan algunos datos:")
data = data[~(data == -999).any(axis=1)]
data = data[~(data == "-999").any(axis=1)]
print(f"Número de filas que quedan después de eliminar las filas que contienen␣
 ↪un -999: {len(data)}")
data = data.dropna()
print(f"Número de filas que quedan después de eliminar las filas que contienen␣
 ↪un NaN: {len(data)}")

data = data.drop(columns=["nombre", "provincia", "indicativo", "altitud"],␣
 ↪axis=1)

# data = data[~(data == "Varias").any(axis=1)]
# print(f"Numeros de filas que quedan después de eliminar las filas que␣
 ↪contienen 'Varias': {len(data)}")
```

```
# Convert 'fecha' to datetime if not already
data['fecha'] = pd.to_datetime(data['fecha'], errors='coerce')

data['day'] = data['fecha'].dt.dayofyear
```

```
Total number of fields: 20
['horaHrMin', 'velmedia', 'horatmin', 'hrMedia', 'horaracha', 'dir', 'racha',
 'indicativo', 'horaHrMax', 'prec', 'horatmax', 'hrMax', 'tmed', 'tmax',
 'hrMin', 'nombre', 'tmin', 'provincia', 'fecha', 'altitud']
Número totoal de filas: 5245 antes de eliminar las filas en las que faltan
algunos datos:
Número de filas que quedan después de eliminar las filas que contienen un -999:
5090
Número de filas que quedan después de eliminar las filas que contienen un NaN:
5090
```

[9]:
```
# elimina las columnas de tiempos, ya que solo nos interesa la temperatura␣
 ↪media, alta y baja del día, no la hora en la que se ha registrado esa␣
 ↪temperatura

data = data.drop(columns=['horatmin', 'horatmax', 'horaHrMax', 'horaracha',␣
 ↪'horaHrMin'], axis=1)
```

0.5 Generar los datos de los últimos N días

[10]:
```
def derive_nth_day_feature(df: pd.DataFrame, feature: str, N: int):
    col_name = f"{feature}_{N}"
    df[col_name] = df[feature].shift(N)
    return df

features = data.columns.tolist()
features.remove('fecha')
features.remove('day')

# Crear nuevas columnas para las temperaturas de los días anteriores
for feature in features:
    for N in range(1, 4):
        data = derive_nth_day_feature(data, feature, N)

# Filtra las columnas con datos del mismo día
# Mentener solo las columnas que terminen con _1, _2, _3
columns_to_keep = []
for col in data.columns:
    if '_' in col or col == 'fecha' or col == 'day' or col == 'tmed':
        columns_to_keep.append(col)
    elif any(col.endswith(f'_{i}') for i in range(1, 4)):
        columns_to_keep.append(col)
```

```
data = data[columns_to_keep]

data = data.dropna()

print(f"Final shape of dataset: {data.shape}")
print("\nColumns in final dataset:")
print(data.columns.tolist())
```

```
Final shape of dataset: (5087, 33)

Columns in final dataset:
['tmed', 'fecha', 'day', 'velmedia_1', 'velmedia_2', 'velmedia_3', 'hrMedia_1',
'hrMedia_2', 'hrMedia_3', 'dir_1', 'dir_2', 'dir_3', 'racha_1', 'racha_2',
'racha_3', 'prec_1', 'prec_2', 'prec_3', 'hrMax_1', 'hrMax_2', 'hrMax_3',
'tmed_1', 'tmed_2', 'tmed_3', 'tmax_1', 'tmax_2', 'tmax_3', 'hrMin_1',
'hrMin_2', 'hrMin_3', 'tmin_1', 'tmin_2', 'tmin_3']
```

[11]:
```
# Guarda el CSV limpio
output_clean_csv = output_file.replace('.json', '-clean.csv')
data.to_csv(output_clean_csv, index=False)
print(f"Cleaned data has been saved to {output_clean_csv}")
```

```
Cleaned data has been saved to weather-00-25-clean.csv
```

1 Comprobar cuales son los parámetros importantes

Mirar que columnas no tienen una correlación alta entre sí

[12]:
```
data = pd.read_csv(output_clean_csv, parse_dates=['fecha'], dayfirst=True)
data['fecha'] = pd.to_datetime(data['fecha'], format='%Y-%m-%d',␣
 ↪errors='coerce')
data = data.dropna(subset=['fecha'])
data = data.sort_values(by='fecha')

# Convertir la columna 'fecha' a un índice de fecha
data.set_index('fecha', inplace=True)
data = data.asfreq('D')


# Convertir las columnas tmin, racha  tmax velmedia altitud prec tmed a float
columns_to_convert = data.columns.tolist()
for col in columns_to_convert:
    data[col] = pd.to_numeric(data[col], errors='coerce')

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5926 entries, 2009-02-04 to 2025-04-26
```

```
Freq: D
Data columns (total 32 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   tmed        5087 non-null   float64
 1   day         5087 non-null   float64
 2   velmedia_1  5087 non-null   float64
 3   velmedia_2  5087 non-null   float64
 4   velmedia_3  5087 non-null   float64
 5   hrMedia_1   5087 non-null   float64
 6   hrMedia_2   5087 non-null   float64
 7   hrMedia_3   5087 non-null   float64
 8   dir_1       5087 non-null   float64
 9   dir_2       5087 non-null   float64
 10  dir_3       5087 non-null   float64
 11  racha_1     5087 non-null   float64
 12  racha_2     5087 non-null   float64
 13  racha_3     5087 non-null   float64
 14  prec_1      5087 non-null   float64
 15  prec_2      5087 non-null   float64
 16  prec_3      5087 non-null   float64
 17  hrMax_1     5087 non-null   float64
 18  hrMax_2     5087 non-null   float64
 19  hrMax_3     5087 non-null   float64
 20  tmed_1      5087 non-null   float64
 21  tmed_2      5087 non-null   float64
 22  tmed_3      5087 non-null   float64
 23  tmax_1      5087 non-null   float64
 24  tmax_2      5087 non-null   float64
 25  tmax_3      5087 non-null   float64
 26  hrMin_1     5087 non-null   float64
 27  hrMin_2     5087 non-null   float64
 28  hrMin_3     5087 non-null   float64
 29  tmin_1      5087 non-null   float64
 30  tmin_2      5087 non-null   float64
 31  tmin_3      5087 non-null   float64
dtypes: float64(32)
memory usage: 1.5 MB
```

[13]:
```
data.corr()[['tmed']].abs().sort_values('tmed')
```

[13]:

	tmed
dir_1	0.000140
dir_2	0.005786
dir_3	0.009321
prec_3	0.091201
prec_2	0.092970

```
prec_1      0.093921
velmedia_1  0.147602
velmedia_2  0.153739
racha_1     0.176867
racha_2     0.177169
velmedia_3  0.181227
racha_3     0.202118
day         0.207439
hrMin_1     0.627331
hrMin_3     0.631258
hrMin_2     0.631863
hrMax_3     0.679317
hrMax_2     0.687778
hrMax_1     0.702521
hrMedia_3   0.707277
hrMedia_2   0.709740
hrMedia_1   0.711071
tmin_3      0.846604
tmin_2      0.866324
tmax_3      0.892796
tmin_1      0.902454
tmed_3      0.907520
tmax_2      0.916948
tmed_2      0.930634
tmax_1      0.948360
tmed_1      0.965384
tmed        1.000000
```

[14]:
```python
# Seleccionar las columnas que tienen una correlación mayor a 0.5 con tmed
correlated_columns = data.corr()[['tmed']].abs().sort_values('tmed',␣
 ↪ascending=False)
correlated_columns = correlated_columns[correlated_columns['tmed'] > 0.5].index.
 ↪tolist()
correlated_columns.remove('tmed')  # Remove 'tmed' itself from the list
print(f"Las columnas que tienen una correlación mayor a 0.5 con tmed son:␣
 ↪{correlated_columns}")
print()
non_correlated_columns = data.corr()[['tmed']].abs().sort_values('tmed',␣
 ↪ascending=False)
non_correlated_columns = non_correlated_columns[non_correlated_columns['tmed']␣
 ↪< 0.5].index.tolist()
print(f"Las columnas que tienen una correlación menor a 0.5 con tmed son:␣
 ↪{non_correlated_columns}")
```

```
Las columnas que tienen una correlación mayor a 0.5 con tmed son: ['tmed_1',
'tmax_1', 'tmed_2', 'tmax_2', 'tmed_3', 'tmin_1', 'tmax_3', 'tmin_2', 'tmin_3',
'hrMedia_1', 'hrMedia_2', 'hrMedia_3', 'hrMax_1', 'hrMax_2', 'hrMax_3',
```

```
'hrMin_2', 'hrMin_3', 'hrMin_1']

Las columnas que tienen una correlación menor a 0.5 con tmed son: ['day',
'racha_3', 'velmedia_3', 'racha_2', 'racha_1', 'velmedia_2', 'velmedia_1',
'prec_1', 'prec_2', 'prec_3', 'dir_3', 'dir_2', 'dir_1']
```

De los datos obtenidos de hacer la correlación con la temperatura media del día, podemos ver que los parámetros que menos influyen en esta son: - dir - day - year - prec - velmedia - month - racha

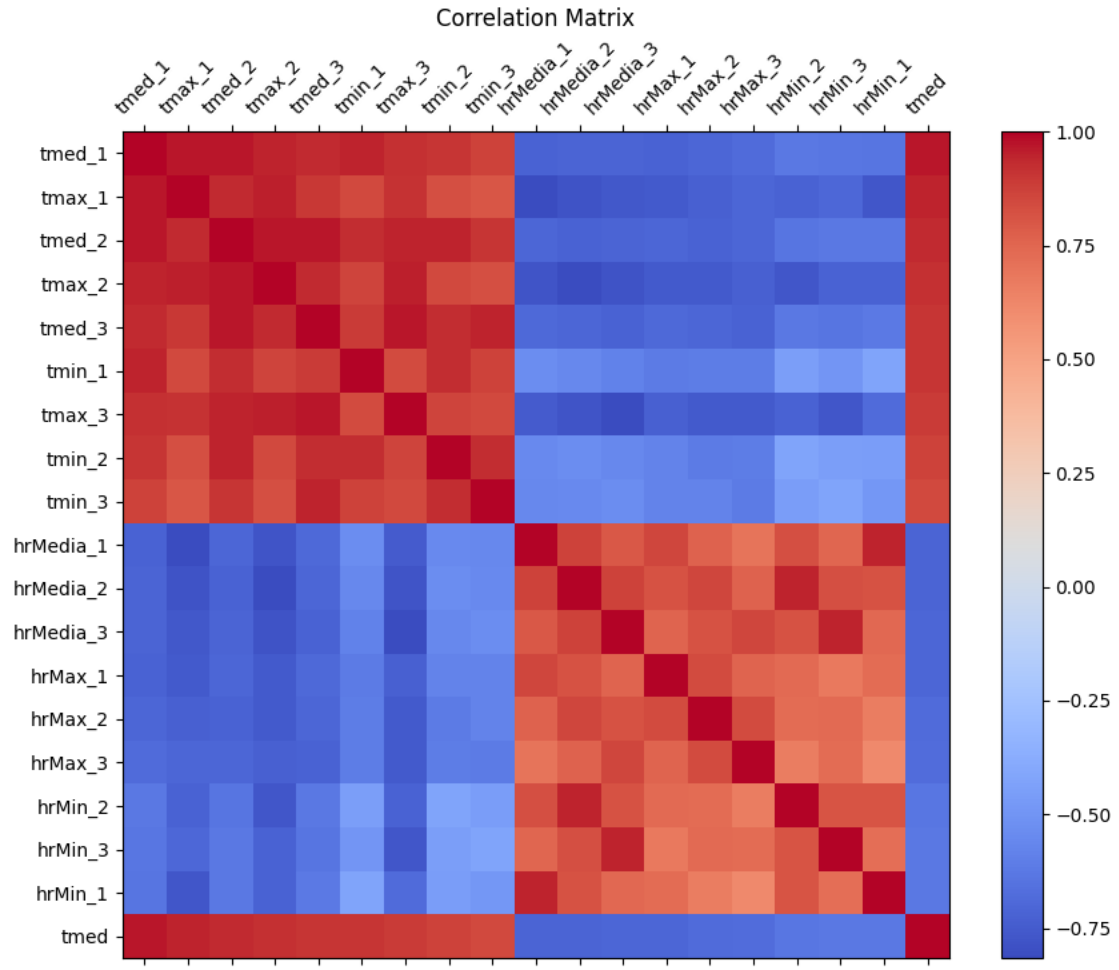
Por eso, descartaremos estos datos a la hora de entrenar nuestro modelo de predicción.

Por decirlo de otra manera, los parámetros que más influyen en la temperatura media del día son: - hrMin - hrMax - hrMedia - tmin - tmax

[15]:
```python
# Obtener los datos de las columnas seleccionadas
data_filtered = data[correlated_columns + ['tmed']].copy()
data_filtered = data_filtered.dropna()

# Guardar los datos filtrados en un nuevo CSV
output_filtered_csv = output_file.replace('.json', '-filtered.csv')
data_filtered.to_csv(output_filtered_csv, index=True)
print(f"Filtered data has been saved to {output_filtered_csv}")
```

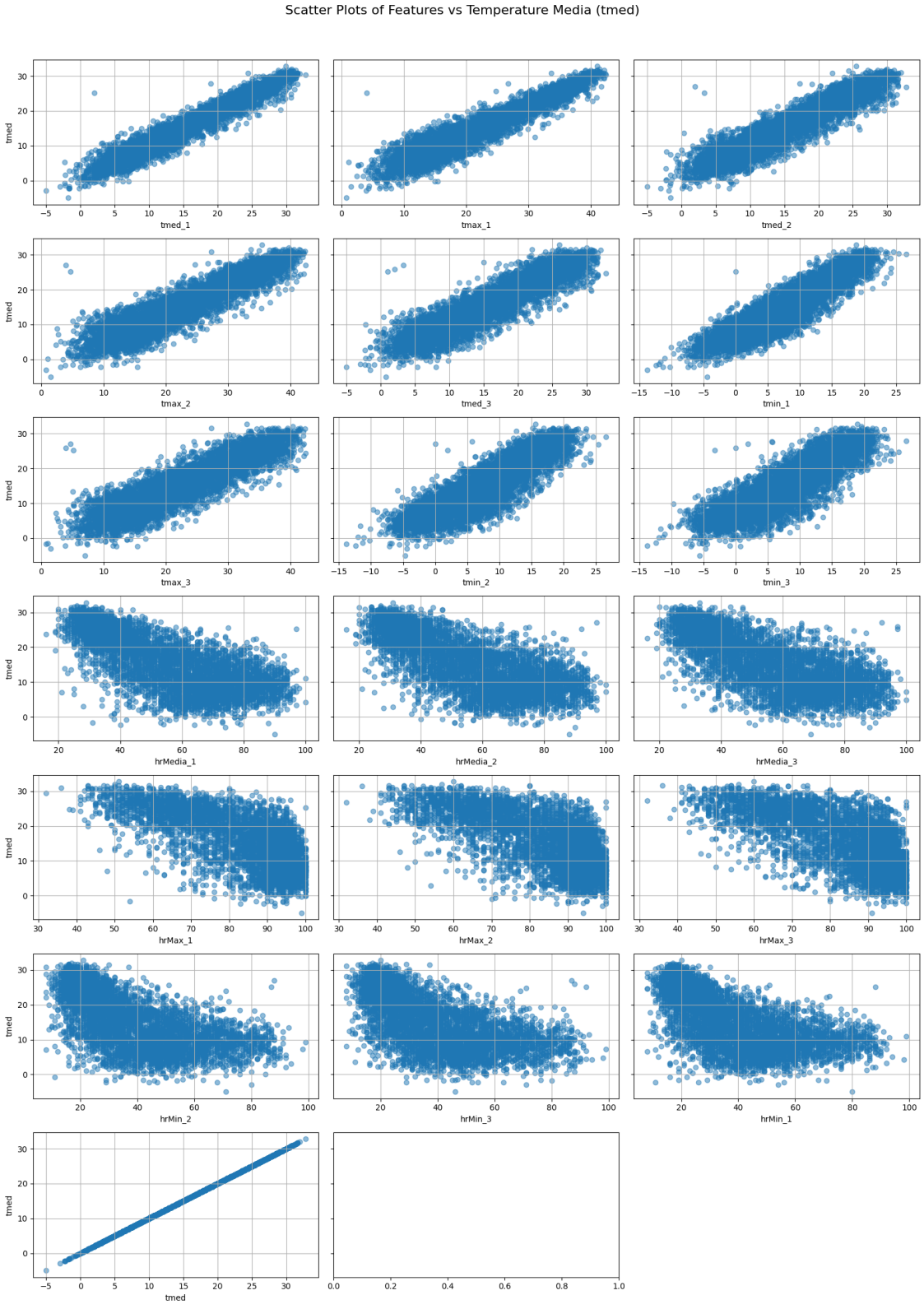
```
Filtered data has been saved to weather-00-25-filtered.csv
```

[16]:
```python
# Visualizar las relaciones entre las variables

def plot_correlation_matrix(data):
    """
    Plots the correlation matrix of the given DataFrame.
    """
    plt.figure(figsize=(10, 8))
    corr = data.corr()
    plt.matshow(corr, cmap='coolwarm', fignum=1)
    plt.colorbar()
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=45)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.title('Correlation Matrix')
    plt.show()
plot_correlation_matrix(data_filtered)
```



[17]:
```
%matplotlib inline

# Set figure size
plt.rcParams['figure.figsize'] = [16, 22]

# Create subplots
fig, axes = plt.subplots(nrows=7, ncols=3, sharey=True)

# Features to plot against tmed (target variable)
features = data_filtered.columns.tolist()

# Flatten axes for easier iteration
axes_flat = axes.flatten()

# Create scatter plots
for idx, feature in enumerate(features):
```

```
    axes_flat[idx].scatter(data[feature], data['tmed'], alpha=0.5)
    axes_flat[idx].set_xlabel(feature)
    axes_flat[idx].set_ylabel('tmed' if idx % 3 == 0 else '')
    axes_flat[idx].grid(True)

# Remove the last empty subplot
if len(features) < len(axes_flat):
    fig.delaxes(axes_flat[-1])

plt.suptitle('Scatter Plots of Features vs Temperature Media (tmed)', y=1.02,␣
 ↪fontsize=16)
plt.tight_layout()
plt.show()
```


Scatter Plots of Features vs Temperature Media (tmed)
tmed
tmed_1
tmax_1
tmed_2
tmax_2
tmed_3
tmin_1
tmax_3
tmin_2
tmin_3
hrMedia_1
hrMedia_2
hrMedia_3
hrMax_1
hrMax_2
hrMax_3
hrMin_2
hrMin_3
hrMin_1

2 Entrenar el modelo

[18]:
```
# Lee el fichero CSV con los datos ya limpios, para entrenar el modelo
data = pd.read_csv(output_filtered_csv, parse_dates=['fecha'],␣
 ↪index_col='fecha')
```

[19]:
```
data.columns
```

[19]:
```
Index(['tmed_1', 'tmax_1', 'tmed_2', 'tmax_2', 'tmed_3', 'tmin_1', 'tmax_3',
       'tmin_2', 'tmin_3', 'hrMedia_1', 'hrMedia_2', 'hrMedia_3', 'hrMax_1',
       'hrMax_2', 'hrMax_3', 'hrMin_2', 'hrMin_3', 'hrMin_1', 'tmed'],
      dtype='object')
```

2.1 Dividir los datos en train, test y val

[20]:
```
# Crea los datasets de entrenamiento, validación y test
X = data.drop('tmed', axis=1)
y = data['tmed']

# Divide los datos en conjuntos de entrenamiento y test
X_temp, X_test, y_temp, y_test = train_test_split(X, y, test_size=0.15,␣
 ↪shuffle=False)

# Divide los datos de entrenamiento en conjuntos de entrenamiento y validación
X_train, X_val, y_train, y_val = train_test_split(X_temp, y_temp, test_size=0.
 ↪165, shuffle=False)

print(f"Training set size: {len(X_train)}")
print(f"Validation set size: {len(X_val)}")
print(f"Test set size: {len(X_test)}")
```

```
Training set size: 3609
Validation set size: 714
Test set size: 764
```

2.2 Entrenar el modelo

[21]:
```
import warnings
warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")
warnings.filterwarnings("ignore", category=RuntimeWarning, module="sklearn")
warnings.filterwarnings("ignore", category=RuntimeWarning, module="scipy")


def train_with_hyperparameter_optimization(
    X_train, y_train, X_val, y_val, X_test, y_test
):
    # Como es una serie temporal, no se puede hacer shuffle
    # Se usa TimeSeriesSplit para hacer la validación cruzada
```

```
    tscv = TimeSeriesSplit(n_splits=5)

    # Escala las features
    scaler = StandardScaler()
    X_train_scaled = pd.DataFrame(
        scaler.fit_transform(X_train), columns=X_train.columns
    )
    X_val_scaled = pd.DataFrame(scaler.transform(X_val), columns=X_train.
↪columns)
    X_test_scaled = pd.DataFrame(scaler.transform(X_test), columns=X_train.
↪columns)

    # Define el modelo y sus hiperparámetros
    model_config = {
        "LinearRegression": {
            "model": LinearRegression(),
            "params": {
                "fit_intercept": [True, False],
                "copy_X": [True],
                "positive": [False],
                "n_jobs": [None, -1],
            },
        }
    }

    # Puntuador personalizado de RMSE (negativo para optimización)
    rmse_scorer = make_scorer(
        lambda y, y_pred: -np.sqrt(mean_squared_error(y, y_pred))
    )

    results = {}

    # Entrena y evalua el modelo
    for model_name, model_info in model_config.items():
        with tqdm(total=1, desc=f"Training {model_name}") as pbar:
            # Haz una búsqueda Cross Validation aleatorea para reducir el␣
↪tiempo de entrenamiento
            search = RandomizedSearchCV(
                estimator=model_info["model"],
                param_distributions=model_info["params"],
                n_iter=100,
                cv=tscv,
                scoring=rmse_scorer,
                n_jobs=-1,
                random_state=42,
                verbose=0,
            )
```

```
            # Fit model
            search.fit(X_train_scaled, y_train)
            pbar.update(1)

            # Obten el moejor modelo de la búsqueda
            # y haz predicciones en los conjuntos de entrenamiento, validación
 ↪y test
            best_model = search.best_estimator_
            train_pred = best_model.predict(X_train_scaled)
            val_pred = best_model.predict(X_val_scaled)
            test_pred = best_model.predict(X_test_scaled)

            # Guarda los resultados
            results[model_name] = {
                "Train RMSE": np.sqrt(mean_squared_error(y_train, train_pred)),
                "Val RMSE": np.sqrt(mean_squared_error(y_val, val_pred)),
                "Test RMSE": np.sqrt(mean_squared_error(y_test, test_pred)),
                "Best Parameters": search.best_params_,
                "Coefficients": dict(zip(X_train.columns, best_model.coef_)),
                "Intercept": best_model.intercept_,
                "Model": best_model,
            }

    return results, scaler
```

[22]:
```
# Train the model
results, scaler = train_with_hyperparameter_optimization(
    X_train, y_train, X_val, y_val, X_test, y_test
)
```

```
Training LinearRegression:   0%|          | 0/1 [00:00<?,
?it/s]/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
```

```
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
```

```
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
```

```
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
```

```
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
```

```
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: divide by zero
encountered in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: overflow encountered
in matmul
  return X @ coef_ + self.intercept_
/Users/edu/Documents/Personal/juan-tiempo/.venv/lib/python3.12/site-
packages/sklearn/linear_model/_base.py:279: RuntimeWarning: invalid value
encountered in matmul
  return X @ coef_ + self.intercept_
Training LinearRegression: 100%|      | 1/1 [00:02<00:00,  2.22s/it]
```

[23]:
```python
# Print results
for model_name, metrics in results.items():
    print(f"\nResults for {model_name}:")
    print(f"Train RMSE: {metrics['Train RMSE']:.4f}")
    print(f"Validation RMSE: {metrics['Val RMSE']:.4f}")
    print(f"Test RMSE: {metrics['Test RMSE']:.4f}")
    print("\nBest Parameters:")
    print(metrics['Best Parameters'])
```

```
Results for LinearRegression:
Train RMSE: 1.8734
Validation RMSE: 1.8251
Test RMSE: 1.8672
```

```
Best Parameters:
{'positive': False, 'n_jobs': None, 'fit_intercept': True, 'copy_X': True}
```

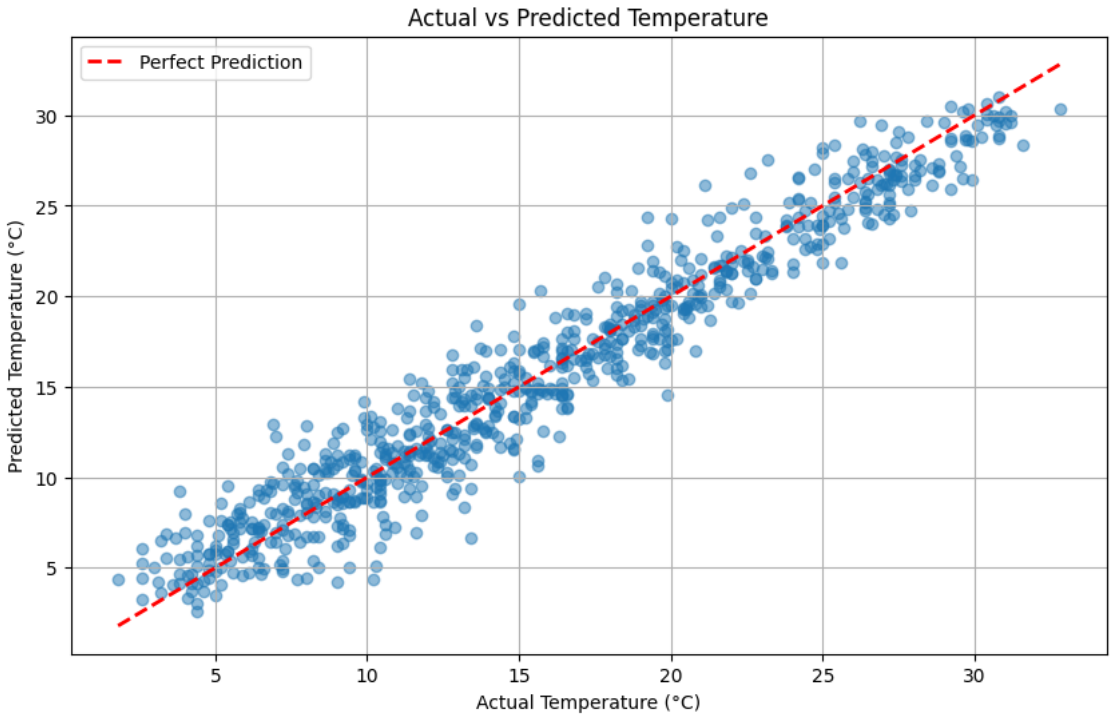
2.3 Evaluar el modelo

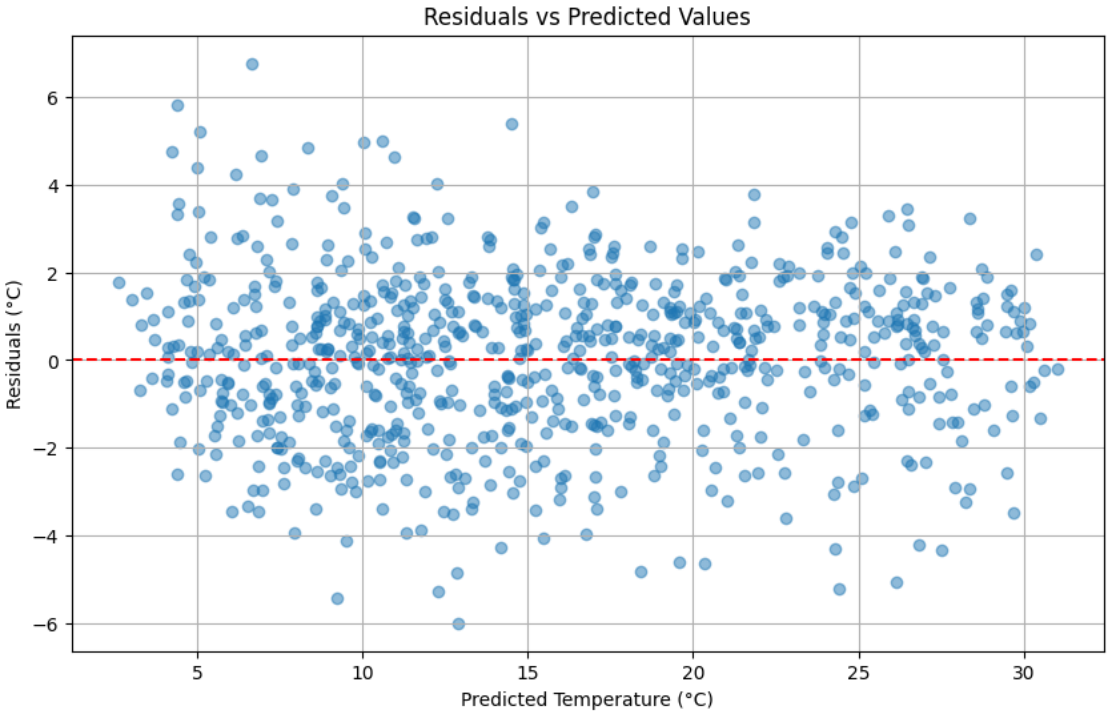
[24]:
```python
def evaluate_model(results, X_test, y_test, scaler):
    # Obten el mejor modelo de los entrenados.
    model = results['LinearRegression']['Model']

    # Haz predicciones en el conjunto de test
    X_test_scaled = pd.DataFrame(scaler.transform(X_test), columns=X_test.
 ↪columns)
    y_pred = model.predict(X_test_scaled)

    # Calcula las métricas de rendimiento
    r2 = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100

    # Imprime las metricas
    print("\nModel Performance Metrics:")
    print(f"R² Score: {r2:.4f}")
    print(f"Mean Absolute Error: {mae:.4f}°C")
    print(f"Root Mean Squared Error: {rmse:.4f}°C")
    print(f"Mean Absolute Percentage Error: {mape:.2f}%")

    # 1. Actual vs predicción
    plt.figure(figsize=(10, 6))
    plt.scatter(y_test, y_pred, alpha=0.5)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
             'r--', lw=2, label='Perfect Prediction')
    plt.xlabel('Actual Temperature (°C)')
    plt.ylabel('Predicted Temperature (°C)')
    plt.title('Actual vs Predicted Temperature')
    plt.legend()
    plt.grid(True)
    plt.show()

    # 2. Plot de los Residuales
    residuals = y_test - y_pred
    plt.figure(figsize=(10, 6))
    plt.scatter(y_pred, residuals, alpha=0.5)
    plt.axhline(y=0, color='r', linestyle='--')
    plt.xlabel('Predicted Temperature (°C)')
    plt.ylabel('Residuals (°C)')
```

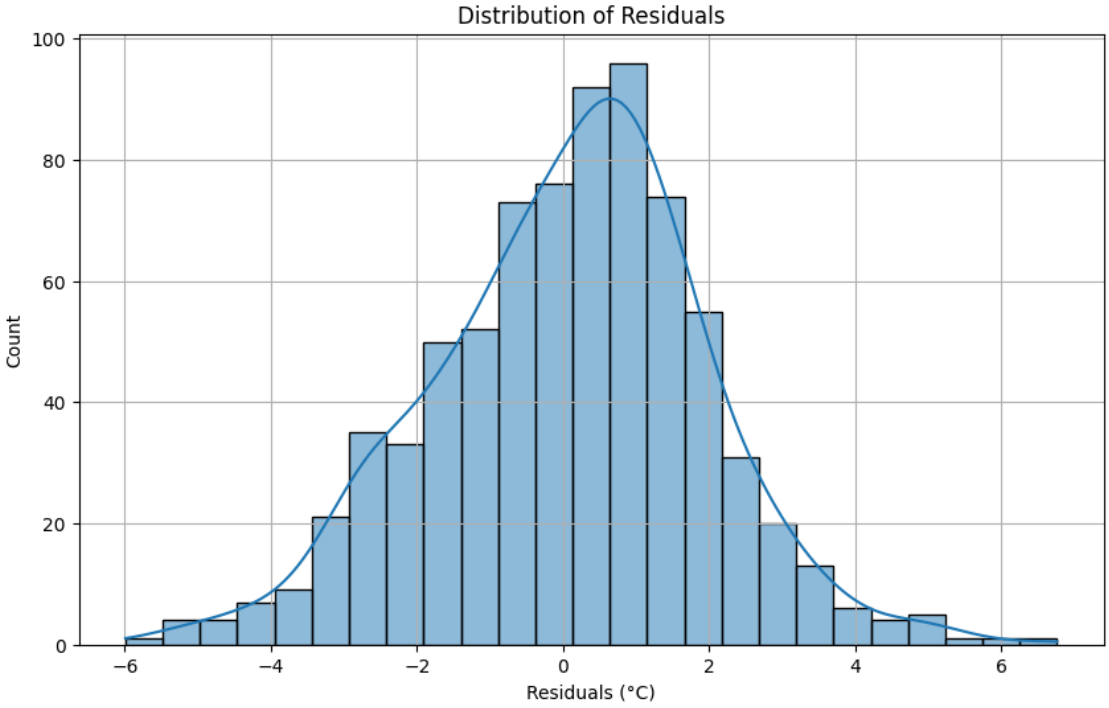
```
    plt.title('Residuals vs Predicted Values')
    plt.grid(True)
    plt.show()

    # 3. Distribution de Residuales
    plt.figure(figsize=(10, 6))
    sns.histplot(residuals, kde=True)
    plt.xlabel('Residuals (°C)')
    plt.ylabel('Count')
    plt.title('Distribution of Residuals')
    plt.grid(True)
    plt.show()

    # 4. Gráfico de Importancia de las Características
    coefficients = results['LinearRegression']['Coefficients']
    coef_df = pd.DataFrame({
        'Feature': coefficients.keys(),
        'Coefficient': np.abs(list(coefficients.values()))
    }).sort_values('Coefficient', ascending=True)

    plt.figure(figsize=(10, 6))
    plt.barh(coef_df['Feature'], coef_df['Coefficient'])
    plt.xlabel('Absolute Coefficient Value')
    plt.title('Feature Importance')
    plt.grid(True)
    plt.show()

    # 5. Gráfico de Serie Temporal de Predicciones
    if isinstance(X_test.index, pd.DatetimeIndex):
        plt.figure(figsize=(15, 6))
        plt.plot(X_test.index, y_test, label='Actual', alpha=0.7)
        plt.plot(X_test.index, y_pred, label='Predicted', alpha=0.7)
        plt.xlabel('Date')
        plt.ylabel('Temperature (°C)')
        plt.title('Actual vs Predicted Temperature Over Time')
        plt.legend()
        plt.grid(True)
        plt.show()

    return {
        'R2': r2,
        'MAE': mae,
        'RMSE': rmse,
        'MAPE': mape,
        'Residuals': residuals
    }
```

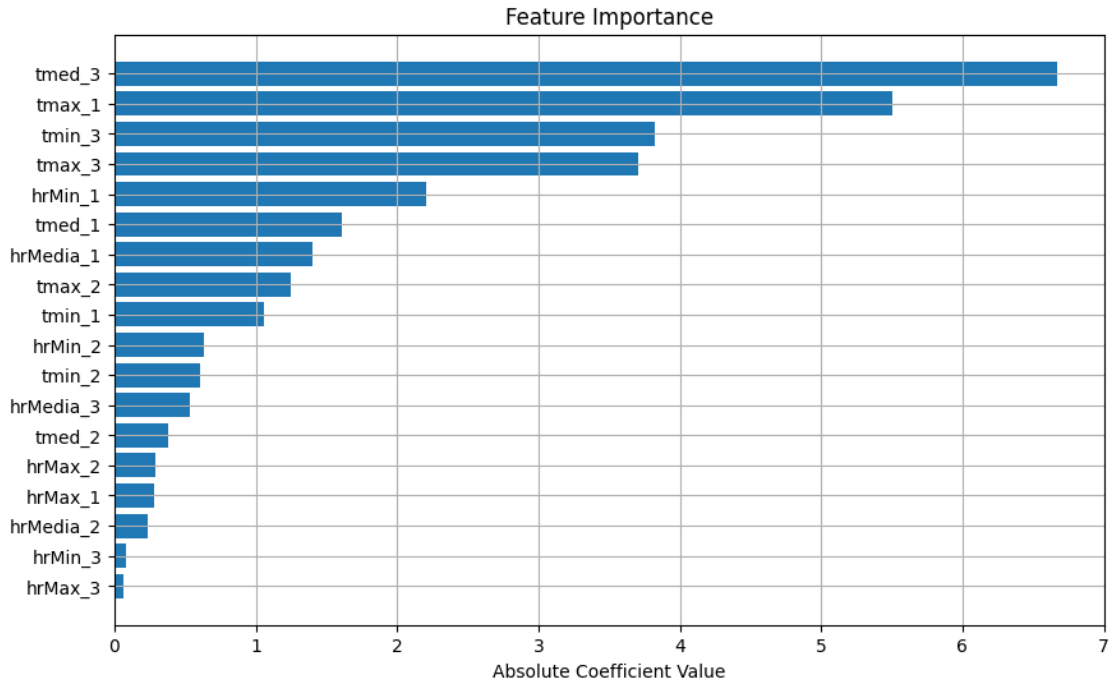
```
# Usa la función
metrics = evaluate_model(results, X_test, y_test, scaler)
```

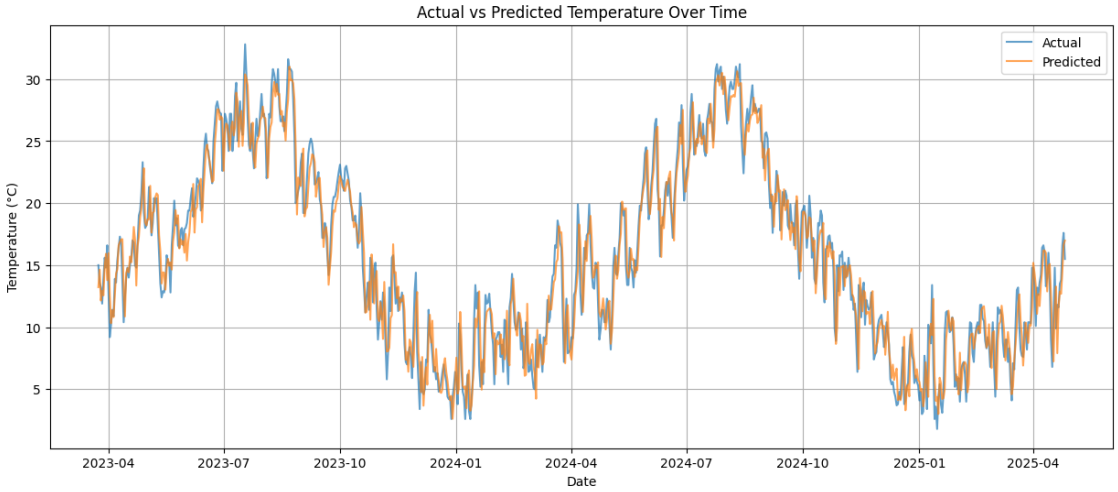
```
Model Performance Metrics:
R² Score: 0.9365
Mean Absolute Error: 1.4724°C
Root Mean Squared Error: 1.8672°C
Mean Absolute Percentage Error: 13.29%
```




Residuals vs Predicted Values
Residuals (°C)
Predicted Temperature (°C)
6
4
2
0
−2
−4
−6
5
10
15
20
25
30


Distribution of Residuals
Count
Residuals (°C)
100
80
60
40
20
0
−6
−4
−2
0
2
4
6





3 Entrenando redes de neuronas

[ ]:
```
# Primero, creemos algunos datos de muestra
np.random.seed(42)
n_samples = 1000
n_features = 5
```

```
# Crear conjunto de datos sintético
X = np.random.randn(n_samples, n_features)
y = 2 * X[:, 0] + 0.5 * X[:, 1] - X[:, 2] + 0.1 * X[:, 3] + np.random.
 ↪randn(n_samples) * 0.1
y = pd.Series(y)

# Dividir los datos en conjuntos de entrenamiento, validación y prueba
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,␣
 ↪random_state=42)

# Ahora puedes ejecutar el resto de tu código...


# Clase de conjunto de datos personalizada
class TemperatureDataset(Dataset):
    def __init__(self, X, y, scaler=None):
        if scaler is None:
            self.scaler = StandardScaler()
            self.X = torch.FloatTensor(self.scaler.fit_transform(X))
        else:
            self.scaler = scaler
            self.X = torch.FloatTensor(self.scaler.transform(X))

        self.y = torch.FloatTensor(y.values.reshape(-1, 1))

    def __len__(self):
        return len(self.X)

    def __getitem__(self, idx):
        return self.X[idx], self.y[idx]

# Modelo de Red Neuronal
class TemperatureNN(nn.Module):
    def __init__(self, input_size):
        super(TemperatureNN, self).__init__()
        self.layers = nn.Sequential(
            nn.Linear(input_size, 64),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(64, 32),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(32, 16),
            nn.ReLU(),
            nn.Linear(16, 1)
```

```
        )

    def forward(self, x):
        return self.layers(x)

# Función de entrenamiento
def train_model(model, train_loader, val_loader, criterion, optimizer,
                num_epochs, device, scheduler):
    train_losses = []
    val_losses = []

    epoch_bar = tqdm(range(num_epochs), desc="Entrenando", unit="epoch")

    for epoch in epoch_bar:
        # Fase de entrenamiento
        model.train()
        train_loss = 0
        for X_batch, y_batch in train_loader:
            X_batch, y_batch = X_batch.to(device), y_batch.to(device)

            optimizer.zero_grad()
            y_pred = model(X_batch)
            loss = criterion(y_pred, y_batch)

            loss.backward()
            optimizer.step()

            train_loss += loss.item()

        train_loss = train_loss / len(train_loader)
        train_losses.append(train_loss)

        # Fase de validación
        model.eval()
        val_loss = 0
        with torch.no_grad():
            for X_batch, y_batch in val_loader:
                X_batch, y_batch = X_batch.to(device), y_batch.to(device)
                y_pred = model(X_batch)
                loss = criterion(y_pred, y_batch)
                val_loss += loss.item()

        val_loss = val_loss / len(val_loader)
        val_losses.append(val_loss)

        scheduler.step(val_loss)
```

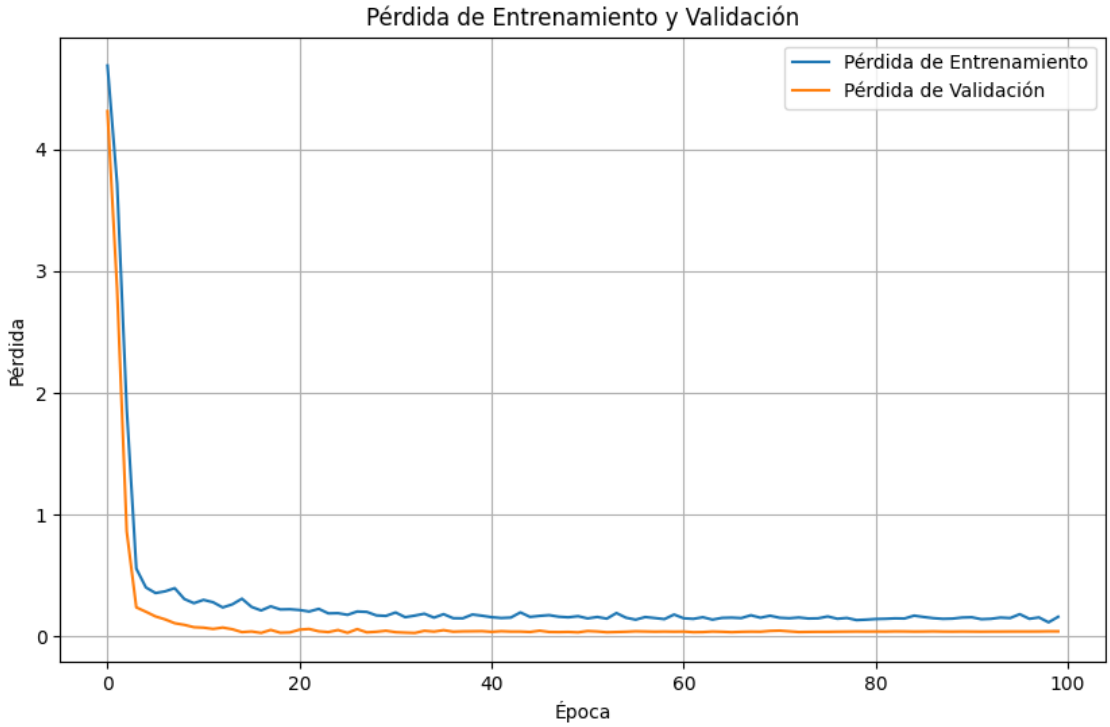
```
        if (epoch + 1) % 10 == 0:
            epoch_bar.set_postfix(
                train_loss=train_loss, val_loss=val_loss
            )
    print("Entrenamiento completo.")

    return train_losses, val_losses

# Función para evaluar el modelo
def evaluate_model(model, test_loader, criterion, device):
    model.eval()
    test_loss = 0
    predictions = []
    actuals = []

    with torch.no_grad():
        for X_batch, y_batch in test_loader:
            X_batch, y_batch = X_batch.to(device), y_batch.to(device)
            y_pred = model(X_batch)
            loss = criterion(y_pred, y_batch)
            test_loss += loss.item()

            predictions.extend(y_pred.cpu().numpy())
            actuals.extend(y_batch.cpu().numpy())

    test_loss = test_loss / len(test_loader)
    return test_loss, np.array(predictions), np.array(actuals)

# Pipeline principal de entrenamiento y evaluación
def run_neural_network(X_train, X_val, X_test, y_train, y_val, y_test):
    # Hiperparámetros
    # En la función run_neural_network, añadir:

    batch_size = 32
    learning_rate = 0.001
    num_epochs = 100
    device = torch.device("cuda" if torch.cuda.is_available() else "mps" if␣
 ↪torch.mps.is_available() else "cpu")
    print(f"Usando dispositivo: {device}")
    # Crear conjuntos de datos
    train_dataset = TemperatureDataset(X_train, y_train)
    val_dataset = TemperatureDataset(X_val, y_val, train_dataset.scaler)
    test_dataset = TemperatureDataset(X_test, y_test, train_dataset.scaler)
    print(f"Tamaño del conjunto de entrenamiento: {len(train_dataset)}")
    # Crear cargadores de datos
    train_loader = DataLoader(train_dataset, batch_size=batch_size,␣
 ↪shuffle=True)
```

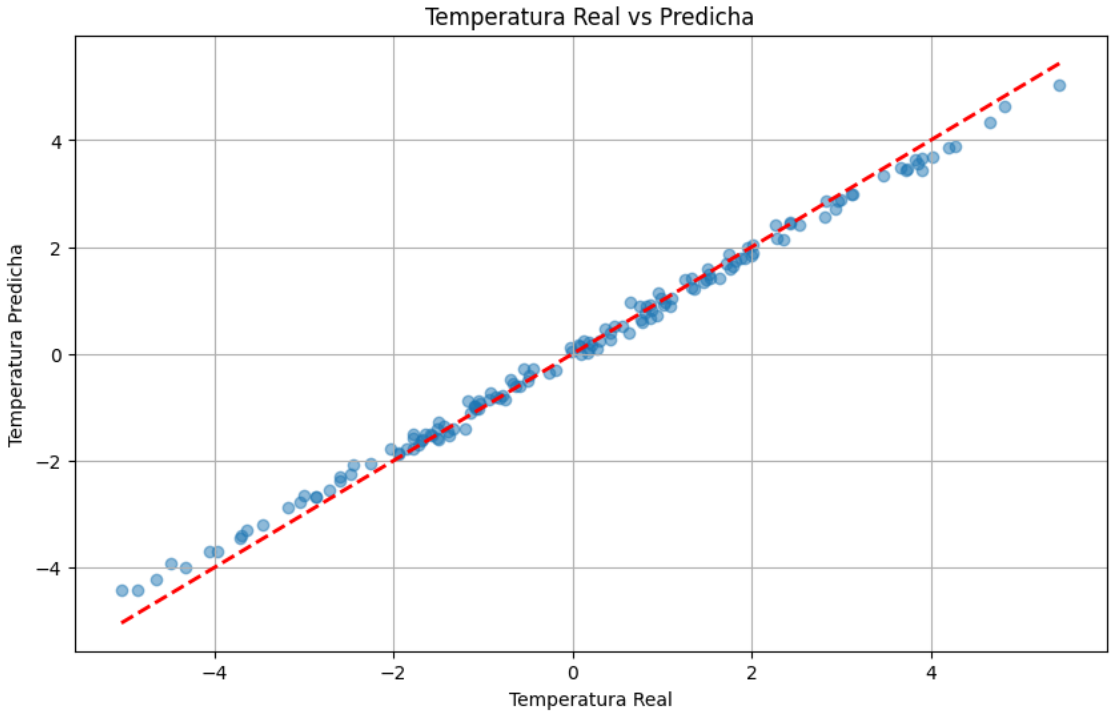
```
    val_loader = DataLoader(val_dataset, batch_size=batch_size)
    test_loader = DataLoader(test_dataset, batch_size=batch_size)
    print(f"Tamaño del cargador de entrenamiento: {len(train_loader)}")
    # Inicializar modelo
    model = TemperatureNN(input_size=X_train.shape[1]).to(device)

    # Función de pérdida y optimizador
    criterion = nn.MSELoss()
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
    scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(
        optimizer,
        mode='min',
        factor=0.5,
        patience=10
    )
    print(f"Parámetros del modelo: {sum(p.numel() for p in model.
↪parameters())}")
    # Entrenar el modelo
    train_losses, val_losses = train_model(
        model, train_loader, val_loader, criterion, optimizer, num_epochs,␣
↪device, scheduler=scheduler
    )
    print(f"Pérdidas de entrenamiento: {train_losses}")
    # Evaluar en el conjunto de prueba
    _, predictions, actuals = evaluate_model(
        model, test_loader, criterion, device
    )

    # Graficar curvas de entrenamiento
    plt.figure(figsize=(10, 6))
    plt.plot(train_losses, label='Pérdida de Entrenamiento')
    plt.plot(val_losses, label='Pérdida de Validación')
    plt.xlabel('Época')
    plt.ylabel('Pérdida')
    plt.title('Pérdida de Entrenamiento y Validación')
    plt.legend()
    plt.grid(True)
    plt.show()

    # Graficar predicciones vs valores reales
    plt.figure(figsize=(10, 6))
    plt.scatter(actuals, predictions.reshape(-1), alpha=0.5)
    plt.plot([actuals.min(), actuals.max()],
             [actuals.min(), actuals.max()],
             'r--', lw=2)
    plt.xlabel('Temperatura Real')
    plt.ylabel('Temperatura Predicha')
```

```
    plt.title('Temperatura Real vs Predicha')
    plt.grid(True)
    plt.show()

    # Calcular métricas
    mse = np.mean((predictions - actuals) ** 2)
    rmse = np.sqrt(mse)
    mae = np.mean(np.abs(predictions - actuals))
    r2 = 1 - np.sum((actuals - predictions) ** 2) / np.sum(
        (actuals - np.mean(actuals)) ** 2
    )

    print('\nMétricas de Prueba:')
    print(f'MSE: {mse:.4f}')
    print(f'RMSE: {rmse:.4f}')
    print(f'MAE: {mae:.4f}')
    print(f'R²: {r2:.4f}')

    return model, train_dataset.scaler

# Ejecutar la red neuronal
model, scaler = run_neural_network(X_train, X_val, X_test, y_train, y_val,␣
 ↪y_test)
```

```
Usando dispositivo: mps
Tamaño del conjunto de entrenamiento: 700
Tamaño del cargador de entrenamiento: 22
Parámetros del modelo: 3009

Entrenando: 100%|      | 100/100 [00:06<00:00, 15.01epoch/s,
train_loss=0.158, val_loss=0.0381]

Entrenamiento completo.
Pérdidas de entrenamiento: [4.690262566913258, 3.719262795014815,
1.8716158108277754, 0.554124577478929, 0.3989101594144648, 0.35316336290402844,
0.3668891733342951, 0.3930787762457674, 0.30397210676561703,
0.26975585181604733, 0.29682479121468286, 0.27725647999481723,
0.23448127575896002, 0.2600293450734832, 0.30658870799974963,
0.23908730867234143, 0.2099145461212505, 0.2436977946622805,
0.21817003495313905, 0.21971301869912582, 0.21340813487768173,
0.20113048736344685, 0.22290654141794553, 0.18580836701122197,
0.1873704449019649, 0.17388489198955623, 0.20127836140719327,
0.1979553648693995, 0.1691557839512825, 0.16512398387898097, 0.1932113908908584,
0.1545053127814423, 0.16750361736525188, 0.18287293037230318, 0.150958813726902,
0.17926104773174634, 0.14624035358428955, 0.14531275274401362,
0.17640797319737347, 0.16668435795740646, 0.15446451645005832,
0.1478761382062327, 0.15133806995370172, 0.19387255744500595,
0.1568323356861418, 0.16535552591085434, 0.17128170721910216,
0.15824069624597376, 0.15345826660367576, 0.16275519742207092,
```

0.14445795220407573, 0.15630496123974974, 0.14244998517361554,
0.188457795672796, 0.15012909471988678, 0.13418233072893185,
0.15568483942611652, 0.14739990268241276, 0.1386165591803464,
0.17629137601364742, 0.14562391794540666, 0.14072046632116492,
0.1541004285893657, 0.13453991948203606, 0.14879987321116708,
0.15086219747635451, 0.14716665853153577, 0.1702614240348339,
0.15015822310339322, 0.16670010950077663, 0.1498150737448172,
0.14591773633252492, 0.15200798365880142, 0.14367495240135628,
0.145187733694911, 0.1605608994987878, 0.14165496893904425, 0.14814209497787736,
0.13071627135981212, 0.13427995890378952, 0.13938709754835477,
0.14126964861696417, 0.14480751380324364, 0.14364805174144832,
0.16727473146536134, 0.1560153365135193, 0.14653625660999256,
0.14086255939169365, 0.14282868700948628, 0.1516052521765232,
0.15390651774677364, 0.13801831955259497, 0.14117202078076926,
0.151402662423524, 0.1473937752571973, 0.17924799329855226, 0.1411189619790424,
0.1525793153453957, 0.11218444342640313, 0.1576541828160936]






```
Métricas de Prueba:
MSE: 0.0380
RMSE: 0.1948
MAE: 0.1568
R²: 0.9925

Exception ignored in: <function ResourceTracker.__del__ at 0x124805ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x106e79ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
```

```
Exception ignored in: <function ResourceTracker.__del__ at 0x1067fdee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x108875ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x10693dee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x106ee5ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x103c4dee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x1070adee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
```

```
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x1196c1ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x112fe1ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x105561ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x107701ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x104495ee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
```

```
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
Exception ignored in: <function ResourceTracker.__del__ at 0x10af9dee0>
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 77, in __del__
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 86, in _stop
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/multip
rocessing/resource_tracker.py", line 111, in _stop_locked
ChildProcessError: [Errno 10] No child processes
```