

```
!wget https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip
!unzip jena_climate_2009_2016.csv.zip
```

```
--2025-04-06 20:50:56-- https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip
Resolving s3.amazonaws.com (s3.amazonaws.com)... 54.231.134.232, 52.217.199.48, 16.182.73.224, ...
Connecting to s3.amazonaws.com (s3.amazonaws.com)|54.231.134.232|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13565642 (13M) [application/zip]
Saving to: 'jena_climate_2009_2016.csv.zip'
```

```
jena_climate_2009_2 100%[=====] 12.94M 20.0MB/s in 0.6s
```

```
2025-04-06 20:50:57 (20.0 MB/s) - 'jena_climate_2009_2016.csv.zip' saved [13565642/13565642]
```

```
Archive: jena_climate_2009_2016.csv.zip
  inflating: jena_climate_2009_2016.csv
  inflating: __MACOSX/._jena_climate_2009_2016.csv
```

```
import os
input_data = os.path.join("jena_climate_2009_2016.csv")
```

```
with open(input_data) as f:
    input_data = f.read()
```

```
data_records = input_data.split("\n")
title = data_records[0].split(",")
data_records = data_records[1:]
print(title)
print(len(data_records))
import os
input_data = os.path.join("jena_climate_2009_2016.csv")
```

```
['Date Time', 'p (mbar)', 'T (degC)', 'Tpot (K)', 'Tdew (degC)', 'rh (%)', 'VPmax (mbar)', 'VPact (mbar)', '
420451
```

```
with open(input_data) as f:
    input_data = f.read()
```

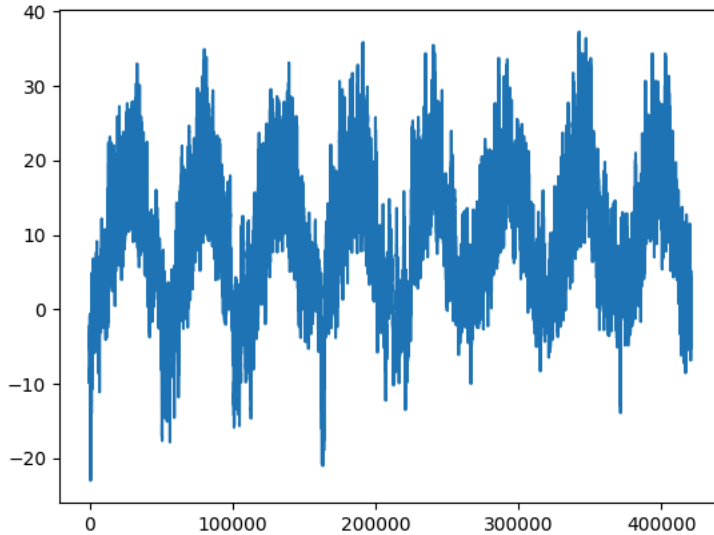
```
data_records = input_data.split("\n")
title = data_records[0].split(",")
data_records = data_records[1:]
print(title)
print(len(data_records))
#Handling and interpreting the dataset
```

```
['Date Time', 'p (mbar)', 'T (degC)', 'Tpot (K)', 'Tdew (degC)', 'rh (%)', 'VPmax (mbar)', 'VPact (mbar)', '
420451
```

```
import numpy as np
temp = np.zeros((len(data_records),))
original_input_data = np.zeros((len(data_records), len(title) - 1))
for i, line in enumerate(data_records):
    values = [float(x) for x in line.split(",")[1:]]
    temp[i] = values[1]
    original_input_data[i, :] = values[:]
import numpy as np
temp = np.zeros((len(data_records),))
original_input_data = np.zeros((len(data_records), len(title) - 1))
for i, line in enumerate(data_records):
    values = [float(x) for x in line.split(",")[1:]]
    temp[i] = values[1]
    original_input_data[i, :] = values[:]
#Plotting the temperature time series
```

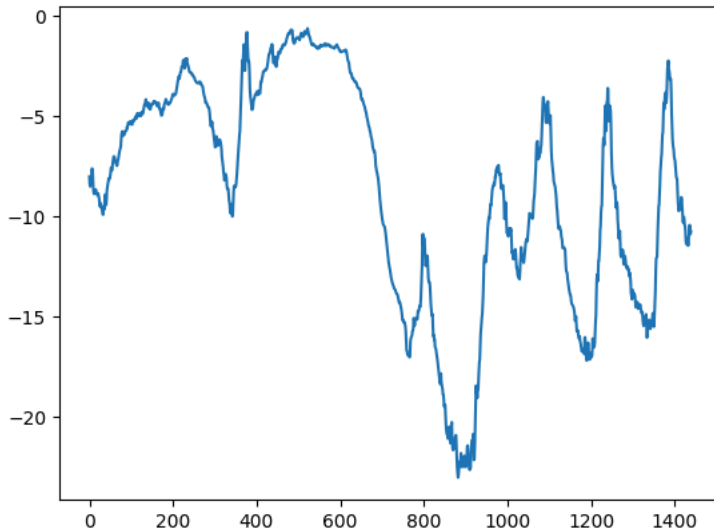
```
from matplotlib import pyplot as plt
plt.plot(range(len(temp)), temp)
#Plotting the temperature time series for the initial 10 days.
```

↗ [`<matplotlib.lines.Line2D at 0x7d3af3104210>`]



```
plt.plot(range(1440), temp[:1440])
#Determining the sample count for each data split.
```

↗ [`<matplotlib.lines.Line2D at 0x7d3ae47dd290>`]



```
train_Sample_Num = int(0.5 * len(original_input_data))
val_original_input_data = int(0.25 * len(original_input_data))
test_Sample_Number = len(original_input_data) - train_Sample_Num - val_original_input_data
print("train_Sample_Num:", train_Sample_Num)
print("val_original_input_data:", val_original_input_data)
print("test_Sample_Number:", test_Sample_Number)
#Data Preparation
#Scaling the data
```

↗

```
train_Sample_Num: 210225
val_original_input_data: 105112
test_Sample_Number: 105114
```

```
mean = original_input_data[:train_Sample_Num].mean(axis=0)
original_input_data -= mean
std = original_input_data[:train_Sample_Num].std(axis=0)
original_input_data /= std
import numpy as np
from tensorflow import keras
number_of_series = np.arange(10)
dummy_dataset = keras.utils.timeseries_dataset_from_array(
    data=number_of_series[:-3],
    targets=number_of_series[3:],
    sequence_length=3,
```

```

    batch_size=2,
)

for input_sequence, targets in dummy_dataset:
    for i in range(input_sequence.shape[0]):
        print([int(x) for x in input_sequence[i]], int(targets[i]))
#Creating datasets for training, validation, and testing.

```

```

↗ [0, 1, 2] 3
   [1, 2, 3] 4
   [2, 3, 4] 5
   [3, 4, 5] 6
   [4, 5, 6] 7

```

```

resample_rate = 6
seq_len = 120
forecast_horizon = resample_rate * (seq_len + 24 - 1)
batch_size = 256

```

```

training_dataset = keras.utils.timeseries_dataset_from_array(
    original_input_data[:-forecast_horizon],
    targets=temp[forecast_horizon:],
    sampling_rate=resample_rate,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=0,
    end_index=train_Sample_Num)

```

```

validation_dataset = keras.utils.timeseries_dataset_from_array(
    original_input_data[:-forecast_horizon],
    targets=temp[forecast_horizon:],
    sampling_rate=resample_rate,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=train_Sample_Num,
    end_index=train_Sample_Num + val_original_input_data)

```

```

test_dataset = keras.utils.timeseries_dataset_from_array(
    original_input_data[:-forecast_horizon],
    targets=temp[forecast_horizon:],
    sampling_rate=resample_rate,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=train_Sample_Num + val_original_input_data)
#Inspecting the contents and quality of one of our datasets.

```

```

for samples, targets in training_dataset:
    print("samples shape:", samples.shape)
    print("targets shape:", targets.shape)
    break
#Establishing a simple, non-machine-learning baseline
#Calculating the common-sense baseline MAE

```

```

↗ samples shape: (256, 120, 14)
   targets shape: (256,)

```

```

def evaluate_naive_method(dataset):
    total_abs_err = 0.
    samples_seen = 0
    for samples, targets in dataset:
        preds = samples[:, -1, 1] * std[1] + mean[1]
        total_abs_err += np.sum(np.abs(preds - targets))
        samples_seen += samples.shape[0]
    return total_abs_err / samples_seen

```

```

print(f"Validation MAE: {evaluate_naive_method(validation_dataset):.2f}")
print(f"Test MAE: {evaluate_naive_method(test_dataset):.2f}")

```

↻ Validation MAE: 2.44  
Test MAE: 2.62

```
!pip install tensorflow==2.12
```

```

Collecting tensorflow==2.12
  Downloading tensorflow-2.12.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (25.2.10)
Collecting gast<=0.4.0,>=0.2.1 (from tensorflow==2.12)
  Downloading gast-0.4.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.71.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (3.13.0)
Requirement already satisfied: jax>=0.3.15 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.5.2)
Collecting keras<2.13,>=2.12.0 (from tensorflow==2.12)
  Downloading keras-2.12.0-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (18.1.1)
Collecting numpy<1.24,>=1.22 (from tensorflow==2.12)
  Downloading numpy-1.23.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.3 kB)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (24.2)
Collecting protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.0dev,>=3.20.3 (from tensorflow==2.12)
  Downloading protobuf-4.25.6-cp37-abi3-manylinux2014_x86_64.whl.metadata (541 bytes)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.17.0)
Collecting tensorboard<2.13,>=2.12 (from tensorflow==2.12)
  Downloading tensorboard-2.12.3-py3-none-any.whl.metadata (1.8 kB)
Collecting tensorflow-estimator<2.13,>=2.12.0 (from tensorflow==2.12)
  Downloading tensorflow_estimator-2.12.0-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (4.12.0)
Collecting wrapt<1.15,>=1.11.0 (from tensorflow==2.12)
  Downloading wrapt-1.14.1-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.37.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.43.0)
Requirement already satisfied: jaxlib<=0.5.2,>=0.5.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.5.2)
Requirement already satisfied: ml_dtypes>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (0.5.2)
INFO: pip is looking at multiple versions of jax to determine which version is compatible with other requirements. This could take a while.
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.5.3-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.5.3,>=0.5.3 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.5.3-cp311-cp311-manylinux2014_x86_64.whl.metadata (1.2 kB)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.5.1-py3-none-any.whl.metadata (22 kB)
  Downloading jax-0.5.0-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.5.0,>=0.5.0 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.5.0-cp311-cp311-manylinux2014_x86_64.whl.metadata (978 bytes)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.38-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.38,>=0.4.38 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.38-cp311-cp311-manylinux2014_x86_64.whl.metadata (1.0 kB)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.37-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.37,>=0.4.36 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.36-cp311-cp311-manylinux2014_x86_64.whl.metadata (1.0 kB)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.36-py3-none-any.whl.metadata (22 kB)
  Downloading jax-0.4.35-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.35,>=0.4.34 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.35-cp311-cp311-manylinux2014_x86_64.whl.metadata (983 bytes)
INFO: pip is still looking at multiple versions of jax to determine which version is compatible with other requirements. This could take a while.
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.34-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.34,>=0.4.34 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.34-cp311-cp311-manylinux2014_x86_64.whl.metadata (983 bytes)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.33-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.33,>=0.4.33 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.33-cp311-cp311-manylinux2014_x86_64.whl.metadata (983 bytes)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.31-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.31,>=0.4.30 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.31-cp311-cp311-manylinux2014_x86_64.whl.metadata (983 bytes)
Collecting jax>=0.3.15 (from tensorflow==2.12)
  Downloading jax-0.4.30-py3-none-any.whl.metadata (22 kB)
Collecting jaxlib<=0.4.30,>=0.4.27 (from jax>=0.3.15->tensorflow==2.12)
  Downloading jaxlib-0.4.30-cp311-cp311-manylinux2014_x86_64.whl.metadata (1.0 kB)
Requirement already satisfied: scipy>=1.9 in /usr/local/lib/python3.11/dist-packages (from jax>=0.3.15->tensorflow==2.12) (1.12.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12) (2.33.0)
Collecting google-auth-oauthlib<1.1,>=0.5 (from tensorboard<2.13,>=2.12->tensorflow==2.12)
  Downloading google_auth_oauthlib-1.0.0-py2.py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12) (3.6.0)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12) (2.32.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12) (0.19.0)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12) (3.0.6)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3) (5.5.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3) (0.6.1)
Requirement already satisfied: rsa<4.14,>=4.1 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3) (4.9.1)

```

```

Requirement already satisfied: rsa<4,>=3.1.4 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.0.3->tensorflow)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from google-auth-oauthlib)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorflow)
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.11/dist-packages (from pyasn1-modules>=0.2.1->tensorflow)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.11/dist-packages (from requests-oauthlib>=0.7.0->tensorflow)
Downloading tensorflow-2.12.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (586.0 MB)
586.0/586.0 MB 3.3 MB/s eta 0:00:00
Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Downloading jax-0.4.30-py3-none-any.whl (2.0 MB)
2.0/2.0 MB 85.7 MB/s eta 0:00:00
Downloading keras-2.12.0-py2.py3-none-any.whl (1.7 MB)
1.7/1.7 MB 80.1 MB/s eta 0:00:00
Downloading numpy-1.23.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
17.1/17.1 MB 108.1 MB/s eta 0:00:00
Downloading protobuf-4.25.6-cp37-abi3-manylinux2014_x86_64.whl (294 kB)
294.6/294.6 kB 27.7 MB/s eta 0:00:00
Downloading tensorboard-2.12.3-py3-none-any.whl (5.6 MB)
5.6/5.6 MB 105.0 MB/s eta 0:00:00
Downloading tensorflow_estimator-2.12.0-py2.py3-none-any.whl (440 kB)
440.7/440.7 kB 33.2 MB/s eta 0:00:00
Downloading wrapt-1.14.1-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (78.4 kB)
78.4/78.4 kB 8.7 MB/s eta 0:00:00
Downloading google_auth_oauthlib-1.0.0-py2.py3-none-any.whl (18 kB)
Downloading jaxlib-0.4.30-cp311-cp311-manylinux2014_x86_64.whl (79.6 MB)
79.6/79.6 MB 9.6 MB/s eta 0:00:00
Installing collected packages: wrapt, tensorflow-estimator, protobuf, numpy, keras, gast, jaxlib, google-auth-oauthlib, tensorflow
Attempting uninstall: wrapt
  Found existing installation: wrapt 1.17.2
  Uninstalling wrapt-1.17.2:
    Successfully uninstalled wrapt-1.17.2
Attempting uninstall: protobuf
  Found existing installation: protobuf 5.29.4
  Uninstalling protobuf-5.29.4:
    Successfully uninstalled protobuf-5.29.4
Attempting uninstall: numpy
  Found existing installation: numpy 2.0.2
  Uninstalling numpy-2.0.2:
    Successfully uninstalled numpy-2.0.2
Attempting uninstall: keras
  Found existing installation: keras 3.8.0
  Uninstalling keras-3.8.0:
    Successfully uninstalled keras-3.8.0
Attempting uninstall: gast
  Found existing installation: gast 0.6.0
  Uninstalling gast-0.6.0:
    Successfully uninstalled gast-0.6.0
Attempting uninstall: jaxlib
  Found existing installation: jaxlib 0.5.1
  Uninstalling jaxlib-0.5.1:
    Successfully uninstalled jaxlib-0.5.1
Attempting uninstall: google-auth-oauthlib
  Found existing installation: google-auth-oauthlib 1.2.1
  Uninstalling google-auth-oauthlib-1.2.1:
    Successfully uninstalled google-auth-oauthlib-1.2.1
Attempting uninstall: tensorboard
  Found existing installation: tensorboard 2.18.0
  Uninstalling tensorboard-2.18.0:
    Successfully uninstalled tensorboard-2.18.0
Attempting uninstall: jax
  Found existing installation: jax 0.5.2
  Uninstalling jax-0.5.2:
    Successfully uninstalled jax-0.5.2
Attempting uninstall: tensorflow
  Found existing installation: tensorflow 2.18.0
  Uninstalling tensorflow-2.18.0:
    Successfully uninstalled tensorflow-2.18.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is
  deprecated. Use pip install --upgrade pip to fix to use the wheelresolver.
  grpcio-status 1.71.0 requires protobuf<6.0dev,>=5.26.1, but you have protobuf 4.25.6 which is incompatible.
  tf-keras 2.18.0 requires tensorflow<2.19,>=2.18, but you have tensorflow 2.12.0 which is incompatible.
  imbalanced-learn 0.13.0 requires numpy<3,>=1.24.3, but you have numpy 1.23.5 which is incompatible.
  chex 0.1.89 requires numpy>=1.24.1, but you have numpy 1.23.5 which is incompatible.
  albucore 0.0.23 requires numpy>=1.24.4, but you have numpy 1.23.5 which is incompatible.
  pymc 5.21.1 requires numpy>=1.25.0, but you have numpy 1.23.5 which is incompatible.
  blocs2 3.2.1 requires numpy>=1.26, but you have numpy 1.23.5 which is incompatible.
  albumations 2.0.5 requires numpy>=1.24.4, but you have numpy 1.23.5 which is incompatible.
  flax 0.10.5 requires jax>=0.5.1, but you have jax 0.4.30 which is incompatible.
  treescope 0.1.9 requires numpy>=1.25.2, but you have numpy 1.23.5 which is incompatible.
  bigframes 1.42.0 requires numpy>=1.24.0, but you have numpy 1.23.5 which is incompatible.
  orbax-checkpoint 0.11.10 requires jax>=0.5.0, but you have jax 0.4.30 which is incompatible.
  tensorflow-text 2.18.1 requires tensorflow<2.19,>=2.18.0, but you have tensorflow 2.12.0 which is incompatible.
  xarray 2025.1.2 requires numpy>=1.24, but you have numpy 1.23.5 which is incompatible.

```

scikit-image 0.25.2 requires numpy>=1.24, but you have numpy 1.23.5 which is incompatible.

Successfully installed gast-0.4.0 google-auth-oauthlib-1.0.0 jax-0.4.30 jaxlib-0.4.30 keras-2.12.0 numpy-1.23.5 protobuf-4.2

**WARNING: The following packages were previously imported in this runtime:**

[gast,jax,jaxlib,keras,numpy,tensorflow,wrap]

**You must restart the runtime in order to use newly installed versions.**

RESTART SESSION

```

#!pip install tensorflow

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

input_sequence = keras.Input(shape=(seq_len, original_input_data.shape[-1]))
x = layers.Flatten()(input_sequence)
x = layers.Dense(64, activation="relu")(x)
outputs = layers.Dense(1)(x)
model = keras.Model(input_sequence, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_dense.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(training_dataset,
                    epochs=10,
                    validation_data=validation_dataset,
                    callbacks=callbacks)

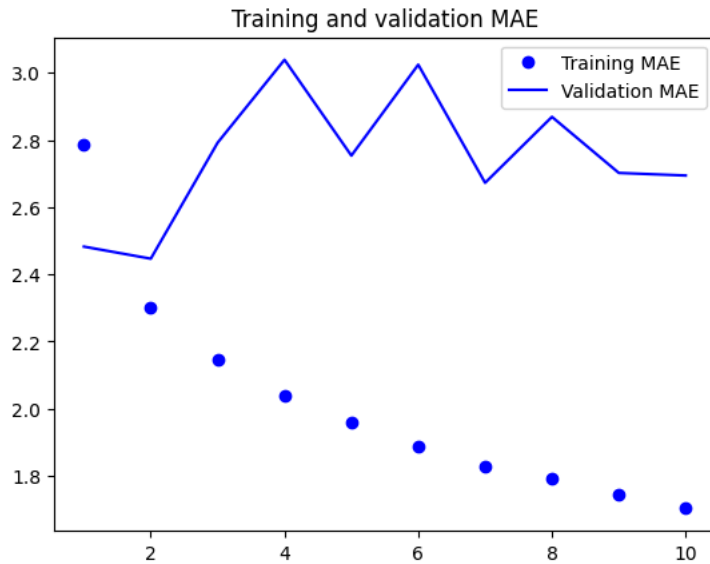
model = keras.models.load_model("jena_dense.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
#Plotting the results

↩ Epoch 1/10
819/819 [=====] - 41s 50ms/step - loss: 12.9411 - mae: 2.7845 - val_loss: 9.9679 - val_mae: 2.4825
Epoch 2/10
819/819 [=====] - 40s 49ms/step - loss: 8.5806 - mae: 2.3000 - val_loss: 9.7218 - val_mae: 2.4465
Epoch 3/10
819/819 [=====] - 40s 49ms/step - loss: 7.4567 - mae: 2.1461 - val_loss: 12.5211 - val_mae: 2.7921
Epoch 4/10
819/819 [=====] - 39s 48ms/step - loss: 6.7179 - mae: 2.0394 - val_loss: 14.5891 - val_mae: 3.0392
Epoch 5/10
819/819 [=====] - 40s 48ms/step - loss: 6.1964 - mae: 1.9590 - val_loss: 12.2066 - val_mae: 2.7536
Epoch 6/10
819/819 [=====] - 41s 49ms/step - loss: 5.7521 - mae: 1.8876 - val_loss: 14.6493 - val_mae: 3.0245
Epoch 7/10
819/819 [=====] - 40s 48ms/step - loss: 5.3997 - mae: 1.8271 - val_loss: 11.5340 - val_mae: 2.6726
Epoch 8/10
819/819 [=====] - 40s 49ms/step - loss: 5.1346 - mae: 1.7904 - val_loss: 13.2660 - val_mae: 2.8690
Epoch 9/10
819/819 [=====] - 42s 51ms/step - loss: 4.8744 - mae: 1.7423 - val_loss: 11.7320 - val_mae: 2.7020
Epoch 10/10
819/819 [=====] - 50s 60ms/step - loss: 4.6546 - mae: 1.7023 - val_loss: 11.7609 - val_mae: 2.6943
405/405 [=====] - 12s 30ms/step - loss: 10.8928 - mae: 2.6049
Test MAE: 2.60

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE")
plt.legend()
plt.show()
#Let's try a 1D convolutional model
input_sequence = keras.Input(shape=(seq_len, original_input_data.shape[-1]))
x = layers.Conv1D(8, 24, activation="relu")(input_sequence)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 12, activation="relu")(x)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 6, activation="relu")(x)
x = layers.GlobalAveragePooling1D()(x)
outputs = layers.Dense(1)(x)
model = keras.Model(input_sequence, outputs)

```





```
callbacks = [
    keras.callbacks.ModelCheckpoint("jena_conv.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(training_dataset,
                    epochs=10,
                    validation_data=validation_dataset,
                    callbacks=callbacks)
```



```
Epoch 1/10
819/819 [=====] - 74s 90ms/step - loss: 22.4213 - mae: 3.6987 - val_loss: 15.0283 - val_mae: 3.0712
Epoch 2/10
819/819 [=====] - 72s 87ms/step - loss: 15.6346 - mae: 3.1391 - val_loss: 15.7515 - val_mae: 3.1269
Epoch 3/10
819/819 [=====] - 74s 90ms/step - loss: 14.1954 - mae: 2.9883 - val_loss: 16.0253 - val_mae: 3.1784
Epoch 4/10
819/819 [=====] - 70s 86ms/step - loss: 13.2898 - mae: 2.8888 - val_loss: 15.7244 - val_mae: 3.1353
Epoch 5/10
819/819 [=====] - 70s 86ms/step - loss: 12.5617 - mae: 2.8033 - val_loss: 19.0051 - val_mae: 3.4566
Epoch 6/10
819/819 [=====] - 73s 89ms/step - loss: 12.0608 - mae: 2.7462 - val_loss: 15.9391 - val_mae: 3.1481
Epoch 7/10
819/819 [=====] - 75s 91ms/step - loss: 11.6750 - mae: 2.7008 - val_loss: 14.0282 - val_mae: 2.9327
Epoch 8/10
819/819 [=====] - 74s 91ms/step - loss: 11.3166 - mae: 2.6612 - val_loss: 14.9161 - val_mae: 3.0352
Epoch 9/10
819/819 [=====] - 74s 90ms/step - loss: 11.0434 - mae: 2.6308 - val_loss: 14.3984 - val_mae: 2.9748
Epoch 10/10
819/819 [=====] - 75s 91ms/step - loss: 10.7691 - mae: 2.5982 - val_loss: 15.5831 - val_mae: 3.1051
```

```
model = keras.models.load_model("jena_conv.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
#Setting up an initial recurrent model as a baseline
#A basic LSTM-based model
```



```
405/405 [=====] - 17s 42ms/step - loss: 14.8875 - mae: 3.0596
Test MAE: 3.06
```

```
input_sequence = keras.Input(shape=(seq_len, original_input_data.shape[-1]))
x = layers.LSTM(16)(input_sequence)
outputs = layers.Dense(1)(x)
model = keras.Model(input_sequence, outputs)
```

```
callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(training_dataset,
                    epochs=10,
```

```
validation_data=validation_dataset,
callbacks=callbacks)
```

```
Epoch 1/10
819/819 [=====] - 97s 114ms/step - loss: 40.5929 - mae: 4.6170 - val_loss: 13.2996 - val_mae: 2.753
Epoch 2/10
819/819 [=====] - 95s 116ms/step - loss: 11.1102 - mae: 2.5998 - val_loss: 9.6430 - val_mae: 2.4144
Epoch 3/10
819/819 [=====] - 93s 113ms/step - loss: 9.9528 - mae: 2.4708 - val_loss: 9.5474 - val_mae: 2.3990
Epoch 4/10
819/819 [=====] - 95s 116ms/step - loss: 9.5048 - mae: 2.4079 - val_loss: 9.4277 - val_mae: 2.3861
Epoch 5/10
819/819 [=====] - 92s 111ms/step - loss: 9.1973 - mae: 2.3623 - val_loss: 9.5526 - val_mae: 2.3956
Epoch 6/10
819/819 [=====] - 92s 112ms/step - loss: 8.8806 - mae: 2.3243 - val_loss: 9.4453 - val_mae: 2.3872
Epoch 7/10
819/819 [=====] - 113s 138ms/step - loss: 8.5475 - mae: 2.2799 - val_loss: 9.8424 - val_mae: 2.4427
Epoch 8/10
819/819 [=====] - 111s 136ms/step - loss: 8.2766 - mae: 2.2401 - val_loss: 9.5921 - val_mae: 2.4260
Epoch 9/10
819/819 [=====] - 91s 111ms/step - loss: 8.0856 - mae: 2.2147 - val_loss: 9.6711 - val_mae: 2.4336
Epoch 10/10
819/819 [=====] - 91s 111ms/step - loss: 7.8972 - mae: 2.1899 - val_loss: 10.1271 - val_mae: 2.4748
```

```
model = keras.models.load_model("jena_lstm.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
#Exploring recurrent neural networks
#NumPy implementation of a basic RNN
```

```
405/405 [=====] - 23s 55ms/step - loss: 10.8176 - mae: 2.5720
Test MAE: 2.57
```

```
import numpy as np
timesteps = 100
input_features = 32
output_features = 64
input_sequence = np.random.random((timesteps, input_features))
state_t = np.zeros((output_features,))
W = np.random.random((output_features, input_features))
U = np.random.random((output_features, output_features))
b = np.random.random((output_features,))
successive_outputs = []
for input_t in input_sequence:
    output_t = np.tanh(np.dot(W, input_t) + np.dot(U, state_t) + b)
    successive_outputs.append(output_t)
    state_t = output_t
final_output_sequence = np.stack(successive_outputs, axis=0)
```

```
total_input_features = 14
input_sequence = keras.Input(shape=(None, total_input_features))
outputs = layers.SimpleRNN(16)(input_sequence)
```

```
total_input_features = 14
steps = 120
input_sequence = keras.Input(shape=(steps, total_input_features))
outputs = layers.SimpleRNN(16, return_sequences=False)(input_sequence)
print(outputs.shape)
```

```
(None, 16)
```

```
total_input_features = 14
steps = 120
input_sequence = keras.Input(shape=(steps, total_input_features))
outputs = layers.SimpleRNN(16, return_sequences=True)(input_sequence)
print(outputs.shape)
```

```
(None, 120, 16)
```

```
input_sequence = keras.Input(shape=(steps, total_input_features))
x = layers.SimpleRNN(16, return_sequences=True)(input_sequence)
x = layers.SimpleRNN(16, return_sequences=True)(x)
```

```

outputs = layers.SimpleRNN(16)(x)
#Delving into advanced techniques with recurrent neural networks
#Implementing recurrent dropout to reduce overfitting
#Training and evaluating an LSTM model with dropout regularization

input_sequence = keras.Input(shape=(seq_len, original_input_data.shape[-1]))
x = layers.LSTM(32, recurrent_dropout=0.25)(input_sequence)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1)(x)
model = keras.Model(input_sequence, outputs)

```

```

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(training_dataset,
                    epochs=10,
                    validation_data=validation_dataset,
                    callbacks=callbacks)

input_sequence = keras.Input(shape=(seq_len, total_input_features))
x = layers.LSTM(32, recurrent_dropout=0.2, unroll=True)(input_sequence)

```

```

↺ Epoch 1/10
819/819 [=====] - 198s 238ms/step - loss: 31.7879 - mae: 4.1215 - val_loss: 9.9875 - val_mae: 2.468
Epoch 2/10
819/819 [=====] - 190s 232ms/step - loss: 14.9627 - mae: 3.0127 - val_loss: 9.3159 - val_mae: 2.376
Epoch 3/10
819/819 [=====] - 192s 234ms/step - loss: 14.0260 - mae: 2.9068 - val_loss: 9.2858 - val_mae: 2.367
Epoch 4/10
819/819 [=====] - 193s 235ms/step - loss: 13.3587 - mae: 2.8376 - val_loss: 9.1758 - val_mae: 2.356
Epoch 5/10
819/819 [=====] - 192s 234ms/step - loss: 12.7564 - mae: 2.7726 - val_loss: 9.5594 - val_mae: 2.413
Epoch 6/10
819/819 [=====] - 190s 232ms/step - loss: 12.3837 - mae: 2.7342 - val_loss: 9.2558 - val_mae: 2.376
Epoch 7/10
819/819 [=====] - 188s 230ms/step - loss: 11.9981 - mae: 2.6930 - val_loss: 9.5601 - val_mae: 2.411
Epoch 8/10
819/819 [=====] - 188s 229ms/step - loss: 11.7042 - mae: 2.6577 - val_loss: 9.7174 - val_mae: 2.427
Epoch 9/10
819/819 [=====] - 203s 248ms/step - loss: 11.4648 - mae: 2.6271 - val_loss: 9.9593 - val_mae: 2.460
Epoch 10/10
819/819 [=====] - 188s 229ms/step - loss: 11.2468 - mae: 2.6000 - val_loss: 9.7238 - val_mae: 2.431

```

```

input_sequence = keras.Input(shape=(seq_len, original_input_data.shape[-1]))
x = layers.GRU(32, recurrent_dropout=0.5, return_sequences=True)(input_sequence)
x = layers.GRU(32, recurrent_dropout=0.5)(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1)(x)
model = keras.Model(input_sequence, outputs)

```

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or generate with AI.