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

```
--2025-04-06 07:36:52--  https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.146.181, 52.217.96.14, 16.182.66.56, ...
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.216.146.181|: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  18.7MB/s    in 0.7s

2025-04-06 07:36:53 (18.7 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
file_path = os.path.join("jena_climate_2009_2016.csv")


with open(file_path) as f:
    file_path = f.read()


climate_records = file_path.split("\n")
column_headers = climate_records[0].split(",")
climate_records = climate_records[1:]
print(column_headers)
print(len(climate_records))
import os
file_path = os.path.join("jena_climate_2009_2016.csv")
```

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

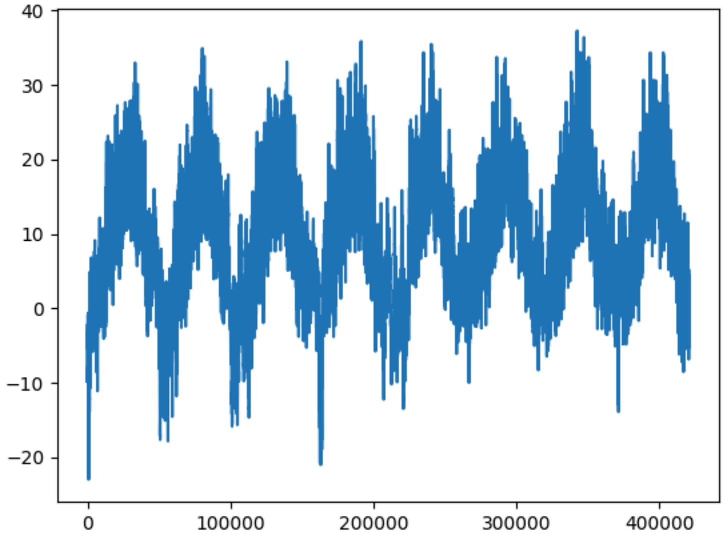
```
with open(file_path) as f:
    file_path = f.read()


climate_records = file_path.split("\n")
column_headers = climate_records[0].split(",")
climate_records = climate_records[1:]
print(column_headers)
print(len(climate_records))
#Loading and exploring the input data.
```

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

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

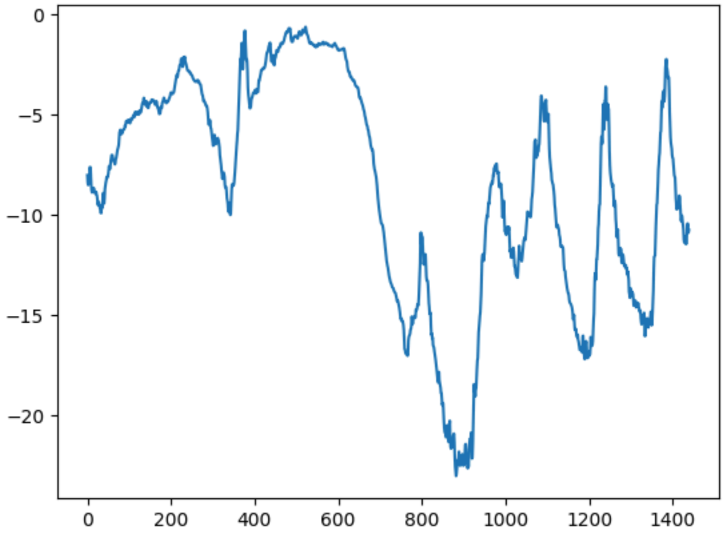
```
from matplotlib import pyplot as plt
plt.plot(range(len(temp)), temp)
#Plotting temperature trends for the initial 10 days
```

```
[<matplotlib.lines.Line2D at 0x7e8095cfdd10>]
```



```
plt.plot(range(1440), temp[:1440])
# Determining dataset split sizes
```

```
[<matplotlib.lines.Line2D at 0x7e809530d490>]
```



```
num_train_samples = int(0.5 * len(original_file_path))
val_original_file_path = int(0.25 * len(original_file_path))
test_Sample_Number = len(original_file_path) - num_train_samples - val_original_file_path
print("train_Sample_Num:", num_train_samples)
print("val_original_file_path:", val_original_file_path)
print("test_Sample_Number:", test_Sample_Number)
# Dataset preparation
#Normalizing the dataset features
```

```
train_Sample_Num: 210225
val_original_file_path: 105112
test_Sample_Number: 105114
```

```
mean = original_file_path[:num_train_samples].mean(axis=0)
original_file_path -= mean
std = original_file_path[:num_train_samples].std(axis=0)
original_file_path /= std
import numpy as np
from tensorflow import keras
num_series = np.arange(10)
dummy_dataset = keras.utils.timeseries_dataset_from_array(
    data=num_series[:-3],
    targets=num_series[3:],
    sequence_length=3,
    batch_size=2,
)


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

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

```
sampling_interval = 6
seq_len = 120
prediction_offset = sampling_interval * (seq_len + 24 - 1)
batch_size = 256


train_file_pathset = keras.utils.timeseries_dataset_from_array(
    original_file_path[:-prediction_offset],
    targets=temp[prediction_offset:],
    sampling_rate=sampling_interval,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=0,
    end_index=num_train_samples)


val_file_pathset = keras.utils.timeseries_dataset_from_array(
    original_file_path[:-prediction_offset],
    targets=temp[prediction_offset:],
    sampling_rate=sampling_interval,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples,
    end_index=num_train_samples + val_original_file_path)


test_file_pathset = keras.utils.timeseries_dataset_from_array(
    original_file_path[:-prediction_offset],
    targets=temp[prediction_offset:],
    sampling_rate=sampling_interval,
    sequence_length=seq_len,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples + val_original_file_path)
#Inspecting a sample batch from the dataset to evaluate structure and content


for samples, targets in train_file_pathset:
    print("samples shape:", samples.shape)
    print("targets shape:", targets.shape)
    break
#A common-sense, non-machine-learning baseline
#Measuring baseline model performance using MAE
```

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

```
def evaluate_naive_method(file_pathset):
    total_abs_err = 0.
    samples_seen = 0
    for samples, targets in file_pathset:
        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(val_file_pathset):.2f}")
print(f"Test MAE: {evaluate_naive_method(test_file_pathset):.2f}")
```

```
Validation MAE: 2.44
Test MAE: 2.62
```

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

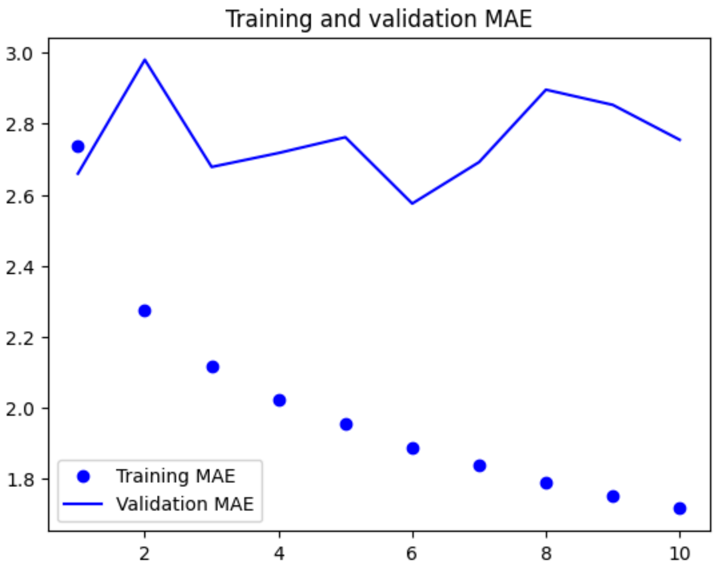
```
Collecting tensorflow==2.12
  Downloading tensorflow-2.12.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1.6
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (25.2
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
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12) (1
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
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
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.
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.1
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.manylinux_2_17_x86_64.manylinux2014_x86_64
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from ten
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->te
Requirement already satisfied: jaxlib<=0.5.2,>=0.5.1 in /usr/local/lib/python3.11/dist-packages (from jax>=0.3.15->tenso
Requirement already satisfied: ml_dtypes>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from jax>=0.3.15->tensorflow
INFO: pip is looking at multiple versions of jax to determine which version is compatible with other requirements. This
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.
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
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>
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-
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from te
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3->ten
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from google-auth-oau
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tenso
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.11/dist-packages (from pyasn1-modules>=0.2
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.11/dist-packages (from requests-oauthlib>=0.7.0
Downloading tensorflow-2.12.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (586.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 586.0/586.0 MB 853.5 kB/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 80.1 MB/s eta 0:00:00
Downloading keras-2.12.0-py2.py3-none-any.whl (1.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 79.4 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 101.9 MB/s eta 0:00:00
Downloading protobuf-4.25.6-cp37-abi3-manylinux2014_x86_64.whl (294 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 294.6/294.6 kB 27.0 MB/s eta 0:00:00
Downloading tensorboard-2.12.3-py3-none-any.whl (5.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 86.7 MB/s eta 0:00:00
Downloading tensorflow_estimator-2.12.0-py2.py3-none-any.whl (440 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 440.7/440.7 kB 35.6 MB/s eta 0:00:00
Downloading wrapt-1.14.1-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.w
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.4/78.4 kB 8.5 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 10.0 MB/s eta 0:00:00
Installing collected packages: wrapt, tensorflow-estimator, protobuf, numpy, keras, gast, jaxlib, google-auth-oauthlib,
  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 behaviou
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.
blosc2 3.2.1 requires numpy>=1.26, but you have numpy 1.23.5 which is incompatible.
albumentations 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
WARNING: The following packages were previously imported in this runtime:
  [gast,jax,jaxlib,keras,numpy,tensorflow,wrapt]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
from tensorflow import keras
from tensorflow.keras import layers

inputs = keras.Input(shape=(seq_len, original_file_path.shape[-1]))
x = layers.Flatten()(inputs)
x = layers.Dense(64, activation="relu")(x)
model_outputs    = layers.Dense(1)(x)
model = keras.Model(inputs, model_outputs   )

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

model = keras.models.load_model("jena_dense.keras")
print(f"Test MAE: {model.evaluate(test_file_pathset)[1]:.2f}")
```

```
Epoch 1/10
819/819 [==============================] - 42s 51ms/step - loss: 12.5079 - mae: 2.7378 - val_loss: 11.2760 - val_mae: 2.
Epoch 2/10
819/819 [==============================] - 41s 49ms/step - loss: 8.3840 - mae: 2.2759 - val_loss: 13.9872 - val_mae: 2.9
Epoch 3/10
819/819 [==============================] - 41s 50ms/step - loss: 7.2584 - mae: 2.1174 - val_loss: 11.5287 - val_mae: 2.6
Epoch 4/10
819/819 [==============================] - 42s 51ms/step - loss: 6.5942 - mae: 2.0230 - val_loss: 11.7205 - val_mae: 2.7
Epoch 5/10
819/819 [==============================] - 49s 60ms/step - loss: 6.1328 - mae: 1.9554 - val_loss: 12.3230 - val_mae: 2.7
Epoch 6/10
819/819 [==============================] - 40s 49ms/step - loss: 5.7205 - mae: 1.8883 - val_loss: 10.6712 - val_mae: 2.5
Epoch 7/10
819/819 [==============================] - 49s 59ms/step - loss: 5.4281 - mae: 1.8376 - val_loss: 11.6565 - val_mae: 2.6
Epoch 8/10
819/819 [==============================] - 40s 49ms/step - loss: 5.1418 - mae: 1.7892 - val_loss: 13.6404 - val_mae: 2.8
Epoch 9/10
819/819 [==============================] - 41s 50ms/step - loss: 4.9173 - mae: 1.7530 - val_loss: 13.1373 - val_mae: 2.8
Epoch 10/10
819/819 [==============================] - 41s 50ms/step - loss: 4.7305 - mae: 1.7168 - val_loss: 12.0464 - val_mae: 2.7
405/405 [==============================] - 13s 31ms/step - loss: 11.6951 - mae: 2.6879
Test MAE: 2.69
```

```
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
inputs = keras.Input(shape=(seq_len, original_file_path.shape[-1]))
x = layers.Conv1D(8, 24, activation="relu")(inputs)
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)
model_outputs    = layers.Dense(1)(x)
model = keras.Model(inputs, model_outputs   )
```



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

```
Epoch 1/10
819/819 [==============================] - 76s 92ms/step - loss: 21.8567 - mae: 3.6682 - val_loss: 16.1385 - val_mae: 3.
Epoch 2/10
819/819 [==============================] - 85s 104ms/step - loss: 15.6008 - mae: 3.1440 - val_loss: 15.1834 - val_mae: 3
Epoch 3/10
819/819 [==============================] - 84s 102ms/step - loss: 14.4357 - mae: 3.0209 - val_loss: 16.3196 - val_mae: 3
Epoch 4/10
819/819 [==============================] - 80s 97ms/step - loss: 13.6544 - mae: 2.9343 - val_loss: 16.5629 - val_mae: 3.
Epoch 5/10
819/819 [==============================] - 79s 96ms/step - loss: 13.0269 - mae: 2.8599 - val_loss: 15.0297 - val_mae: 3.
Epoch 6/10
819/819 [==============================] - 74s 91ms/step - loss: 12.5610 - mae: 2.8051 - val_loss: 16.9485 - val_mae: 3.
Epoch 7/10
819/819 [==============================] - 77s 94ms/step - loss: 12.1834 - mae: 2.7606 - val_loss: 15.5334 - val_mae: 3.
Epoch 8/10
819/819 [==============================] - 77s 93ms/step - loss: 11.8229 - mae: 2.7172 - val_loss: 16.1864 - val_mae: 3.
Epoch 9/10
819/819 [==============================] - 75s 92ms/step - loss: 11.5371 - mae: 2.6855 - val_loss: 15.8597 - val_mae: 3.
Epoch 10/10
819/819 [==============================] - 77s 93ms/step - loss: 11.2517 - mae: 2.6514 - val_loss: 16.0049 - val_mae: 3.
```

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

```
405/405 [==============================] - 17s 40ms/step - loss: 16.0901 - mae: 3.1756
Test MAE: 3.18
```

```
inputs = keras.Input(shape=(seq_len, original_file_path.shape[-1]))
x = layers.LSTM(16)(inputs)
model_outputs    = layers.Dense(1)(x)
model = keras.Model(inputs, model_outputs   )
```

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

```
Epoch 1/10
819/819 [==============================] - 99s 119ms/step - loss: 37.8345 - mae: 4.4654 - val_loss: 12.1753 - val_mae: 2
Epoch 2/10
```

```
819/819 [==============================] - 95s 115ms/step - loss: 10.9784 - mae: 2.5854 - val_loss: 10.0824 - val_mae: 2
Epoch 3/10
819/819 [==============================] - 98s 120ms/step - loss: 9.9057 - mae: 2.4542 - val_loss: 9.9743 - val_mae: 2.4
Epoch 4/10
819/819 [==============================] - 98s 120ms/step - loss: 9.5329 - mae: 2.4016 - val_loss: 9.9208 - val_mae: 2.4
Epoch 5/10
819/819 [==============================] - 96s 117ms/step - loss: 9.2077 - mae: 2.3601 - val_loss: 9.3302 - val_mae: 2.3
Epoch 6/10
819/819 [==============================] - 94s 115ms/step - loss: 8.9401 - mae: 2.3262 - val_loss: 9.2844 - val_mae: 2.3
Epoch 7/10
819/819 [==============================] - 97s 118ms/step - loss: 8.7727 - mae: 2.3032 - val_loss: 9.4830 - val_mae: 2.4
Epoch 8/10
819/819 [==============================] - 115s 140ms/step - loss: 8.5958 - mae: 2.2800 - val_loss: 9.4252 - val_mae: 2.
Epoch 9/10
819/819 [==============================] - 94s 115ms/step - loss: 8.4724 - mae: 2.2593 - val_loss: 9.3826 - val_mae: 2.3
Epoch 10/10
819/819 [==============================] - 92s 112ms/step - loss: 8.3591 - mae: 2.2450 - val_loss: 9.5085 - val_mae: 2.3
```

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

```
405/405 [==============================] - 22s 54ms/step - loss: 11.1936 - mae: 2.6500
Test MAE: 2.65
```

```
import numpy as np
timesteps = 100
input_features = 32
output_features = 64
inputs = 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_model_outputs     = []
for input_t in inputs:
    output_t = np.tanh(np.dot(W, input_t) + np.dot(U, state_t) + b)
    successive_model_outputs    .append(output_t)
    state_t = output_t
final_output_sequence = np.stack(successive_model_outputs   , axis=0)
#Initializing a basic RNN using Keras
#RNN layer configured for variable-length input sequences
```

```
num_features = 14
inputs = keras.Input(shape=(None, num_features))
model_outputs    = layers.SimpleRNN(16)(inputs)
#RNN layer that outputs only the final timestep
```

```
num_features = 14
steps = 120
inputs = keras.Input(shape=(steps, num_features))
model_outputs    = layers.SimpleRNN(16, return_sequences=False)(inputs)
print(model_outputs .shape)
#RNN that provides an output for each input time step
```

```
(None, 16)
```

```
num_features = 14
steps = 120
inputs = keras.Input(shape=(steps, num_features))
model_outputs    = layers.SimpleRNN(16, return_sequences=True)(inputs)
print(model_outputs .shape)
#Stacking RNN layers
```

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

```
inputs = keras.Input(shape=(steps, num_features))
x = layers.SimpleRNN(16, return_sequences=True)(inputs)
x = layers.SimpleRNN(16, return_sequences=True)(x)
model_outputs    = layers.SimpleRNN(16)(x)
#Implementing advanced strategies with recurrent neural networks
# Using recurrent dropout to reduce overfitting in RNNs
#Training and evaluating an LSTM model with dropout
```

```
inputs = keras.Input(shape=(seq_len, original_file_path.shape[-1]))
x = layers.LSTM(32, recurrent_dropout=0.25)(inputs)
x = layers.Dropout(0.5)(x)
```

```
model_outputs    = layers.Dense(1)(x)
model = keras.Model(inputs, model_outputs   )


callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_file_pathset,
                    epochs=10,
                    validation_data=val_file_pathset,
                    callbacks=callbacks)
inputs = keras.Input(shape=(seq_len, num_features))
x = layers.LSTM(32, recurrent_dropout=0.2, unroll=True)(inputs)
```

```
Epoch 1/10
819/819 [==============================] - 205s 247ms/step - loss: 27.8442 - mae: 3.9003 - val_loss: 9.9011 - val_mae: 2
Epoch 2/10
819/819 [==============================] - 197s 241ms/step - loss: 14.8935 - mae: 2.9958 - val_loss: 9.6251 - val_mae: 2
Epoch 3/10
819/819 [==============================] - 197s 240ms/step - loss: 14.0237 - mae: 2.9047 - val_loss: 9.2187 - val_mae: 2
Epoch 4/10
819/819 [==============================] - 197s 240ms/step - loss: 13.4169 - mae: 2.8417 - val_loss: 9.1934 - val_mae: 2
Epoch 5/10
819/819 [==============================] - 195s 238ms/step - loss: 12.8983 - mae: 2.7864 - val_loss: 9.6893 - val_mae: 2
Epoch 6/10
819/819 [==============================] - 197s 241ms/step - loss: 12.3605 - mae: 2.7273 - val_loss: 9.3112 - val_mae: 2
Epoch 7/10
819/819 [==============================] - 197s 240ms/step - loss: 12.0473 - mae: 2.6941 - val_loss: 9.3193 - val_mae: 2
Epoch 8/10
819/819 [==============================] - 197s 240ms/step - loss: 11.8512 - mae: 2.6708 - val_loss: 9.3436 - val_mae: 2
Epoch 9/10
819/819 [==============================] - 199s 243ms/step - loss: 11.6496 - mae: 2.6467 - val_loss: 9.5948 - val_mae: 2
Epoch 10/10
819/819 [==============================] - 198s 242ms/step - loss: 11.4230 - mae: 2.6220 - val_loss: 9.4550 - val_mae: 2
```

```
inputs = keras.Input(shape=(seq_len, original_file_path.shape[-1]))
x = layers.GRU(32, recurrent_dropout=0.5, return_sequences=True)(inputs)
x = layers.GRU(32, recurrent_dropout=0.5)(x)
x = layers.Dropout(0.5)(x)
model_outputs  = layers.Dense(1)(x)
model = keras.Model(inputs, model_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.

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.

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.