- Build CNN Model for Classification Of ***Flowers***

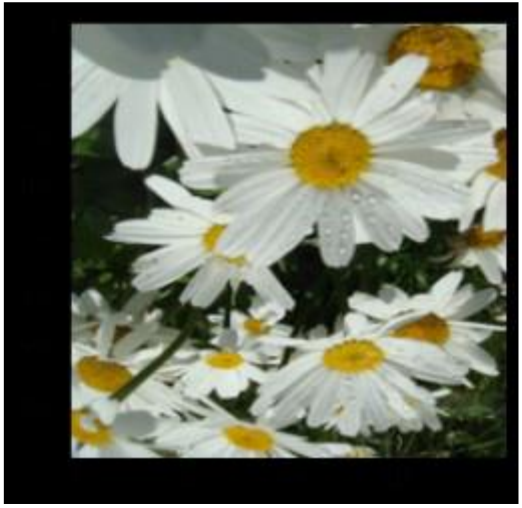
Download the Data Set

Image Augmentation

```
import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import random
from skimage import exposure
from skimage.util import random_noise
from skimage import transform
from cv2 import resize
img=mpimg.imread("/content/1354396826_2868631432_m.jpg")
plt.imshow(img)
img_rescale=resize(img,(400,400))
plt.imshow(img_rescale)
<matplotlib.image.AxesImage at 0x7f161445f610>
```

```
#horizontal flip
horiz=np.fliplr(img_rescale)
plt.imshow(horiz)
<matplotlib.image.AxesImage at 0x7f161274c1d0>
```

```
#vertical flip
vert=np.flipud(img_rescale)
plt.imshow(vert)
<matplotlib.image.AxesImage at 0x7f1612735e10>
```



```
#rotate noise
img_nos=random_noise(img_rescale,mode='s&p',clip=True)
plt.imshow(image_nos)
mpimg.imsave("noise_flower",img_nos)
---------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-9-ef93c8e43a49> in <module>
2
3 img_nos=random_noise(img_rescale,mode='s&p',clip=True)
----> 4 plt.imshow(image_nos)
5 mpimg.imsave("noise_flower",img_nos)
NameError: name 'image_nos' is not defined
```

Create Model Using CNN

```
import tensorflow as tf
tf.__version__
```

{"type":"string"}

```
!pip install --upgrade tensorflow
Looking in indexes: https://pypi.org/simple, https://uspython.
pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.7/dist-packages (2.9.2)
Collecting tensorflow
Downloading tensorflow-2.10.0-cp37-cp37mmanylinux_
2_17_x86_64.manylinux2014_x86_64.whl (578.0 MB)
ent already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/distpackages
(from tensorflow) (1.15.0)
Collecting keras<2.11,>=2.10.0
Downloading keras-2.10.0-py2.py3-none-any.whl (1.7 MB)
ent already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)
Collecting flatbuffers>=2.0
Downloading flatbuffers-22.10.26-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.27.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (21.3)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: keras-preprocessing>=1.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.3.0)
Collecting tensorflow-estimator<2.11,>=2.10.0
Downloading tensorflow_estimator-2.10.0-py2.py3-none-any.whl (438
kB)
ent already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (2.0.1)
Collecting tensorboard<2.11,>=2.10
Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)
ent already satisfied: protobuf<3.20,>=3.9.2 in
```

```
/usr/local/lib/python3.7/dist-packages (from tensorflow) (3.17.3)
Requirement already satisfied: wrapt>=1.11.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (57.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.37.1)
Requirement already satisfied: cached-property in
/usr/local/lib/python3.7/dist-packages (from h5py>=2.9.0->tensorflow)
(1.5.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (1.35.0)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (2.23.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10-
>tensorflow) (1.0.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /usr/local/lib/python3.7/dist-packages (from
tensorboard<2.11,>=2.10->tensorflow) (0.6.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.11,>=2.10->tensorflow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.11,>=2.10->tensorflow) (4.9)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.11,>=2.10->tensorflow) (4.2.4)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.7/dist-packages (from google-authoauthlib<
0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (1.3.1)
```

```
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8-
>tensorboard<2.11,>=2.10->tensorflow) (4.13.0)
Requirement already satisfied: zipp>=0.5 in
/usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4-
>markdown>=2.6.8->tensorboard<2.11,>=2.10->tensorflow) (3.9.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.11,>=2.10->tensorflow) (2022.9.24)
Requirement already satisfied: idna<3,>=2.5 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.11,>=2.10->tensorflow) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.11,>=2.10->tensorflow) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.11,>=2.10->tensorflow) (3.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0-
>google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10-
>tensorflow) (3.2.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->tensorflow)
(3.0.9)
Installing collected packages: tensorflow-estimator, tensorboard,
keras, flatbuffers, tensorflow
Attempting uninstall: tensorflow-estimator
Found existing installation: tensorflow-estimator 2.9.0
Uninstalling tensorflow-estimator-2.9.0:
Successfully uninstalled tensorflow-estimator-2.9.0
Attempting uninstall: tensorboard
Found existing installation: tensorboard 2.9.1
Uninstalling tensorboard-2.9.1:
Successfully uninstalled tensorboard-2.9.1
Attempting uninstall: keras
Found existing installation: keras 2.9.0
Uninstalling keras-2.9.0:
Successfully uninstalled keras-2.9.0
Attempting uninstall: flatbuffers
Found existing installation: flatbuffers 1.12
Uninstalling flatbuffers-1.12:
Successfully uninstalled flatbuffers-1.12
Attempting uninstall: tensorflow
Found existing installation: tensorflow 2.9.2
```

```
Uninstalling tensorflow-2.9.2:
Successfully uninstalled tensorflow-2.9.2
Successfully installed flatbuffers-22.10.26 keras-2.10.0 tensorboard-
2.10.1 tensorflow-2.10.0 tensorflow-estimator-2.10.0
{"pip_warning":{"packages":
["flatbuffers","keras","tensorboard","tensorflow"]}}
import tensorflow
from tensorflow.keras.layers import
Dense,Flatten,Conv2D,MaxPool2D,Dropout
from tensorflow.keras import Model
class MyModel(Model)
def __init__(self):
super(MyModel,self).__init__()
self.conv1=Conv2D(32,3,padding='same',activation='relu')
self.pool1=MaxPool2D((2,2))
self.conv2=Conv2D(64,3,padding='same',actiavtion='relu')
self.pool2=MaxPool2D((2,2))
self.flatten=Flatten()
self.d1=Dense(512,activation='relu')
self.droupout1=Dropout(0.4)
self.d2=Dense(128,activation='relu')
self.dropout2=Dropout(0.4)
self.d3=Dense(43,activation='softmax')
File "<ipython-input-2-b39be2e3b9a6>", line 1
class MyModel(Model)
^
SyntaxError: invalid syntax
def call(self,x):
x=self.conv1(x)
x=self.pool1(x)
x=self.conv2(x)
x=self.pool2(x)
x=self.flatten(x)
x=self.d1(x)
x=self.droupout1(x)
x=self.d2(x)
x=self.dropout2(x)
x=self.d3(x)
return x
model=MyModel()
----------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-3-32515b4edb19> in <module>
12 return x
13
---> 14 model=MyModel()
```

```
NameError: name 'MyModel' is not defined
```

Add Layers

```
#add dense layer
#importing the required libraries
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
#loading data
(X_train,y_train) , (X_test,y_test)=mnist.load_data()
#reshaping data
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1],
X_train.shape[2], 1))
X_test =
X_test.reshape((X_test.shape[0],X_test.shape[1],X_test.shape[2],1))
#checking the shape after reshaping
print(X_train.shape)
print(X_test.shape)
#normalizing the pixel values
X_train=X_train/255
X_test=X_test/255
Downloading data from https://storage.googleapis.com/tensorflow/tfkeras-
datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
(60000, 28, 28, 1)
(10000, 28, 28, 1)
#defining model
model=Sequential()
#adding convolution layer
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
#adding pooling layer
model.add(MaxPool2D(2,2))
#adding fully connected layer
model.add(Flatten())
model.add(Dense(100,activation='relu'))
#adding output layer
model.add(Dense(10,activation='softmax'))
#compiling the model
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',
metrics=['accuracy'])
#fitting the model
model.fit(X_train,y_train,epochs=10)
Epoch 1/10
1875/1875 [==============================] - 14s 3ms/step - loss:
0.1618 - accuracy: 0.9523
```

```
Epoch 2/10
1875/1875 [==============================] - 6s 3ms/step - loss:
0.0563 - accuracy: 0.9827
Epoch 3/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0364 - accuracy: 0.9884
Epoch 4/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0228 - accuracy: 0.9927
Epoch 5/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0171 - accuracy: 0.9946
Epoch 6/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0113 - accuracy: 0.9964
Epoch 7/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0080 - accuracy: 0.9976
Epoch 8/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0069 - accuracy: 0.9979
Epoch 9/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0051 - accuracy: 0.9985
Epoch 10/10
1875/1875 [==============================] - 5s 3ms/step - loss:
0.0049 - accuracy: 0.9982
<keras.callbacks.History at 0x7f15a00b3350>
```

Compile the model

Metrices

```
from numpy import array
from keras.models import Sequential
from keras.layers import Dense
from matplotlib import pyplot
# prepare sequence
X = array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
y = array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])
# create model
model = Sequential()
model.add(Dense(2, input_dim=1))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
# train model
history = model.fit(X, y, epochs=400, batch_size=len(X), verbose=2)
# plot metrics
pyplot.plot(history.history['accuracy'])
pyplot.show()
```

```
Epoch 1/400
1/1 - 0s - loss: 0.6078 - accuracy: 0.5000 - 355ms/epoch - 355ms/step
Epoch 2/400
1/1 - 0s - loss: 0.6073 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 3/400
1/1 - 0s - loss: 0.6068 - accuracy: 0.5000 - 9ms/epoch - 9ms/step
Epoch 4/400
1/1 - 0s - loss: 0.6063 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 5/400
1/1 - 0s - loss: 0.6057 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
Epoch 6/400
1/1 - 0s - loss: 0.6052 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 7/400
1/1 - 0s - loss: 0.6047 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 8/400
1/1 - 0s - loss: 0.6042 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 9/400
1/1 - 0s - loss: 0.6037 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 10/400
1/1 - 0s - loss: 0.6032 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 11/400
1/1 - 0s - loss: 0.6027 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 12/400
1/1 - 0s - loss: 0.6021 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 13/400
1/1 - 0s - loss: 0.6016 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 14/400
1/1 - 0s - loss: 0.6011 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 15/400
1/1 - 0s - loss: 0.6006 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 16/400
1/1 - 0s - loss: 0.6001 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 17/400
1/1 - 0s - loss: 0.5996 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 18/400
1/1 - 0s - loss: 0.5990 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 19/400
1/1 - 0s - loss: 0.5985 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 20/400
1/1 - 0s - loss: 0.5980 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 21/400
1/1 - 0s - loss: 0.5975 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 22/400
1/1 - 0s - loss: 0.5970 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 23/400
1/1 - 0s - loss: 0.5964 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 24/400
1/1 - 0s - loss: 0.5959 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
```

```
Epoch 25/400
1/1 - 0s - loss: 0.5954 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 26/400
1/1 - 0s - loss: 0.5949 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 27/400
1/1 - 0s - loss: 0.5943 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
Epoch 28/400
1/1 - 0s - loss: 0.5938 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 29/400
1/1 - 0s - loss: 0.5933 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 30/400
1/1 - 0s - loss: 0.5928 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 31/400
1/1 - 0s - loss: 0.5922 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 32/400
1/1 - 0s - loss: 0.5917 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
Epoch 33/400
1/1 - 0s - loss: 0.5912 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 34/400
1/1 - 0s - loss: 0.5907 - accuracy: 0.5000 - 5ms/epoch - 5ms/step
Epoch 35/400
1/1 - 0s - loss: 0.5901 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 36/400
1/1 - 0s - loss: 0.5896 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 37/400
1/1 - 0s - loss: 0.5891 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 38/400
1/1 - 0s - loss: 0.5886 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 39/400
1/1 - 0s - loss: 0.5880 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
Epoch 40/400
1/1 - 0s - loss: 0.5875 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 41/400
1/1 - 0s - loss: 0.5870 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 42/400
1/1 - 0s - loss: 0.5864 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 43/400
1/1 - 0s - loss: 0.5859 - accuracy: 0.5000 - 8ms/epoch - 8ms/step
Epoch 44/400
1/1 - 0s - loss: 0.5854 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 45/400
1/1 - 0s - loss: 0.5848 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 46/400
1/1 - 0s - loss: 0.5843 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 47/400
1/1 - 0s - loss: 0.5838 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 48/400
1/1 - 0s - loss: 0.5832 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
```

```
Epoch 49/400
1/1 - 0s - loss: 0.5827 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 50/400
1/1 - 0s - loss: 0.5822 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 51/400
1/1 - 0s - loss: 0.5816 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 52/400
1/1 - 0s - loss: 0.5811 - accuracy: 0.5000 - 7ms/epoch - 7ms/step
Epoch 53/400
1/1 - 0s - loss: 0.5805 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 54/400
1/1 - 0s - loss: 0.5800 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 55/400
1/1 - 0s - loss: 0.5795 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 56/400
1/1 - 0s - loss: 0.5789 - accuracy: 0.5000 - 6ms/epoch - 6ms/step
Epoch 57/400
1/1 - 0s - loss: 0.5784 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 58/400
1/1 - 0s - loss: 0.5779 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 59/400
1/1 - 0s - loss: 0.5773 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 60/400
1/1 - 0s - loss: 0.5768 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 61/400
1/1 - 0s - loss: 0.5762 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 62/400
1/1 - 0s - loss: 0.5757 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 63/400
1/1 - 0s - loss: 0.5751 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 64/400
1/1 - 0s - loss: 0.5746 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 65/400
1/1 - 0s - loss: 0.5740 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 66/400
1/1 - 0s - loss: 0.5735 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 67/400
1/1 - 0s - loss: 0.5730 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 68/400
1/1 - 0s - loss: 0.5724 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 69/400
1/1 - 0s - loss: 0.5719 - accuracy: 0.6000 - 8ms/epoch - 8ms/step
Epoch 70/400
1/1 - 0s - loss: 0.5713 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 71/400
1/1 - 0s - loss: 0.5708 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 72/400
1/1 - 0s - loss: 0.5702 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
```

```
Epoch 73/400
1/1 - 0s - loss: 0.5697 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 74/400
1/1 - 0s - loss: 0.5691 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 75/400
1/1 - 0s - loss: 0.5686 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 76/400
1/1 - 0s - loss: 0.5680 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 77/400
1/1 - 0s - loss: 0.5675 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 78/400
1/1 - 0s - loss: 0.5669 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 79/400
1/1 - 0s - loss: 0.5664 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 80/400
1/1 - 0s - loss: 0.5658 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 81/400
1/1 - 0s - loss: 0.5653 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 82/400
1/1 - 0s - loss: 0.5647 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 83/400
1/1 - 0s - loss: 0.5642 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 84/400
1/1 - 0s - loss: 0.5636 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 85/400
1/1 - 0s - loss: 0.5630 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 86/400
1/1 - 0s - loss: 0.5625 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 87/400
1/1 - 0s - loss: 0.5619 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 88/400
1/1 - 0s - loss: 0.5614 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 89/400
1/1 - 0s - loss: 0.5608 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 90/400
1/1 - 0s - loss: 0.5603 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 91/400
1/1 - 0s - loss: 0.5597 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 92/400
1/1 - 0s - loss: 0.5591 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 93/400
1/1 - 0s - loss: 0.5586 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 94/400
1/1 - 0s - loss: 0.5580 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 95/400
1/1 - 0s - loss: 0.5575 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 96/400
1/1 - 0s - loss: 0.5569 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
```

```
Epoch 97/400
1/1 - 0s - loss: 0.5563 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 98/400
1/1 - 0s - loss: 0.5558 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 99/400
1/1 - 0s - loss: 0.5552 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 100/400
1/1 - 0s - loss: 0.5546 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 101/400
1/1 - 0s - loss: 0.5541 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 102/400
1/1 - 0s - loss: 0.5535 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 103/400
1/1 - 0s - loss: 0.5530 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 104/400
1/1 - 0s - loss: 0.5524 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 105/400
1/1 - 0s - loss: 0.5518 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 106/400
1/1 - 0s - loss: 0.5513 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 107/400
1/1 - 0s - loss: 0.5507 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 108/400
1/1 - 0s - loss: 0.5501 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 109/400
1/1 - 0s - loss: 0.5496 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 110/400
1/1 - 0s - loss: 0.5490 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 111/400
1/1 - 0s - loss: 0.5484 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 112/400
1/1 - 0s - loss: 0.5479 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 113/400
1/1 - 0s - loss: 0.5473 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 114/400
1/1 - 0s - loss: 0.5467 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 115/400
1/1 - 0s - loss: 0.5461 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
Epoch 116/400
1/1 - 0s - loss: 0.5456 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 117/400
1/1 - 0s - loss: 0.5450 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 118/400
1/1 - 0s - loss: 0.5444 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 119/400
1/1 - 0s - loss: 0.5439 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 120/400
1/1 - 0s - loss: 0.5433 - accuracy: 0.6000 - 5ms/epoch - 5ms/step
```

```
Epoch 121/400
1/1 - 0s - loss: 0.5427 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 122/400
1/1 - 0s - loss: 0.5422 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 123/400
1/1 - 0s - loss: 0.5416 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 124/400
1/1 - 0s - loss: 0.5410 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 125/400
1/1 - 0s - loss: 0.5404 - accuracy: 0.6000 - 7ms/epoch - 7ms/step
Epoch 126/400
1/1 - 0s - loss: 0.5399 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 127/400
1/1 - 0s - loss: 0.5393 - accuracy: 0.6000 - 6ms/epoch - 6ms/step
Epoch 128/400
1/1 - 0s - loss: 0.5387 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 129/400
1/1 - 0s - loss: 0.5381 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 130/400
1/1 - 0s - loss: 0.5376 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 131/400
1/1 - 0s - loss: 0.5370 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 132/400
1/1 - 0s - loss: 0.5364 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 133/400
1/1 - 0s - loss: 0.5358 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 134/400
1/1 - 0s - loss: 0.5353 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 135/400
1/1 - 0s - loss: 0.5347 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 136/400
1/1 - 0s - loss: 0.5341 - accuracy: 0.7000 - 11ms/epoch - 11ms/step
Epoch 137/400
1/1 - 0s - loss: 0.5335 - accuracy: 0.7000 - 9ms/epoch - 9ms/step
Epoch 138/400
1/1 - 0s - loss: 0.5330 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 139/400
1/1 - 0s - loss: 0.5324 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 140/400
1/1 - 0s - loss: 0.5318 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 141/400
1/1 - 0s - loss: 0.5312 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 142/400
1/1 - 0s - loss: 0.5306 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 143/400
1/1 - 0s - loss: 0.5301 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 144/400
1/1 - 0s - loss: 0.5295 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
```

```
Epoch 145/400
1/1 - 0s - loss: 0.5289 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 146/400
1/1 - 0s - loss: 0.5283 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 147/400
1/1 - 0s - loss: 0.5277 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 148/400
1/1 - 0s - loss: 0.5272 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 149/400
1/1 - 0s - loss: 0.5266 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 150/400
1/1 - 0s - loss: 0.5260 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 151/400
1/1 - 0s - loss: 0.5254 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 152/400
1/1 - 0s - loss: 0.5248 - accuracy: 0.7000 - 10ms/epoch - 10ms/step
Epoch 153/400
1/1 - 0s - loss: 0.5243 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 154/400
1/1 - 0s - loss: 0.5237 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 155/400
1/1 - 0s - loss: 0.5231 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 156/400
1/1 - 0s - loss: 0.5225 - accuracy: 0.7000 - 4ms/epoch - 4ms/step
Epoch 157/400
1/1 - 0s - loss: 0.5219 - accuracy: 0.7000 - 4ms/epoch - 4ms/step
Epoch 158/400
1/1 - 0s - loss: 0.5214 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 159/400
1/1 - 0s - loss: 0.5208 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 160/400
1/1 - 0s - loss: 0.5202 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 161/400
1/1 - 0s - loss: 0.5196 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 162/400
1/1 - 0s - loss: 0.5190 - accuracy: 0.7000 - 4ms/epoch - 4ms/step
Epoch 163/400
1/1 - 0s - loss: 0.5184 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 164/400
1/1 - 0s - loss: 0.5179 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 165/400
1/1 - 0s - loss: 0.5173 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 166/400
1/1 - 0s - loss: 0.5167 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 167/400
1/1 - 0s - loss: 0.5161 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 168/400
1/1 - 0s - loss: 0.5155 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
```

```
Epoch 169/400
1/1 - 0s - loss: 0.5149 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 170/400
1/1 - 0s - loss: 0.5143 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 171/400
1/1 - 0s - loss: 0.5138 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 172/400
1/1 - 0s - loss: 0.5132 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 173/400
1/1 - 0s - loss: 0.5126 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 174/400
1/1 - 0s - loss: 0.5120 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 175/400
1/1 - 0s - loss: 0.5114 - accuracy: 0.7000 - 12ms/epoch - 12ms/step
Epoch 176/400
1/1 - 0s - loss: 0.5108 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 177/400
1/1 - 0s - loss: 0.5103 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 178/400
1/1 - 0s - loss: 0.5097 - accuracy: 0.7000 - 12ms/epoch - 12ms/step
Epoch 179/400
1/1 - 0s - loss: 0.5091 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 180/400
1/1 - 0s - loss: 0.5085 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 181/400
1/1 - 0s - loss: 0.5079 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 182/400
1/1 - 0s - loss: 0.5073 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 183/400
1/1 - 0s - loss: 0.5067 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 184/400
1/1 - 0s - loss: 0.5062 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 185/400
1/1 - 0s - loss: 0.5056 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 186/400
1/1 - 0s - loss: 0.5050 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 187/400
1/1 - 0s - loss: 0.5044 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 188/400
1/1 - 0s - loss: 0.5038 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 189/400
1/1 - 0s - loss: 0.5032 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 190/400
1/1 - 0s - loss: 0.5026 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 191/400
1/1 - 0s - loss: 0.5020 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 192/400
1/1 - 0s - loss: 0.5015 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
```

```
Epoch 193/400
1/1 - 0s - loss: 0.5009 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 194/400
1/1 - 0s - loss: 0.5003 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 195/400
1/1 - 0s - loss: 0.4997 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 196/400
1/1 - 0s - loss: 0.4991 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 197/400
1/1 - 0s - loss: 0.4985 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 198/400
1/1 - 0s - loss: 0.4979 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 199/400
1/1 - 0s - loss: 0.4974 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 200/400
1/1 - 0s - loss: 0.4968 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 201/400
1/1 - 0s - loss: 0.4962 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 202/400
1/1 - 0s - loss: 0.4956 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 203/400
1/1 - 0s - loss: 0.4950 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 204/400
1/1 - 0s - loss: 0.4944 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 205/400
1/1 - 0s - loss: 0.4938 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 206/400
1/1 - 0s - loss: 0.4932 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 207/400
1/1 - 0s - loss: 0.4927 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 208/400
1/1 - 0s - loss: 0.4921 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 209/400
1/1 - 0s - loss: 0.4915 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 210/400
1/1 - 0s - loss: 0.4909 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 211/400
1/1 - 0s - loss: 0.4903 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 212/400
1/1 - 0s - loss: 0.4897 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 213/400
1/1 - 0s - loss: 0.4891 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 214/400
1/1 - 0s - loss: 0.4886 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 215/400
1/1 - 0s - loss: 0.4880 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 216/400
1/1 - 0s - loss: 0.4874 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
```

```
Epoch 217/400
1/1 - 0s - loss: 0.4868 - accuracy: 0.7000 - 8ms/epoch - 8ms/step
Epoch 218/400
1/1 - 0s - loss: 0.4862 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 219/400
1/1 - 0s - loss: 0.4856 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 220/400
1/1 - 0s - loss: 0.4850 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 221/400
1/1 - 0s - loss: 0.4845 - accuracy: 0.7000 - 7ms/epoch - 7ms/step
Epoch 222/400
1/1 - 0s - loss: 0.4839 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 223/400
1/1 - 0s - loss: 0.4833 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 224/400
1/1 - 0s - loss: 0.4827 - accuracy: 0.7000 - 6ms/epoch - 6ms/step
Epoch 225/400
1/1 - 0s - loss: 0.4821 - accuracy: 0.7000 - 5ms/epoch - 5ms/step
Epoch 226/400
1/1 - 0s - loss: 0.4815 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 227/400
1/1 - 0s - loss: 0.4809 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 228/400
1/1 - 0s - loss: 0.4804 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 229/400
1/1 - 0s - loss: 0.4798 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 230/400
1/1 - 0s - loss: 0.4792 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 231/400
1/1 - 0s - loss: 0.4786 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 232/400
1/1 - 0s - loss: 0.4780 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 233/400
1/1 - 0s - loss: 0.4774 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 234/400
1/1 - 0s - loss: 0.4769 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 235/400
1/1 - 0s - loss: 0.4763 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 236/400
1/1 - 0s - loss: 0.4757 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 237/400
1/1 - 0s - loss: 0.4751 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 238/400
1/1 - 0s - loss: 0.4745 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 239/400
1/1 - 0s - loss: 0.4739 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 240/400
1/1 - 0s - loss: 0.4734 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
```

```
Epoch 241/400
1/1 - 0s - loss: 0.4728 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 242/400
1/1 - 0s - loss: 0.4722 - accuracy: 0.8000 - 8ms/epoch - 8ms/step
Epoch 243/400
1/1 - 0s - loss: 0.4716 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 244/400
1/1 - 0s - loss: 0.4710 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 245/400
1/1 - 0s - loss: 0.4704 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 246/400
1/1 - 0s - loss: 0.4699 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 247/400
1/1 - 0s - loss: 0.4693 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 248/400
1/1 - 0s - loss: 0.4687 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 249/400
1/1 - 0s - loss: 0.4681 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 250/400
1/1 - 0s - loss: 0.4675 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 251/400
1/1 - 0s - loss: 0.4669 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 252/400
1/1 - 0s - loss: 0.4664 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 253/400
1/1 - 0s - loss: 0.4658 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 254/400
1/1 - 0s - loss: 0.4652 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 255/400
1/1 - 0s - loss: 0.4646 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 256/400
1/1 - 0s - loss: 0.4640 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 257/400
1/1 - 0s - loss: 0.4635 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 258/400
1/1 - 0s - loss: 0.4629 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 259/400
1/1 - 0s - loss: 0.4623 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 260/400
1/1 - 0s - loss: 0.4617 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 261/400
1/1 - 0s - loss: 0.4612 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 262/400
1/1 - 0s - loss: 0.4606 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 263/400
1/1 - 0s - loss: 0.4600 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 264/400
1/1 - 0s - loss: 0.4594 - accuracy: 0.8000 - 8ms/epoch - 8ms/step
```

```
Epoch 265/400
1/1 - 0s - loss: 0.4588 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 266/400
1/1 - 0s - loss: 0.4583 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 267/400
1/1 - 0s - loss: 0.4577 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 268/400
1/1 - 0s - loss: 0.4571 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 269/400
1/1 - 0s - loss: 0.4565 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 270/400
1/1 - 0s - loss: 0.4560 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 271/400
1/1 - 0s - loss: 0.4554 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 272/400
1/1 - 0s - loss: 0.4548 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 273/400
1/1 - 0s - loss: 0.4542 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 274/400
1/1 - 0s - loss: 0.4537 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 275/400
1/1 - 0s - loss: 0.4531 - accuracy: 0.8000 - 12ms/epoch - 12ms/step
Epoch 276/400
1/1 - 0s - loss: 0.4525 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 277/400
1/1 - 0s - loss: 0.4519 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 278/400
1/1 - 0s - loss: 0.4514 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 279/400
1/1 - 0s - loss: 0.4508 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 280/400
1/1 - 0s - loss: 0.4502 - accuracy: 0.8000 - 9ms/epoch - 9ms/step
Epoch 281/400
1/1 - 0s - loss: 0.4496 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 282/400
1/1 - 0s - loss: 0.4491 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 283/400
1/1 - 0s - loss: 0.4485 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 284/400
1/1 - 0s - loss: 0.4479 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 285/400
1/1 - 0s - loss: 0.4474 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 286/400
1/1 - 0s - loss: 0.4468 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 287/400
1/1 - 0s - loss: 0.4462 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 288/400
1/1 - 0s - loss: 0.4456 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
```

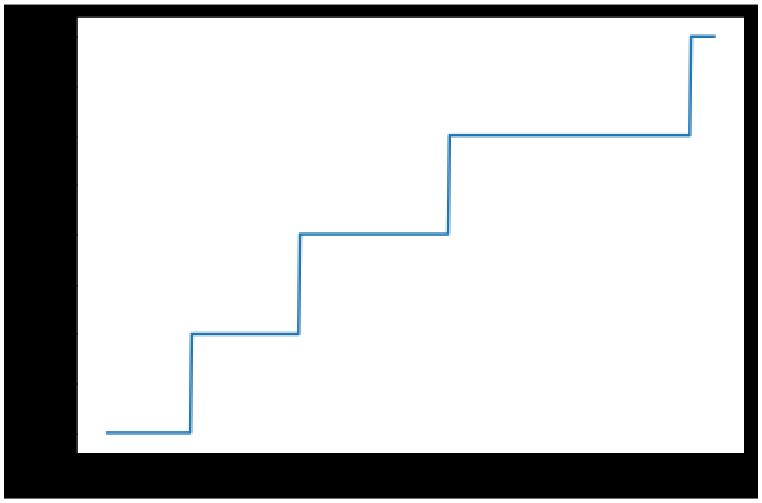
```
Epoch 289/400
1/1 - 0s - loss: 0.4451 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 290/400
1/1 - 0s - loss: 0.4445 - accuracy: 0.8000 - 8ms/epoch - 8ms/step
Epoch 291/400
1/1 - 0s - loss: 0.4439 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 292/400
1/1 - 0s - loss: 0.4434 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 293/400
1/1 - 0s - loss: 0.4428 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 294/400
1/1 - 0s - loss: 0.4422 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 295/400
1/1 - 0s - loss: 0.4417 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 296/400
1/1 - 0s - loss: 0.4411 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 297/400
1/1 - 0s - loss: 0.4405 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 298/400
1/1 - 0s - loss: 0.4400 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 299/400
1/1 - 0s - loss: 0.4394 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 300/400
1/1 - 0s - loss: 0.4388 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 301/400
1/1 - 0s - loss: 0.4383 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 302/400
1/1 - 0s - loss: 0.4377 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 303/400
1/1 - 0s - loss: 0.4371 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 304/400
1/1 - 0s - loss: 0.4366 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 305/400
1/1 - 0s - loss: 0.4360 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 306/400
1/1 - 0s - loss: 0.4355 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 307/400
1/1 - 0s - loss: 0.4349 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 308/400
1/1 - 0s - loss: 0.4343 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 309/400
1/1 - 0s - loss: 0.4338 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 310/400
1/1 - 0s - loss: 0.4332 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 311/400
1/1 - 0s - loss: 0.4327 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 312/400
1/1 - 0s - loss: 0.4321 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
```

```
Epoch 313/400
1/1 - 0s - loss: 0.4315 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 314/400
1/1 - 0s - loss: 0.4310 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 315/400
1/1 - 0s - loss: 0.4304 - accuracy: 0.8000 - 8ms/epoch - 8ms/step
Epoch 316/400
1/1 - 0s - loss: 0.4299 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 317/400
1/1 - 0s - loss: 0.4293 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 318/400
1/1 - 0s - loss: 0.4287 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 319/400
1/1 - 0s - loss: 0.4282 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 320/400
1/1 - 0s - loss: 0.4276 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 321/400
1/1 - 0s - loss: 0.4271 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 322/400
1/1 - 0s - loss: 0.4265 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 323/400
1/1 - 0s - loss: 0.4260 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 324/400
1/1 - 0s - loss: 0.4254 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 325/400
1/1 - 0s - loss: 0.4249 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 326/400
1/1 - 0s - loss: 0.4243 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 327/400
1/1 - 0s - loss: 0.4237 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 328/400
1/1 - 0s - loss: 0.4232 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 329/400
1/1 - 0s - loss: 0.4226 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 330/400
1/1 - 0s - loss: 0.4221 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 331/400
1/1 - 0s - loss: 0.4215 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 332/400
1/1 - 0s - loss: 0.4210 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 333/400
1/1 - 0s - loss: 0.4204 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 334/400
1/1 - 0s - loss: 0.4199 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 335/400
1/1 - 0s - loss: 0.4193 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 336/400
1/1 - 0s - loss: 0.4188 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
```

```
Epoch 337/400
1/1 - 0s - loss: 0.4182 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 338/400
1/1 - 0s - loss: 0.4177 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 339/400
1/1 - 0s - loss: 0.4172 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 340/400
1/1 - 0s - loss: 0.4166 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 341/400
1/1 - 0s - loss: 0.4161 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 342/400
1/1 - 0s - loss: 0.4155 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 343/400
1/1 - 0s - loss: 0.4150 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 344/400
1/1 - 0s - loss: 0.4144 - accuracy: 0.8000 - 11ms/epoch - 11ms/step
Epoch 345/400
1/1 - 0s - loss: 0.4139 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 346/400
1/1 - 0s - loss: 0.4133 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 347/400
1/1 - 0s - loss: 0.4128 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 348/400
1/1 - 0s - loss: 0.4123 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 349/400
1/1 - 0s - loss: 0.4117 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 350/400
1/1 - 0s - loss: 0.4112 - accuracy: 0.8000 - 5ms/epoch - 5ms/step
Epoch 351/400
1/1 - 0s - loss: 0.4106 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 352/400
1/1 - 0s - loss: 0.4101 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 353/400
1/1 - 0s - loss: 0.4096 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 354/400
1/1 - 0s - loss: 0.4090 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 355/400
1/1 - 0s - loss: 0.4085 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 356/400
1/1 - 0s - loss: 0.4079 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 357/400
1/1 - 0s - loss: 0.4074 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 358/400
1/1 - 0s - loss: 0.4069 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 359/400
1/1 - 0s - loss: 0.4063 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 360/400
1/1 - 0s - loss: 0.4058 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
```

```
Epoch 361/400
1/1 - 0s - loss: 0.4053 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 362/400
1/1 - 0s - loss: 0.4047 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 363/400
1/1 - 0s - loss: 0.4042 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 364/400
1/1 - 0s - loss: 0.4037 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 365/400
1/1 - 0s - loss: 0.4031 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 366/400
1/1 - 0s - loss: 0.4026 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 367/400
1/1 - 0s - loss: 0.4021 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 368/400
1/1 - 0s - loss: 0.4015 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 369/400
1/1 - 0s - loss: 0.4010 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 370/400
1/1 - 0s - loss: 0.4005 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 371/400
1/1 - 0s - loss: 0.4000 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 372/400
1/1 - 0s - loss: 0.3994 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 373/400
1/1 - 0s - loss: 0.3989 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 374/400
1/1 - 0s - loss: 0.3984 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 375/400
1/1 - 0s - loss: 0.3978 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 376/400
1/1 - 0s - loss: 0.3973 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 377/400
1/1 - 0s - loss: 0.3968 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 378/400
1/1 - 0s - loss: 0.3963 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 379/400
1/1 - 0s - loss: 0.3957 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 380/400
1/1 - 0s - loss: 0.3952 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 381/400
1/1 - 0s - loss: 0.3947 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 382/400
1/1 - 0s - loss: 0.3942 - accuracy: 0.8000 - 6ms/epoch - 6ms/step
Epoch 383/400
1/1 - 0s - loss: 0.3937 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
Epoch 384/400
1/1 - 0s - loss: 0.3931 - accuracy: 0.8000 - 7ms/epoch - 7ms/step
```

```
Epoch 385/400
1/1 - 0s - loss: 0.3926 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 386/400
1/1 - 0s - loss: 0.3921 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 387/400
1/1 - 0s - loss: 0.3916 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 388/400
1/1 - 0s - loss: 0.3911 - accuracy: 0.9000 - 8ms/epoch - 8ms/step
Epoch 389/400
1/1 - 0s - loss: 0.3905 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 390/400
1/1 - 0s - loss: 0.3900 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 391/400
1/1 - 0s - loss: 0.3895 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 392/400
1/1 - 0s - loss: 0.3890 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 393/400
1/1 - 0s - loss: 0.3885 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 394/400
1/1 - 0s - loss: 0.3880 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 395/400
1/1 - 0s - loss: 0.3875 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 396/400
1/1 - 0s - loss: 0.3869 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 397/400
1/1 - 0s - loss: 0.3864 - accuracy: 0.9000 - 6ms/epoch - 6ms/step
Epoch 398/400
1/1 - 0s - loss: 0.3859 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 399/400
1/1 - 0s - loss: 0.3854 - accuracy: 0.9000 - 7ms/epoch - 7ms/step
Epoch 400/400
1/1 - 0s - loss: 0.3849 - accuracy: 0.9000 - 12ms/epoch - 12ms/step
```



Loss Functions

```
import numpy as np
def mean_squared_error(act, pred):
diff = pred - act
differences_squared = diff ** 2
mean_diff = differences_squared.mean()
return mean_diff
act = np.array([1.1,2,1.7])
pred = np.array([1,1.7,1.5])
print(mean_squared_error(act,pred))
0.04666666666666667
from sklearn.metrics import mean_squared_error
act = np.array([1.1,2,1.7])
pred = np.array([1,1.7,1.5])
mean_squared_error(act, pred)
0.04666666666666667
import numpy as np
def root_mean_squared_error(act, pred):
diff = pred - act
differences_squared = diff ** 2
mean_diff = differences_squared.mean()
rmse_val = np.sqrt(mean_diff)
return rmse_val
act = np.array([1.1,2,1.7])
pred = np.array([1,1.7,1.5])
print(root_mean_squared_error(act,pred))
0.21602468994692867
```

Fit the Model

```
from sklearn.datasets import load_boston
from keras.models import Sequential
from keras.layers import Dense, Conv1D, Flatten
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
boston = load_boston()
x, y = boston.data, boston.target
print(x.shape)
(506, 13)
(506, 13)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/
deprecation.py:87: FutureWarning: Function load_boston is deprecated;
`load_boston` is deprecated in 1.0 and will be removed in 1.2.
The Boston housing prices dataset has an ethical problem. You can
refer to
the documentation of this function for further details.
The scikit-learn maintainers therefore strongly discourage the use
of this
dataset unless the purpose of the code is to study and educate
```

```
about
ethical issues in data science and machine learning.
In this special case, you can fetch the dataset from the original
source::
import pandas as pd
import numpy as np
data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22,
header=None)
data = np.hstack([raw_df.values[::2, :],
raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]
Alternative datasets include the California housing dataset (i.e.
:func:`~sklearn.datasets.fetch_california_housing`) and the Ames
housing
dataset. You can load the datasets as follows::
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
for the California housing dataset and::
from sklearn.datasets import fetch_openml
housing = fetch_openml(name="house_prices", as_frame=True)
for the Ames housing dataset.
warnings.warn(msg, category=FutureWarning)
(506, 13)
x = x.reshape(x.shape[0], x.shape[1], 1)
print(x.shape)
(506, 13, 1)
(506, 13, 1)
(506, 13, 1)
model = Sequential()
model.add(Conv1D(32, 2, activation="relu", input_shape=(13, 1)))
model.add(Flatten())
model.add(Dense(64, activation="relu"))
model.add(Dense(1))
model.compile(loss="mse", optimizer="adam")
model.summary()
Model: "sequential_3"
_________________________________________________________________
Layer (type) Output Shape Param #
=================================================================
conv1d (Conv1D) (None, 12, 32) 96
flatten_1 (Flatten) (None, 384) 0
dense_6 (Dense) (None, 64) 24640
dense_7 (Dense) (None, 1) 65
=================================================================
Total params: 24,801
Trainable params: 24,801
Non-trainable params: 0
```

```
ypred = model.predict(xtest)
print(model.evaluate(xtrain, ytrain))
21.21026409947595
----------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-21-f7e2d420a5c1> in <module>
----> 1 print(model.evaluate(xtrain, ytrain))
2 21.21026409947595
NameError: name 'xtrain' is not defined
print(model.evaluate(xtrain, ytrain))
21.21026409947595
print("MSE: %.4f" % mean_squared_error(ytest, ypred))
MSE: 19.8953
x_ax = range(len(ypred))
plt.scatter(x_ax, ytest, s=5, color="blue", label="original")
plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
----------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-22-c2bb07290788> in <module>
----> 1 print(model.evaluate(xtrain, ytrain))
2 21.21026409947595
3
4 print("MSE: %.4f" % mean_squared_error(ytest, ypred))
5 MSE: 19.8953
NameError: name 'xtrain' is not defined
```

Save the Model

```
#trying to save the model
model_json = dvc_classifier.to_json()
with open("/content/1354396826_2868631432_m.jpg") as json_file:
json_file.write(model_json)
----------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-23-53cf0ca21347> in <module>
1 #trying to save the model
----> 2 model_json = dvc_classifier.to_json()
3 with open("/content/1354396826_2868631432_m.jpg") as
json_file:
4 json_file.write(model_json)
NameError: name 'dvc_classifier' is not defined
```

```
# MLP for Pima Indians Dataset Serialize to JSON and HDF5
from tensorflow.keras.models import Sequential, model_from_json
from tensorflow.keras.layers import Dense
import numpy
import os
# fix random seed for reproducibility
numpy.random.seed(7)
# load pima indians dataset
dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
# Fit the model
model.fit(X, Y, epochs=150, batch_size=10, verbose=0)
# evaluate the model
scores = model.evaluate(X, Y, verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
# serialize model to JSON
model_json = model.to_json()
with open("model.json", "w") as json_file:
json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("model.h5")
print("Saved model to disk")
# later...
# load json and create model
json_file = open('model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("model.h5")
print("Loaded model from disk")
# evaluate loaded model on test data
loaded_model.compile(loss='binary_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])
score = loaded_model.evaluate(X, Y, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
# MLP for Pima Indians Dataset Serialize to JSON and HDF5
from tensorflow.keras.models import Sequential, model_from_json
from tensorflow.keras.layers import Dense
import numpy
import os
# fix random seed for reproducibility
numpy.random.seed(7)
# load pima indians dataset
dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
# Fit the model
model.fit(X, Y, epochs=150, batch_size=10, verbose=0)
# evaluate the model
scores = model.evaluate(X, Y, verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
# serialize model to JSON
model_json = model.to_json()
with open("model.json", "w") as json_file:
json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("model.h5")
print("Saved model to disk")
# later...
# load json and create model
json_file = open('model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("model.h5")
print("Loaded model from disk")
# evaluate loaded model on test data
loaded_model.compile(loss='binary_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])
score = loaded_model.evaluate(X, Y, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
---------------------------------------------------------------------
-----
OSError Traceback (most recent call
last)
```

```
<ipython-input-24-3e4eaa27bacb> in <module>
7 numpy.random.seed(7)
8 # load pima indians dataset
----> 9 dataset = numpy.loadtxt("pima-indians-diabetes.csv",
delimiter=",")
10 # split into input (X) and output (Y) variables
11 X = dataset[:,0:8]
/usr/local/lib/python3.7/dist-packages/numpy/lib/npyio.py in
loadtxt(fname, dtype, comments, delimiter, converters, skiprows,
usecols, unpack, ndmin, encoding, max_rows, like)
1065 fname = os_fspath(fname)
1066 if _is_string_like(fname):
-> 1067 fh = np.lib._datasource.open(fname, 'rt',
encoding=encoding)
1068 fencoding = getattr(fh, 'encoding', 'latin1')
1069 fh = iter(fh)
/usr/local/lib/python3.7/dist-packages/numpy/lib/_datasource.py in
open(path, mode, destpath, encoding, newline)
191
192 ds = DataSource(destpath)
--> 193 return ds.open(path, mode, encoding=encoding,
newline=newline)
194
195
/usr/local/lib/python3.7/dist-packages/numpy/lib/_datasource.py in
open(self, path, mode, encoding, newline)
531 encoding=encoding,
newline=newline)
532 else:
--> 533 raise IOError("%s not found." % path)
534
535
OSError: pima-indians-diabetes.csv not found.
```

Test the Model

```
import requests
url='/content/1354396826_2868631432_m.jpg'
response=requests.get(url,stream=True)
```

```
---------------------------------------------------------------------------
-----
MissingSchema Traceback (most recent call
last)
<ipython-input-31-65b509a1a469> in <module>
----> 1 response=requests.get(url,stream=True)
/usr/local/lib/python3.7/dist-packages/requests/api.py in get(url,
params, **kwargs)
74
75 kwargs.setdefault('allow_redirects', True)
---> 76 return request('get', url, params=params, **kwargs)
```

```
77
78
/usr/local/lib/python3.7/dist-packages/requests/api.py in
request(method, url, **kwargs)
59 # cases, and look like a memory leak in others.
60 with sessions.Session() as session:
---> 61 return session.request(method=method, url=url,
**kwargs)
62
63
/usr/local/lib/python3.7/dist-packages/requests/sessions.py in
request(self, method, url, params, data, headers, cookies, files,
auth, timeout, allow_redirects, proxies, hooks, stream, verify, cert,
json)
514 hooks=hooks,
515 )
--> 516 prep = self.prepare_request(req)
517
518 proxies = proxies or {}
/usr/local/lib/python3.7/dist-packages/requests/sessions.py in
prepare_request(self, request)
457 auth=merge_setting(auth, self.auth),
458 cookies=merged_cookies,
--> 459 hooks=merge_hooks(request.hooks, self.hooks),
460 )
461 return p
/usr/local/lib/python3.7/dist-packages/requests/models.py in
prepare(self, method, url, headers, files, data, params, auth,
cookies, hooks, json)
312
313 self.prepare_method(method)
--> 314 self.prepare_url(url, params)
315 self.prepare_headers(headers)
316 self.prepare_cookies(cookies)
/usr/local/lib/python3.7/dist-packages/requests/models.py in
prepare_url(self, url, params)
386 error = error.format(to_native_string(url,
'utf8'))
387
--> 388 raise MissingSchema(error)
389
390 if not host:
MissingSchema: Invalid URL '/content/1354396826_2868631432_m.jpg': No
schema supplied. Perhaps you meant
http:///content/1354396826_2868631432_m.jpg?
```

```
from PIL import Image
img=Image.open(response.raw)
---------------------------------------------------------------------------
```

```
-----
NameError Traceback (most recent call
last)
<ipython-input-30-a5a92772ad5e> in <module>
----> 1 img=Image.open(response.raw)
NameError: name 'response' is not defined
plt.imshow(img)
plt.show()
```

```
img=PIL.ImageOps.invert(img)
plt.show()
---------------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-36-be3f6ff0e4db> in <module>
----> 1 img=PIL.ImageOps.invert(img)
2 plt.show()
NameError: name 'PIL' is not defined
import PIL.ImageOps
plt.imshow(im_convert(img))
---------------------------------------------------------------------------
-----
NameError Traceback (most recent call
last)
<ipython-input-37-afd34ca72d06> in <module>
1 import PIL.ImageOps
----> 2 plt.imshow(im_convert(img))
NameError: name 'im_convert' is not defined
```