Bounding box detection - Racoon data

Data files

- images_racoon.rar: contain images of racoons
- train_labels.cv: contains coordinates for bounding box for every image

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318

Enter your authorization code:
··········
Mounted at /content/drive
```

Import the necessary libraries

```
# IMPORT LIBRARIES AND PACKAGES
import tensorflow as tf
import csv
import numpy as np
from PIL import Image

from keras import Model
from keras.applications.mobilenet import MobileNet, preprocess_input
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, Callback
from keras.layers import Conv2D, Reshape
from keras.utils import Sequence
from keras.backend import epsilon
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the `%tens` `1.x` magic: more info.

```
Using TensorFlow backend
```

Change directory

```
import os

DATASET_FOLDER = "/content/drive/My Drive/greatlakes/Residency9/InternalLab/"
os.chdir(DATASET_FOLDER)
```

Load the training data from train.csv file

```
173
```

Print the shape of the train dataset

```
batch_images.shape
```

```
(173, 128, 128, 3)
```

Declare a variable IMAGE_SIZE = 128 as we will be using MobileNet which will be

```
IMAGE_SIZE = 128 # MobileNet takes images of size 128*128*3
```

With the help of csv.reader write a for loop which can load the train.csv file and s x0,y0,x1,y1 in induvidual variables.

1. Create a list variable known as 'path' which has all the path for all the training images
2. Create an array 'coords' which has the resized coordinates of the bounding box for the training images

Note: All the training images should be downsampled to 128 * 128 as it is the input shape of MobileN detection). Hence the corresponding coordinates of the bounding boxes should be changed to match

```
TRAIN_CSV = DATASET_FOLDER+"train_labels.csv"
images_path = DATASET_FOLDER + "images/"

import csv
with open(TRAIN_CSV, 'r') as csvfile:

    paths = []
    coords = np.zeros((sum(1 for line in csvfile)-1, 4))
    reader = csv.reader(csvfile, delimiter=',')
    csvfile.seek(0)
    next(csvfile)
    for col, row in enumerate(reader):

        path = images_path + row[0]
        image_width,image_height,xmin, ymin, xmax, ymax = int(row[1]),int(row[2]),int(row[4])
        #print(image_width,image_height,xmin, ymin, xmax, ymax)
        coords[col, 0] = xmin * IMAGE_SIZE / image_width # Normalize bounding box by image si
        coords[col, 1] = ymin * IMAGE_SIZE / image_height # Normalize bounding box by image s
```

```
        coords[col, 2] = (xmax - xmin) * IMAGE_SIZE / image_width # Normalize bounding box by
        coords[col, 3] = (ymax - ymin) * IMAGE_SIZE / image_height
        paths.append(path)

print(len(paths))
print(len(coords))
```

```
173
173
```

Write a for loop which can load all the training images into a variable 'batch_imag 'paths' variable

Note: Convert the image to RGB scale as the MobileNet accepts 3 channels as inputs

```
batch_images = np.zeros((len(paths), IMAGE_SIZE, IMAGE_SIZE, 3), dtype=np.float32)
for i, f in enumerate(paths):
    img = Image.open(f) # Read image
    img = img.resize((IMAGE_SIZE, IMAGE_SIZE)) # Resize image
    img = img.convert('RGB')
    batch_images[i] = preprocess_input(np.array(img, dtype=np.float32))
```

Import MobileNet and load MobileNet into a variable named 'model' which takes Freeze all the layers. Add convolution and reshape layers at the end to ensure the

```
model = MobileNet(input_shape=(IMAGE_SIZE, IMAGE_SIZE, 3), include_top=False) # Load pre-trai
# Do not include classification (top) layer

# to freeze layers, except the new top layer, of course, which will be added below
for layer in model.layers:
    layer.trainable = False

# Add new top layer which is a conv layer of the same size as the previous layer so that only
x = model.layers[-1].output
x = Conv2D(4, kernel_size=4, name="coords")(x)
# In the line above kernel size should be 3 for img size 96, 4 for img size 128, 5 for img si
x = Reshape((4,))(x) # These are the 4 predicted coordinates of one BBox

model = Model(inputs=model.input, outputs=x)
```

Define a custom loss function IoU which calculates Intersection Over Union

```
def loss(gt,pred):
    intersections = 0
    unions = 0
```

```
    unions = 0
    diff_width = np.minimum(gt[:,0] + gt[:,2], pred[:,0] + pred[:,2]) - np.maximum(gt[:,0], p
    diff_height = np.minimum(gt[:,1] + gt[:,3], pred[:,1] + pred[:,3]) - np.maximum(gt[:,1],
    intersection = diff_width * diff_height

    # Compute union
    area_gt = gt[:,2] * gt[:,3]
    area_pred = pred[:,2] * pred[:,3]
    union = area_gt + area_pred - intersection

#     Compute intersection and union over multiple boxes
    for j, _ in enumerate(union):
        if union[j] > 0 and intersection[j] > 0 and union[j] >= intersection[j]:
            intersections += intersection[j]
            unions += union[j]

    # Compute IOU. Use epsilon to prevent division by zero
    iou = np.round(intersections / (unions + epsilon()), 4)
    iou = iou.astype(np.float32)
    return iou

def IoU(y_true, y_pred):
    iou = tf.py_func(loss, [y_true, y_pred], tf.float32)
    return iou


model.summary()
```

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_2 (InputLayer)         (None, 128, 128, 3)       0         
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 129, 129, 3)       0         
_________________________________________________________________
conv1 (Conv2D)               (None, 64, 64, 32)        864       
_________________________________________________________________
conv1_bn (BatchNormalization (None, 64, 64, 32)        128       
_________________________________________________________________
conv1_relu (ReLU)            (None, 64, 64, 32)        0         
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 64, 64, 32)        288       
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 64, 64, 32)        128       
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 64, 64, 32)        0         
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 64, 64, 64)        2048      
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 64, 64, 64)        256       
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 64, 64, 64)        0         
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 65, 65, 64)        0         
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 32, 32, 64)        576       
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 32, 32, 64)        256       
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 32, 32, 64)        0         
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 32, 32, 128)       8192      
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 32, 32, 128)       512       
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 32, 32, 128)       0         
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 32, 32, 128)       1152      
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 32, 32, 128)       512       
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 32, 32, 128)       0         
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 32, 32, 128)       16384     
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 32, 32, 128)       512       
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 32, 32, 128)       0         
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 33, 33, 128)       0         
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 16, 16, 128)       1152      
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 16, 16, 128)       512       
```

```
conv_dw_4_relu (ReLU)        (None, 16, 16, 128)       0
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 16, 16, 256)       32768
_________________________________________________________________
conv_pw_4_bn (BatchNormaliza (None, 16, 16, 256)       1024
_________________________________________________________________
conv_pw_4_relu (ReLU)        (None, 16, 16, 256)       0
_________________________________________________________________
conv_dw_5 (DepthwiseConv2D)  (None, 16, 16, 256)       2304
_________________________________________________________________
conv_dw_5_bn (BatchNormaliza (None, 16, 16, 256)       1024
_________________________________________________________________
conv_dw_5_relu (ReLU)        (None, 16, 16, 256)       0
_________________________________________________________________
conv_pw_5 (Conv2D)           (None, 16, 16, 256)       65536
_________________________________________________________________
conv_pw_5_bn (BatchNormaliza (None, 16, 16, 256)       1024
_________________________________________________________________
conv_pw_5_relu (ReLU)        (None, 16, 16, 256)       0
_________________________________________________________________
conv_pad_6 (ZeroPadding2D)   (None, 17, 17, 256)       0
_________________________________________________________________
conv_dw_6 (DepthwiseConv2D)  (None, 8, 8, 256)         2304
_________________________________________________________________
conv_dw_6_bn (BatchNormaliza (None, 8, 8, 256)         1024
_________________________________________________________________
conv_dw_6_relu (ReLU)        (None, 8, 8, 256)         0
_________________________________________________________________
conv_pw_6 (Conv2D)           (None, 8, 8, 512)         131072
_________________________________________________________________
conv_pw_6_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_6_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_dw_7 (DepthwiseConv2D)  (None, 8, 8, 512)         4608
_________________________________________________________________
conv_dw_7_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_dw_7_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_pw_7 (Conv2D)           (None, 8, 8, 512)         262144
_________________________________________________________________
conv_pw_7_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_7_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_dw_8 (DepthwiseConv2D)  (None, 8, 8, 512)         4608
_________________________________________________________________
conv_dw_8_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_dw_8_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_pw_8 (Conv2D)           (None, 8, 8, 512)         262144
_________________________________________________________________
conv_pw_8_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_8_relu (ReLU)        (None, 8, 8, 512)         0
```

```
_________________________________________________________________
conv_dw_9 (DepthwiseConv2D)  (None, 8, 8, 512)         4608
_________________________________________________________________
conv_dw_9_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_dw_9_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_pw_9 (Conv2D)           (None, 8, 8, 512)         262144
_________________________________________________________________
conv_pw_9_bn (BatchNormaliza (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_9_relu (ReLU)        (None, 8, 8, 512)         0
_________________________________________________________________
conv_dw_10 (DepthwiseConv2D) (None, 8, 8, 512)         4608
_________________________________________________________________
conv_dw_10_bn (BatchNormaliz (None, 8, 8, 512)         2048
_________________________________________________________________
conv_dw_10_relu (ReLU)       (None, 8, 8, 512)         0
_________________________________________________________________
conv_pw_10 (Conv2D)          (None, 8, 8, 512)         262144
_________________________________________________________________
conv_pw_10_bn (BatchNormaliz (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_10_relu (ReLU)       (None, 8, 8, 512)         0
_________________________________________________________________
conv_dw_11 (DepthwiseConv2D) (None, 8, 8, 512)         4608
_________________________________________________________________
conv_dw_11_bn (BatchNormaliz (None, 8, 8, 512)         2048
_________________________________________________________________
conv_dw_11_relu (ReLU)       (None, 8, 8, 512)         0
_________________________________________________________________
conv_pw_11 (Conv2D)          (None, 8, 8, 512)         262144
_________________________________________________________________
conv_pw_11_bn (BatchNormaliz (None, 8, 8, 512)         2048
_________________________________________________________________
conv_pw_11_relu (ReLU)       (None, 8, 8, 512)         0
_________________________________________________________________
conv_pad_12 (ZeroPadding2D)  (None, 9, 9, 512)         0
_________________________________________________________________
conv_dw_12 (DepthwiseConv2D) (None, 4, 4, 512)         4608
_________________________________________________________________
conv_dw_12_bn (BatchNormaliz (None, 4, 4, 512)         2048
_________________________________________________________________
conv_dw_12_relu (ReLU)       (None, 4, 4, 512)         0
_________________________________________________________________
conv_pw_12 (Conv2D)          (None, 4, 4, 1024)        524288
_________________________________________________________________
conv_pw_12_bn (BatchNormaliz (None, 4, 4, 1024)        4096
_________________________________________________________________
conv_pw_12_relu (ReLU)       (None, 4, 4, 1024)        0
_________________________________________________________________
conv_dw_13 (DepthwiseConv2D) (None, 4, 4, 1024)        9216
_________________________________________________________________
conv_dw_13_bn (BatchNormaliz (None, 4, 4, 1024)        4096
_________________________________________________________________
conv_dw_13_relu (ReLU)       (None, 4, 4, 1024)        0
_________________________________________________________________
conv_pw_13 (Conv2D)          (None, 4, 4, 1024)        1048576
```

```
conv_pw_13 (Conv2D)          (None, 4, 4, 1024)        1048576
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, 4, 4, 1024)        4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, 4, 4, 1024)        0
_________________________________________________________________
coords (Conv2D)              (None, 1, 1, 4)           65540
_________________________________________________________________
reshape_2 (Reshape)          (None, 4)                 0
=================================================================
Total params: 3,294,404
Trainable params: 65,540
Non-trainable params: 3,228,864
```

Write model.compile function & model.fit function with:

1. Optimizer = Adam, Loss = 'mse' and metrics = IoU
2. Epochs = 30, batch_size = 32, verbose = 1

```
len(coords)
```

```
174
```

```
EPOCHS = 30 # Number of epochs. I got decent performance with just 5.
BATCH_SIZE = 32 # Depends on your GPU or CPU RAM.
ground_truth = coords
model.compile(optimizer='Adam', loss='mse', metrics=[IoU]) # Regression loss is MSE

#checkpoint = ModelCheckpoint("model-{val_iou:.2f}.h5", verbose=1, save_best_only=True,
#                              save_weights_only=True, mode="max", period=1) # Checkpoint bes
#stop = EarlyStopping(monitor="val_iou", patience=PATIENCE, mode="max") # Stop early, if the
#reduce_lr = ReduceLROnPlateau(monitor="val_iou", factor=0.2, patience=10, min_lr=1e-7, verbo
# Reduce learning rate if Validation IOU does not improve

model.fit(batch_images,ground_truth,
            epochs=EPOCHS,batch_size = BATCH_SIZE,
            verbose=1)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

Epoch 1/30
173/173 [==============================] - 6s 33ms/step - loss: 2887.0585 - IoU: 0.0774
Epoch 2/30
173/173 [==============================] - 5s 26ms/step - loss: 699.2910 - IoU: 0.4297
Epoch 3/30
173/173 [==============================] - 5s 27ms/step - loss: 652.4600 - IoU: 0.5381
Epoch 4/30
173/173 [==============================] - 5s 27ms/step - loss: 586.4492 - IoU: 0.5254
Epoch 5/30
173/173 [==============================] - 5s 28ms/step - loss: 343.5283 - IoU: 0.5862
Epoch 6/30
173/173 [==============================] - 5s 27ms/step - loss: 274.0325 - IoU: 0.5791
Epoch 7/30
173/173 [==============================] - 5s 27ms/step - loss: 241.8863 - IoU: 0.6143
Epoch 8/30
173/173 [==============================] - 5s 27ms/step - loss: 199.9005 - IoU: 0.6554
Epoch 9/30
173/173 [==============================] - 5s 27ms/step - loss: 165.8996 - IoU: 0.7029
Epoch 10/30
173/173 [==============================] - 5s 27ms/step - loss: 156.4762 - IoU: 0.7159
Epoch 11/30
173/173 [==============================] - 5s 27ms/step - loss: 146.9060 - IoU: 0.7208
Epoch 12/30
173/173 [==============================] - 5s 27ms/step - loss: 119.1781 - IoU: 0.7342
Epoch 13/30
173/173 [==============================] - 5s 27ms/step - loss: 113.4647 - IoU: 0.7321
Epoch 14/30
173/173 [==============================] - 5s 27ms/step - loss: 100.2294 - IoU: 0.7453
Epoch 15/30
173/173 [==============================] - 5s 27ms/step - loss: 93.0235 - IoU: 0.7731
Epoch 16/30
173/173 [==============================] - 5s 27ms/step - loss: 89.4955 - IoU: 0.7780
Epoch 17/30
173/173 [==============================] - 5s 27ms/step - loss: 82.8049 - IoU: 0.7824
Epoch 18/30
173/173 [==============================] - 5s 27ms/step - loss: 81.1191 - IoU: 0.7942
Epoch 19/30
173/173 [==============================] - 5s 27ms/step - loss: 78.1954 - IoU: 0.7929
Epoch 20/30
173/173 [==============================] - 5s 27ms/step - loss: 70.5219 - IoU: 0.8061
Epoch 21/30
173/173 [==============================] - 5s 27ms/step - loss: 67.6122 - IoU: 0.8128
Epoch 22/30
173/173 [==============================] - 5s 27ms/step - loss: 68.0727 - IoU: 0.8158
Epoch 23/30
173/173 [==============================] - 5s 27ms/step - loss: 67.7651 - IoU: 0.8114
Epoch 24/30
173/173 [==============================] - 5s 27ms/step - loss: 66.3373 - IoU: 0.8090
Epoch 25/30
173/173 [==============================] - 5s 27ms/step - loss: 66.5067 - IoU: 0.8176
Epoch 26/30
173/173 [==============================] - 5s 27ms/step - loss: 61.7543 - IoU: 0.8223
Epoch 27/30
```

```
173/173 [==============================] - 5s 26ms/step - loss: 61.7269 - IoU: 0.8313
Epoch 28/30
173/173 [==============================] - 5s 27ms/step - loss: 59.2167 - IoU: 0.8316
Epoch 29/30
173/173 [==============================] - 5s 27ms/step - loss: 60.2528 - IoU: 0.8427
Epoch 30/30
173/173 [==============================] - 5s 27ms/step - loss: 58.2374 - IoU: 0.8436
<keras.callbacks.History at 0x7f3aa7334198>
```

Pick a test image from the given data

```
import cv2
filename = './images/raccoon-16.jpg'
unscaled = cv2.imread(filename) # Original image for display
```

Resize the image to 128 * 128 and preprocess the image for the MobileNet mode

```
image_height, image_width, _ = unscaled.shape
image = cv2.resize(unscaled, (IMAGE_SIZE, IMAGE_SIZE)) # Rescaled image to run the network
feat_scaled = preprocess_input(np.array(image, dtype=np.float32))
```

Predict the coordinates of the bounding box for the given test image

```
region = model.predict(x=np.array([feat_scaled]))[0]
print(region)
```

```
[ 30.74896   29.288248  85.985504 107.34286 ]
```

Plot the test image using .imshow and draw a boundary box around the image w the model

```
x0 = int(region[0] * image_width / IMAGE_SIZE) # Scale the BBox
y0 = int(region[1] * image_height / IMAGE_SIZE)

x1 = int((region[2]) * image_width / IMAGE_SIZE)
y1 = int((region[3]) * image_height / IMAGE_SIZE)
```

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from PIL import Image
import numpy as np
```

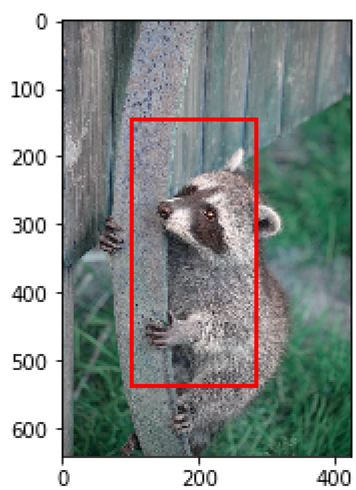
```
# Create figure and axes
fig,ax = plt.subplots(1)

# Display the image
ax.imshow(unscaled)

# Create a Rectangle patch
rect = patches.Rectangle((x0, y0), (x1 - x0) , (y1 - y0) , linewidth=2, edgecolor='r', facecc

# Add the patch to the Axes
ax.add_patch(rect)

plt.show()
```



Time Series Prediction using LSTM

Download Data

Link: https://datamarket.com/data/set/2324/daily-minimum-temperatures-in-melbourne-australia-19

Description

Daily minimum temperatures in Melbourne, Australia, 1981-1990

Units: Degrees Celcius

Steps before loading

- Rename the column name with temprature values to "Temprature"
- In the last, there is one extra row in the data, remove it by opening the file and save it again.
- There are some values in Temprature column which have a "?" before them, they will give error,
- If you don't want to do these steps, just load the data file given by Great Learning.

Mount google drive

Change your present working directory

Load your data file

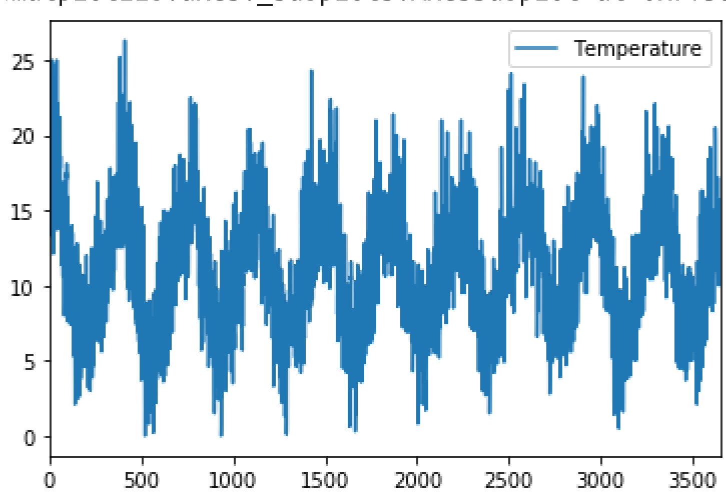
```
import pandas as pd
df = pd.read_csv('/content/drive/My Drive/greatlakes/Residency9/InternalLab/daily-minimum-ten
df.sort_index(inplace=True)
df.head()
```

	Date	Temperature
0	1981-01-01	20.7
1	1981-01-02	17.9
2	1981-01-03	18.8
3	1981-01-04	14.6
4	1981-01-05	15.8

Plot data

```
df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3a9c61b128>
```



Descibe your dataframe

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Temperature	3650.0	11.177753	4.071837	0.0	8.3	11.0	14.0	26.3

Check for null values

```
df.isnull().sum()
```

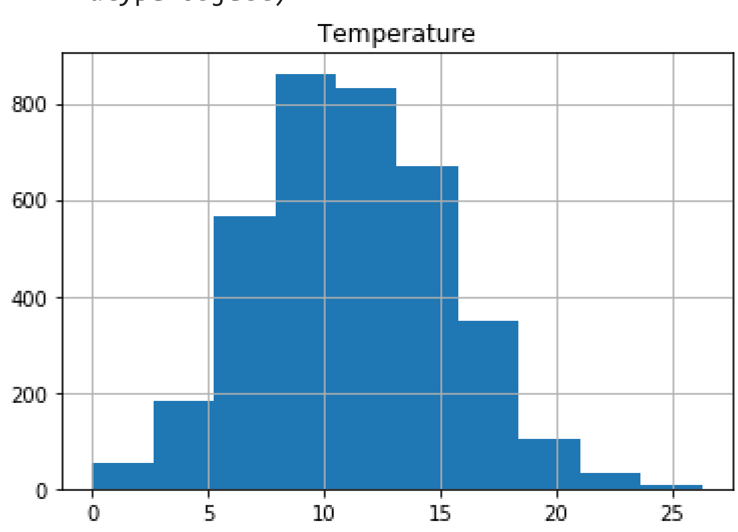
```
Date           0
Temperature    0
dtype: int64
```

Drop null values

Get the representation of the distribution of data in the form of histogram

```
df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f3aa7cc5198>]],
      dtype=object)
```



Check the maximum and minimum values

```
#Check Data Range
```

```
print('Min', np.min(df))
print('Max', np.max(df))
```

```
Min Date           1981-01-01
Temperature             0
dtype: object
Max Date           1990-12-31
Temperature          26.3
dtype: object
```

Normalize the data

```
df.drop("Date", axis=1, inplace=True)
from sklearn.preprocessing import MinMaxScaler
#Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(df)
```

Check the maximum and minimum values of scaled data

```
#Check Data Range
print('Min', np.min(scaled))
print('Max', np.max(scaled))
```

```
Min 0.0
Max 1.0
```

Look into some of the scaled values

```
scaled[:10]
```

```
array([[0.78707224],
       [0.68060837],
       [0.7148289 ],
       [0.55513308],
       [0.60076046],
       [0.60076046],
       [0.60076046],
       [0.66159696],
       [0.82889734],
       [0.76045627]])
```

Split data into Training and Testing

```
#70% examples will used for training
train_size = int(len(scaled) * 0.70)
#30% will be used for Test
```

```
test_size = len(scaled - train_size)

#Split the data
train, test = scaled[0:train_size, :], scaled[train_size: len(scaled), :]
```

Print train and test size

```
print('train: {}\ntest: {}'.format(len(train), len(test)))
```

```
train: 2555
test: 1095
```

Create the sequential data

Map the temprature at a particular time t to the temprature at time t+n, where n is any number you de

For example: to map tempratures of consecutive days, use t+1, i.e. loop_back = 1

Define your function to create dataset

```
#window - how long the sequence will be
def create_dataset(dataset, window=1):

    dataX, dataY = [], []

    for i in range(len(dataset)-window):

        a = dataset[i:(i+window), 0]
        dataX.append(a)
        dataY.append(dataset[i + window, 0])

    return np.array(dataX), np.array(dataY)
```

Use function to get training and test set

```
#Create Input and Output
window_size = 1
X_train, y_train = create_dataset(train, window_size)
X_test, y_test = create_dataset(test, window_size)
```

Transform the prepared train and test input data into the expected structure using numpy.

```
#Make it 3 Dimensional Data - needed for LSTM
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
```

```
print(X_train.shape)
print(X_test.shape)
```

```
(2554, 1, 1)
(1094, 1, 1)
```

Define Model

Define sequntial model, add LSTM layer and compile the model

```
import tensorflow as tf
tf.keras.backend.clear_session()
model = tf.keras.Sequential()
model.add(tf.keras.layers.LSTM(32,input_shape=(window_size,1)))
model.add(tf.keras.layers.Dense(1))
model.compile(optimizer='adam',loss='mse')
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/op
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```

Summarize your model

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 32)                4352
_________________________________________________________________
dense (Dense)                (None, 1)                 33
=================================================================
Total params: 4,385
Trainable params: 4,385
Non-trainable params: 0
_________________________________________________________________
```

Train the model

```
model.fit(X_train, y_train,epochs=200,  validation_data=(X_test, y_test), batch_size=32)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/op
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Train on 2554 samples, validate on 1094 samples
Epoch 1/200
2554/2554 [==============================] - 1s 292us/sample - loss: 0.1092 - val_loss:
Epoch 2/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0192 - val_loss: 0
Epoch 3/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0150 - val_loss: 0
Epoch 4/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0141 - val_loss: 0
Epoch 5/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0132 - val_loss: 0
Epoch 6/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0124 - val_loss: 0
Epoch 7/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0117 - val_loss: 0
Epoch 8/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0111 - val_loss: 0
Epoch 9/200
2554/2554 [==============================] - 0s 100us/sample - loss: 0.0107 - val_loss:
Epoch 10/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0104 - val_loss: 0
Epoch 11/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0102 - val_loss: 0
Epoch 12/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0101 - val_loss: 0
Epoch 13/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0101 - val_loss: 0
Epoch 14/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0101 - val_loss: 0
Epoch 15/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0101 - val_loss: 0
Epoch 16/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 17/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0100 - val_loss: 0
Epoch 18/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 19/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 20/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 21/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 22/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 23/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 24/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 25/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0101 - val_loss: 0
Epoch 26/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 27/200
```

```
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0101 - val_loss: 0
Epoch 28/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 29/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 30/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0100 - val_loss: 0
Epoch 31/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 32/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 33/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 34/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0101 - val_loss: 0
Epoch 35/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 36/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0101 - val_loss: 0
Epoch 37/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0101 - val_loss: 0
Epoch 38/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 39/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0100 - val_loss: 0
Epoch 40/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 41/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0101 - val_loss: 0
Epoch 42/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 43/200
2554/2554 [==============================] - 0s 98us/sample - loss: 0.0100 - val_loss: 0
Epoch 44/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 45/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 46/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 47/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 48/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0101 - val_loss: 0
Epoch 49/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0101 - val_loss: 0
Epoch 50/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 51/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0100 - val_loss: 0
Epoch 52/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0100 - val_loss: 0
Epoch 53/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 54/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 55/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 56/200
```

```
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 57/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 58/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 59/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0101 - val_loss: 0
Epoch 60/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 61/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 62/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 63/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 64/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 65/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 66/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 67/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 68/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 69/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 70/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 71/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 72/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 73/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 74/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 75/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 76/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 77/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 78/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 79/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 80/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 81/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0101 - val_loss: 0
Epoch 82/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 83/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 84/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 85/200
```

```
Epoch 85/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 86/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 87/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 88/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 89/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 90/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 91/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 92/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 93/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 94/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 95/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 96/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 97/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 98/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 99/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 100/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 101/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 102/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 103/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 104/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 105/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 106/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 107/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 108/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 109/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 110/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0100 - val_loss: 0
Epoch 111/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 112/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0101 - val_loss: 0
Epoch 113/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
```

```
Epoch 114/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 115/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 116/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 117/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 118/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 119/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 120/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 121/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 122/200
2554/2554 [==============================] - 0s 98us/sample - loss: 0.0100 - val_loss: 0
Epoch 123/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0101 - val_loss: 0
Epoch 124/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 125/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0100 - val_loss: 0
Epoch 126/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 127/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 128/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 129/200
2554/2554 [==============================] - 0s 100us/sample - loss: 0.0100 - val_loss:
Epoch 130/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 131/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 132/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 133/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 134/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 135/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 136/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0099 - val_loss: 0
Epoch 137/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 138/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 139/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 140/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 141/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 142/200
2554/2554 [==============================] - 0s 99us/sample - loss: 0.0100 - val_loss: 0
```

```
Epoch 143/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0099 - val_loss: 0
Epoch 144/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 145/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 146/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 147/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 148/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 149/200
2554/2554 [==============================] - 0s 102us/sample - loss: 0.0100 - val_loss:
Epoch 150/200
2554/2554 [==============================] - 0s 105us/sample - loss: 0.0100 - val_loss:
Epoch 151/200
2554/2554 [==============================] - 0s 103us/sample - loss: 0.0100 - val_loss:
Epoch 152/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 153/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 154/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 155/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 156/200
2554/2554 [==============================] - 0s 86us/sample - loss: 0.0100 - val_loss: 0
Epoch 157/200
2554/2554 [==============================] - 0s 85us/sample - loss: 0.0100 - val_loss: 0
Epoch 158/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 159/200
2554/2554 [==============================] - 0s 95us/sample - loss: 0.0100 - val_loss: 0
Epoch 160/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0100 - val_loss: 0
Epoch 161/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 162/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 163/200
2554/2554 [==============================] - 0s 88us/sample - loss: 0.0100 - val_loss: 0
Epoch 164/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 165/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 166/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 167/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 168/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 169/200
2554/2554 [==============================] - 0s 97us/sample - loss: 0.0100 - val_loss: 0
Epoch 170/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 171/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
```

```
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 172/200
2554/2554 [==============================] - 0s 94us/sample - loss: 0.0100 - val_loss: 0
Epoch 173/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 174/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 175/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 176/200
2554/2554 [==============================] - 0s 103us/sample - loss: 0.0100 - val_loss:
Epoch 177/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 178/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 179/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 180/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 181/200
2554/2554 [==============================] - 0s 103us/sample - loss: 0.0100 - val_loss:
Epoch 182/200
2554/2554 [==============================] - 0s 102us/sample - loss: 0.0100 - val_loss:
Epoch 183/200
2554/2554 [==============================] - 0s 103us/sample - loss: 0.0100 - val_loss:
Epoch 184/200
2554/2554 [==============================] - 0s 96us/sample - loss: 0.0099 - val_loss: 0
Epoch 185/200
2554/2554 [==============================] - 0s 89us/sample - loss: 0.0100 - val_loss: 0
Epoch 186/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 187/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 188/200
2554/2554 [==============================] - 0s 87us/sample - loss: 0.0100 - val_loss: 0
Epoch 189/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 190/200
2554/2554 [==============================] - 0s 92us/sample - loss: 0.0100 - val_loss: 0
Epoch 191/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 192/200
2554/2554 [==============================] - 0s 90us/sample - loss: 0.0100 - val_loss: 0
Epoch 193/200
2554/2554 [==============================] - 0s 104us/sample - loss: 0.0100 - val_loss:
Epoch 194/200
2554/2554 [==============================] - 0s 97us/sample - loss: 0.0100 - val_loss: 0
Epoch 195/200
2554/2554 [==============================] - 0s 101us/sample - loss: 0.0100 - val_loss:
Epoch 196/200
2554/2554 [==============================] - 0s 100us/sample - loss: 0.0100 - val_loss:
Epoch 197/200
2554/2554 [==============================] - 0s 93us/sample - loss: 0.0099 - val_loss: 0
Epoch 198/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
Epoch 199/200
2554/2554 [==============================] - 0s 91us/sample - loss: 0.0100 - val_loss: 0
```

Make Predictions and Evaluate your model

```
#Un-normalize the predited data
trainPredict = model.predict(X_train)
testPredict = model.predict(X_test)
trainPredict = scaler.inverse_transform(trainPredict)
testPredict = scaler.inverse_transform(testPredict)
```

Plot the results

```
import matplotlib.pyplot as plt
trainPredictPlot = np.empty_like(scaled)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[window_size:len(trainPredict)+window_size, :] = trainPredict
# shift test predictions for plotting
testPredictPlot = np.empty_like(scaled)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(trainPredict)+(window_size*2):len(scaled), :] = testPredict
# plot baseline and predictions
plt.figure(figsize=(10,6))
plt.plot(scaler.inverse_transform(scaled))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```