


```
[15]: #convert an array values into a dataset matrix
def create_dataset(dataset,time_step=1):
    datax_data=[]
    for i in range(1+(dataset)-time_step+1):
        a=dataset[(1:(1+time_step)+0)]
        datax.append(a)
        datax.append(dataset[i+ time_step,0])
        return np.array(datax), np.array(datay)

In [16]: time_step=100
X_train, y_train = create_dataset(train_data,time_step)
X_test, y_test= create_dataset(test_data,time_step)

In [17]: X_train.shape,y_train.shape

Out[17]: ((716, 100), (716,))

In [18]: X_train=X_train.reshape(X_train.shape[0],X_train.shape[1],1)
X_test=X_test.reshape(X_test.shape[0],X_test.shape[1],1)

In [19]: from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM

In [25]: model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')

In [26]: model.summary()

Model: "sequential_2"
-----
Layer (type) Output Shape Param #
-----
lstm_3 (LSTM) (None, 100, 50) 10400
lstm_4 (LSTM) (None, 100, 50) 20200
lstm_5 (LSTM) (None, 50) 20200
dense_1 (Dense) (None, 1) 51
-----
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0

In [27]: model.summary()

Model: "sequential_2"
-----
Layer (type) Output Shape Param #
-----
lstm_3 (LSTM) (None, 100, 50) 10400
lstm_4 (LSTM) (None, 50) 20200
lstm_5 (LSTM) (None, 1) 51
-----
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0

In [28]: model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=100,batch_size=64,verbose=1)

Epoch 1/100
12/12 [=====] - 8s 291ms/step - loss: 0.0164 - val_loss: 0.0630
Epoch 2/100
12/12 [=====] - 3s 212ms/step - loss: 0.0042 - val_loss: 0.0050
Epoch 3/100
12/12 [=====] - 3s 210ms/step - loss: 0.0016 - val_loss: 0.0035
Epoch 4/100
12/12 [=====] - 2s 207ms/step - loss: 0.0016 - val_loss: 0.0042
Epoch 5/100
12/12 [=====] - 2s 212ms/step - loss: 0.0048 - val_loss: 0.0045
Epoch 6/100
12/12 [=====] - 1s 115ms/step - loss: 5.9645e-04 - val_loss: 0.0033
Epoch 7/100
12/12 [=====] - 1s 117ms/step - loss: 5.6032e-04 - val_loss: 0.0035
Epoch 8/100
12/12 [=====] - 2s 210ms/step - loss: 5.5704e-04 - val_loss: 0.0032
Epoch 9/100
12/12 [=====] - 2s 195ms/step - loss: 5.5231e-04 - val_loss: 0.0039
Epoch 10/100
12/12 [=====] - 2s 210ms/step - loss: 5.5028e-04 - val_loss: 0.0035
Epoch 11/100
12/12 [=====] - 2s 209ms/step - loss: 5.5704e-04 - val_loss: 0.0032
Epoch 12/100
12/12 [=====] - 2s 213ms/step - loss: 5.5279e-04 - val_loss: 0.0032
Epoch 13/100
12/12 [=====] - 2s 212ms/step - loss: 5.1700e-04 - val_loss: 0.0030
Epoch 14/100
12/12 [=====] - 2s 199ms/step - loss: 5.2503e-04 - val_loss: 0.0029
Epoch 15/100
12/12 [=====] - 2s 206ms/step - loss: 5.2115e-04 - val_loss: 0.0030
Epoch 16/100
12/12 [=====] - 2s 200ms/step - loss: 4.9879e-04 - val_loss: 0.0028
Epoch 17/100
12/12 [=====] - 2s 201ms/step - loss: 5.0819e-04 - val_loss: 0.0027
Epoch 18/100
12/12 [=====] - 2s 189ms/step - loss: 4.8793e-04 - val_loss: 0.0028
Epoch 19/100
12/12 [=====] - 2s 188ms/step - loss: 5.1164e-04 - val_loss: 0.0028
Epoch 20/100
12/12 [=====] - 2s 185ms/step - loss: 4.8808e-04 - val_loss: 0.0023
Epoch 21/100
12/12 [=====] - 2s 193ms/step - loss: 4.6186e-04 - val_loss: 0.0025
Epoch 22/100
12/12 [=====] - 2s 170ms/step - loss: 4.6580e-04 - val_loss: 0.0024
Epoch 23/100
12/12 [=====] - 2s 171ms/step - loss: 4.4547e-04 - val_loss: 0.0025
Epoch 24/100
12/12 [=====] - 2s 179ms/step - loss: 4.3575e-04 - val_loss: 0.0022
Epoch 25/100
12/12 [=====] - 2s 179ms/step - loss: 4.3007e-04 - val_loss: 0.0021
Epoch 26/100
12/12 [=====] - 2s 170ms/step - loss: 4.3581e-04 - val_loss: 0.0020
Epoch 27/100
12/12 [=====] - 2s 176ms/step - loss: 4.4547e-04 - val_loss: 0.0021
Epoch 28/100
12/12 [=====] - 2s 179ms/step - loss: 4.3156e-04 - val_loss: 0.0029
Epoch 29/100
12/12 [=====] - 2s 179ms/step - loss: 4.2649e-04 - val_loss: 0.0022
Epoch 30/100
12/12 [=====] - 2s 181ms/step - loss: 3.8626e-04 - val_loss: 0.0018
Epoch 31/100
12/12 [=====] - 2s 184ms/step - loss: 3.6965e-04 - val_loss: 0.0019
Epoch 32/100
12/12 [=====] - 2s 180ms/step - loss: 3.7820e-04 - val_loss: 0.0017
Epoch 33/100
12/12 [=====] - 2s 171ms/step - loss: 3.7708e-04 - val_loss: 0.0017
Epoch 34/100
12/12 [=====] - 2s 179ms/step - loss: 3.4536e-04 - val_loss: 0.0016
Epoch 35/100
12/12 [=====] - 2s 178ms/step - loss: 3.4536e-04 - val_loss: 0.0016
Epoch 36/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 37/100
12/12 [=====] - 2s 173ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 38/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 39/100
12/12 [=====] - 2s 179ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 40/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 41/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 42/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 43/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 44/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 45/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
Epoch 46/100
12/12 [=====] - 2s 179ms/step - loss: 3.4529e-04 - val_loss: 0.0016
Epoch 47/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 48/100
12/12 [=====] - 2s 179ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 49/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 50/100
12/12 [=====] - 2s 166ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 51/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 52/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 53/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 54/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 55/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 56/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
Epoch 57/100
12/12 [=====] - 2s 179ms/step - loss: 3.4529e-04 - val_loss: 0.0016
Epoch 58/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 59/100
12/12 [=====] - 2s 179ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 60/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 61/100
12/12 [=====] - 2s 166ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 62/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 63/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 64/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 65/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 66/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 67/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
Epoch 68/100
12/12 [=====] - 2s 179ms/step - loss: 3.4529e-04 - val_loss: 0.0016
Epoch 69/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 70/100
12/12 [=====] - 2s 179ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 71/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 72/100
12/12 [=====] - 2s 166ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 73/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 74/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 75/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 76/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 77/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 78/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
Epoch 79/100
12/12 [=====] - 2s 179ms/step - loss: 3.4529e-04 - val_loss: 0.0016
Epoch 80/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 81/100
12/12 [=====] - 2s 179ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 82/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 83/100
12/12 [=====] - 2s 166ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 84/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 85/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 86/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 87/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 88/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 89/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
Epoch 90/100
12/12 [=====] - 2s 179ms/step - loss: 3.4529e-04 - val_loss: 0.0016
Epoch 91/100
12/12 [=====] - 2s 173ms/step - loss: 3.4784e-04 - val_loss: 0.0016
Epoch 92/100
12/12 [=====] - 2s 179ms/step - loss: 3.5223e-04 - val_loss: 0.0019
Epoch 93/100
12/12 [=====] - 2s 167ms/step - loss: 3.5031e-04 - val_loss: 0.0015
Epoch 94/100
12/12 [=====] - 2s 166ms/step - loss: 3.4396e-04 - val_loss: 0.0014
Epoch 95/100
12/12 [=====] - 2s 168ms/step - loss: 3.4100e-04 - val_loss: 0.0014
Epoch 96/100
12/12 [=====] - 2s 179ms/step - loss: 3.4636e-04 - val_loss: 0.0026
Epoch 97/100
12/12 [=====] - 2s 179ms/step - loss: 3.4035e-04 - val_loss: 0.0016
Epoch 98/100
12/12 [=====] - 2s 171ms/step - loss: 3.3876e-04 - val_loss: 0.0020
Epoch 99/100
12/12 [=====] - 2s 181ms/step - loss: 3.0435e-04 - val_loss: 0.0014
Epoch 100/100
12/12 [=====] - 2s 179ms/step - loss: 3.2676e-04 - val_loss: 0.0014
```

Out[28]: <tensorflow.python.keras.callbacks.History at 0x27f8cf8cf10>

```
In [29]: #Prediction
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

```
In [32]: #Transformback to original form
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)
```

```
In [33]: import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))
```

Out[33]: 141.86542895832955

```
In [34]: math.sqrt(mean_squared_error(y_test,test_predict))
```

Out[34]: 238.0848566790705

```
In [38]: look_back=100
trainPredictPlot = np.empty_like(df1)
trainPredictPlot[:,0]=np.nan
trainPredictPlot[:(len(train_predict)-look_back,1)]= train_predict
#plot test prediction for plotting
testPredictPlot = np.empty_like(df1)
testPredictPlot[:,0]= np.nan
testPredictPlot[(len(train_predict)-look_back*2):len(df1)-1,1] = test_predict
#plot baseline and prediction
plt.plot(scaler.inverse_transform(df1))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```



In [] :