

## Creating A Dataset With Sliding Windows

A special data structure is needed to cover n-time stamps, based on which LSTM will predict the n + 1 tt price. Here the number of past timestamps is set to 10.

The function takes two arguments, the dataset which is a NumPy array that we want to convert into a dataset and the time\_step which is the number of previous time steps to use as input variables to predict the next time period, in this case, defaulted to 1.

This default will create a dataset where X is the price of crude oil at a given time (t) and Y is the price of crude oil at the next time (t + 1).

```
# convert an array of values into a dataset matrix
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]    ###i=0, 0,1,2,3-----99    100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)
```

We are applying the function on training data and test data. Hence we get X\_train,y\_train and X\_test,ytest

```
# reshape into X=t,t+1,t+2,t+3 and Y=t+4
time_step = 10
X_train, y_train = create_dataset(train_data, time_step)
X_test, ytest = create_dataset(test_data, time_step)
```

The shape of training data

```
print(X_train.shape), print(y_train.shape)

(5329, 10)
(5329,)
```

The shape of test data

```
print(X_test.shape), print(ytest.shape)
```

```
(2865, 10)  
(2865,)
```

The data of X\_train is as follows

**X\_train**

```
array([[0.11335703, 0.11661484, 0.12053902, ..., 0.10980305, 0.1089886 ,  
        0.11054346],  
       [0.11661484, 0.12053902, 0.11550422, ..., 0.1089886 , 0.11054346,  
        0.10165852],  
       [0.12053902, 0.11550422, 0.1156523 , ..., 0.11054346, 0.10165852,  
        0.09906708],  
       ...,  
       [0.36731823, 0.35176958, 0.36080261, ..., 0.36391234, 0.37042796,  
        0.37042796],  
       [0.35176958, 0.36080261, 0.35354657, ..., 0.37042796, 0.37042796,  
        0.37879461],  
       [0.36080261, 0.35354657, 0.35295424, ..., 0.37042796, 0.37879461,  
        0.37916482]])
```

For LSTM model , it is necessary to reshape the X\_train and X\_test into 3 dimensional array before building the model.

# reshape input to be [samples, time steps, features] which is required for LSTM

```
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)
```