MODEL BUILDING-ADDING OUTPUT LAYERS

Team ID	PNT2022TMID47669
Project Name	Crude Oil Price Prediction

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
data=pd.read_excel("/content/Crude Oil Prices Daily.xlsx")
```

In [3]:
```
data.isnull().any()
```

Out[3]:
```
Date             False
Closing Value     True
dtype: bool
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
Date             0
Closing Value    7
dtype: int64
```

In [5]:
```
data.dropna(axis=0,inplace=True)
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Date             0
Closing Value    0
dtype: int64
```

In [7]:
```
data_oil=data.reset_index()['Closing Value']
data_oil
```

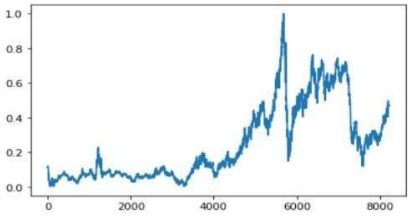
Out[7]:
```
0       25.56
1       26.00
2       26.53

4       25.87
        ...
8211    73.89
8212    74.19
8213    73.05
8214    73.78
8215    73.93
Name: Closing Value, Length: 8216, dtype: float64
```

In [8]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data_oil=scaler.fit_transform(np.array(data_oil).reshape(-1,1))
```

In [9]:
```
data_oil
```

Out[9]:
```
array([[0.11335703],
       [0.11661484],
       [0.12053902],
       ...,
       [0.46497853],
       [0.47038353],
       [0.47149415]])
```

In [10]:
```
plt.plot(data_oil)
```



In [11]:
```
training_size=int(len(data_oil)*0.65)
test_size=len(data_oil)-training_size
train_data,test_data=data_oil[0:training_size,:],data_oil[training_size:len(data_oil),:1]
```

In [12]:
```
training_size,test_size
```

Out[12]: (5340, 2876)

In [13]:
```
train_data.shape
```

Out[13]: (5340, 1)

In [14]:
```
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]
    dataX.append(a)
    dataY.append(dataset[i+time_step,0])
  return np.array(dataX),np.array(dataY)
```

In [12]:
```
training_size,test_size
```

Out[12]: (5340, 2876)

In [13]:
```
train_data.shape
```

Out[13]: (5340, 1)

In [14]:
```
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]
    dataX.append(a)
    dataY.append(dataset[i+time_step,0])
  return np.array(dataX),np.array(dataY)
```

In [15]:
```
time_step=10
x_train,y_train=create_dataset(train_data,time_step)
x_test,y_test=create_dataset(test_data,time_step)
```

In [16]:
```
print(x_train.shape),print(y_train.shape)
```

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

Out[16]: (None, None)

In [17]:
```
print(x_test.shape),print(y_test.shape)
```

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

Out[17]: (None, None)

```
a r ray ( [ [0 . J 133 3703,  0 . J 16 6148  , 0 . 1 Z0 5 3902,      . , 0. 109 80305 , 0. 1089886 ,
        0.1I05^3^6],
       [0.11661"B^, 0.12053902, 0.11530422,      , 0.108f1686, 0.110J"3^6,
        0.10165852],
       [0.12053902, 0.11550422, 0.1156523,      , 0.1105#3#6, 0.10165852,
        0.0f1906708],

       [0 . 3G 31823, 0 . 35175958, 0 . 3508026 J ,      , 0.363J123A, 0.370A2796,
        6. 37042796 ] ,
       [0 . 3 h176958,   0 . 3608026i , 0 . 3 h3h^657,      . , 0. 370^2796, 0. 370^2796,
        0 . 378798M ] ,
       [0 . 350802GJ ,  0 . 35358b57,  0 . 3529 5428,      . , 0. 37082796, 0. 37879461,
        0.37916482]])
```

```
x test= x te st . resh ape tx test . s hape [0 l,x te st . sh ap e l ] , l )
```

```
mom I en sor l o\v . ke ras . models in port Sequent iaJ
•finom I en sor I I a\v . ke ras . I a} ers ñnport Dens e
from tensorflQw.keras.layers import LSTh
```

```
mo d e I . ad ñ ( L STD ( SO, c ct ur n _s eq ue n ces -T rue , in out_s h a o e- ( 18, l ) )
model.add(LSTfl(50,retunn_sequen<es=True])
```

```
model.add(Dense(1))
```

```
m0deT.add(LSTN O,retur _sequences:True,inout_JaDe: 8,1)!)
```

```
mo de T . ad d (Dense (I )
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 10, 50)            10400
_________________________________________________________________
lstm_1 (LSTM)                (None, 10, 50)            20200
_________________________________________________________________
lstm_2 (LSTM)                (None, 50)                20288
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
*rainable params: 58,851
_________________________________________________________________
```