In [2]:
```
import requests
import io
url = r"https://github.com/IBM-EPBL/IBM-Project-51555-1660980405/blob/main/Data%20Collection/DataSet1986-2018.xlsx?raw=true"
download = requests.get(url).content
data = pd.read_excel(url,index_col=0,parse_dates=[0])
print(data.head())
```

```
            Closing Value
Date
1986-01-02          25.56
1986-01-03          26.00
1986-01-06          26.53
1986-01-07          25.85
1986-01-08          25.87
```

In [ ]:
```
```

In [3]:
```
data.isnull().any()
data.isnull().sum()
data.dropna(axis = 0, inplace = True)
data.isnull().sum()
data_oil = data.reset_index()['Closing Value']
data_oil
```

Out[3]:
```
0       25.56
1       26.00
2       26.53
3       25.85
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 [4]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler (feature_range=(0,1))
data_oil=scaler.fit_transform(np.array(data_oil).reshape(-1,1))
```

```
84/84 [==============================] - 3s 42ms/step - loss: 4.0582e-05 - val_loss: 1.9684e-04
Epoch 42/50
84/84 [==============================] - 3s 35ms/step - loss: 4.0487e-05 - val_loss: 1.9967e-04
Epoch 43/50
84/84 [==============================] - 3s 34ms/step - loss: 4.4339e-05 - val_loss: 2.0307e-04
Epoch 44/50
84/84 [==============================] - 3s 30ms/step - loss: 3.7815e-05 - val_loss: 2.1742e-04
Epoch 45/50
84/84 [==============================] - 3s 33ms/step - loss: 3.6488e-05 - val_loss: 2.2140e-04
Epoch 46/50
84/84 [==============================] - 3s 34ms/step - loss: 3.6334e-05 - val_loss: 2.1881e-04
Epoch 47/50
84/84 [==============================] - 3s 38ms/step - loss: 4.1092e-05 - val_loss: 4.8617e-04
Epoch 48/50
84/84 [==============================] - 3s 31ms/step - loss: 3.6932e-05 - val_loss: 2.4415e-04
Epoch 49/50
84/84 [==============================] - 3s 33ms/step - loss: 3.5177e-05 - val_loss: 1.8624e-04
Epoch 50/50
84/84 [==============================] - 3s 33ms/step - loss: 3.6481e-05 - val_loss: 2.0401e-04
```

Out[13]: <keras.callbacks.History at 0x7f7341326d30>

In [14]:
```
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)
train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)

import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))
```

Out[14]: 29.71311299049386

In [15]:
```
from tensorflow.keras.models import load_model
model.save("crude_oil.h5")
```

In [16]:
```
look_back=10
trainpredictPlot = np.empty_like(data_oil)
trainpredictPlot[:, :] = np.nan
trainpredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
```