Delivery Of Sprint1

Date	29 oct 2022
Team ID	PNT2022TMID13112
Project Name	Crude Oil Price Prediction

DATA COLLECTION

Dataset:

Crude oil daily price dataset is used in the project.

Dataset is included in the drive link.

https://drive.google.com/drive/folders/1WxVAVBfu3wL8jHlMJeeBOdL1Wtyf78DL

DATA PREPROCESSING

DATA PREPROCESSING

Importing the libraries

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

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

In [3]: data.head()
```

Out[3]:

	Date	Closing Value
0	1986-01-02	25.56
1	1986-01-03	26.00
2	1986-01-06	26.53
3	1986-01-07	25.85
4	1986-01-08	25.87

Handling missing values

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

Out[4]:
```
Date             False
Closing Value     True
dtype: bool
```

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
Date             0
Closing Value    7
dtype: int64
```

In [6]:
```
data.dropna(axis=0,inplace=True)
```

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

In [8]:
```
data_oil
```

Out[8]:
```
0       25.56
1       26.00
2       26.53
3       25.85
4       25.87
```

In [9]:
```
data.isnull().any()
```

Out[9]:
```
Date             False
Closing Value    False
dtype: bool
```

Feature Scaling

In [10]:
```
from sklearn.preprocessing import MinMaxScaler
scalar=MinMaxScaler(feature_range=(0,1))
data_oil=scalar.fit_transform(np.array(data_oil).reshape(-1,1))
```

Splitting data into Train and Test Data

In [12]:
```
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 [13]:
```
training_size,test_size
```

Out[13]:
```
(5340, 2876)
```

In [14]:
```
train_data.shape
```

Out[14]:
```
(5340, 1)
```

Creating a dataset with sliding windows

In [15]:
```
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 [16]:
```
time_step = 10

X_train, y_train=create_dataset(train_data,time_step)

X_test, y_test = create_dataset(test_data,time_step)
```

In [17]:
```
print(X_train.shape),print(y_train.shape)
```

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

Out[17]:
```
(None, None)
```

In [18]:
```
print(X_test.shape),print(y_test.shape)
```

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

Out[18]: (None, None)

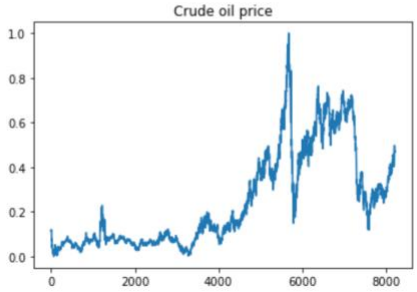
In [19]:
```
X_train
```

Out[19]:
```
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]])
```

Data Visualization

In [11]:
```
plt.title('Crude oil price')
plt.plot(data_oil)
```

Out[11]: [<matplotlib.lines.Line2D at 0x1bf71b03fa0>]



Splitting data into Train and Test Data

In [12]:
```
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 [13]:
```
training_size,test_size
```

Out[13]: (5340, 2876)

In [14]:
```
train_data.shape
```

Out[14]: (5340, 1)

Creating a dataset with sliding windows

In [15]:
```
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 [16]:
```
time_step = 10

X_train, y_train=create_dataset(train_data,time_step)

X_test, y_test = create_dataset(test_data,time_step)
```

In [17]:
```
print(X_train.shape),print(y_train.shape)
```

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

Out[17]: (None, None)

In [18]:
```
print(X_test.shape),print(y_test.shape)
```

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

Out[18]: (None, None)

In [19]:
```
X_train
```

Out[19]:
```
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]])
```

In [20]:
```
X_train.shape
```

Out[20]: (5329, 10)

In [21]:
```
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)
```