

```
In [5]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import datetime
from pylab import rcParams
import matplotlib.pyplot as plt
import warnings
import itertools
import statsmodels.api as sm
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from sklearn.metrics import mean_squared_error
from keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
import seaborn as sns
sns.set_context("paper", font_scale=1.3)
sns.set_style('white')
import math
from sklearn.preprocessing import MinMaxScaler
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import os
for dirname, __, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

**Activate Windows**  
Go to Settings to activate Windows.

IBM Watson Studio

Search in your workspaces

Buy

Reshma M's Account

Dallas

RM

Projects / Crude Oil Price Prediction / Crude Oil Price Prediction

Trusted | Python 3.9

FileEditViewInsertCellKernelHelp

Importing Data

In [7]:

```
import requests
import io
url = r"https://github.com/IBM-EPBL/IBM-Project-44530-1660725104/blob/main/IBM/Pre-Development/Data%20Collection/Crude%20oil%20Prices%20Daily.xlsx?raw=true"
download = requests.get(url).content
df = pd.read_excel(url,index_col=0,parse_dates=[0])
print (df.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 [8]:

```
#Sort dataset by column Date
df = df.sort_values('Date')
df = df.groupby('Date')['Closing Value'].sum().reset_index()
df.set_index('Date', inplace=True)
df=df.loc[datetime.date(year=2000,month=1,day=1):]
```

In [9]:

df.head()

Out[9]:

Closing Value	
Date	
2000-01-04	25.56
2000-01-05	26.53

Activate Windows

Go to Settings to activate Windows.

Type here to search

26°C Cloudy 3:53 PM 11/11/2022

\_\_\_\_\_

\_\_\_\_\_

Out[9]:

100

01-04

2000-01-05	24.65
------------	-------

2000-01-06 24.79

2000-01-07	24.79
------------	-------

2000-01-10	24.71
------------	-------

1000

```
# gives some infos on
tab.info() and DataF
```

```
tab_info = pd.DataFrame(df_initial.dtypes).T.rename(index={0:
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull()
```

```
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull().sum()).T.rename('initial'))
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull().sum()) / df_initial
```

[illegible]

```
rename(index={0: 'null values (%)'})
return tab.info
```

\_\_\_\_\_

Out[11]:

100

File Edit View Insert Cell Kernel Help Trusted Python 3.9 

```
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull().sum() / df_initial.shape[0] * 100).T.  
                                rename(index={0: 'null values (%)'}))  
return tab_info
```

```
In [11]: DfInfo(df)
```

Out[11]:

	Closing Value
column type	float64
null values (nb)	0
null values (%)	0.0

```
In [12]: df.index
```

```
Out[12]: DatetimeIndex(['2000-01-04', '2000-01-05', '2000-01-06', '2000-01-07',
                        '2000-01-10', '2000-01-11', '2000-01-12', '2000-01-13',
                        '2000-01-14', '2000-01-18',
                        ...,
                        '2018-06-26', '2018-06-27', '2018-06-28', '2018-06-29',
                        '2018-07-02', '2018-07-03', '2018-07-04', '2018-07-05',
                        '2018-07-06', '2018-07-09'],
                        dtype='datetime64[ns]', name='Date', length=4673, freq=None)
```

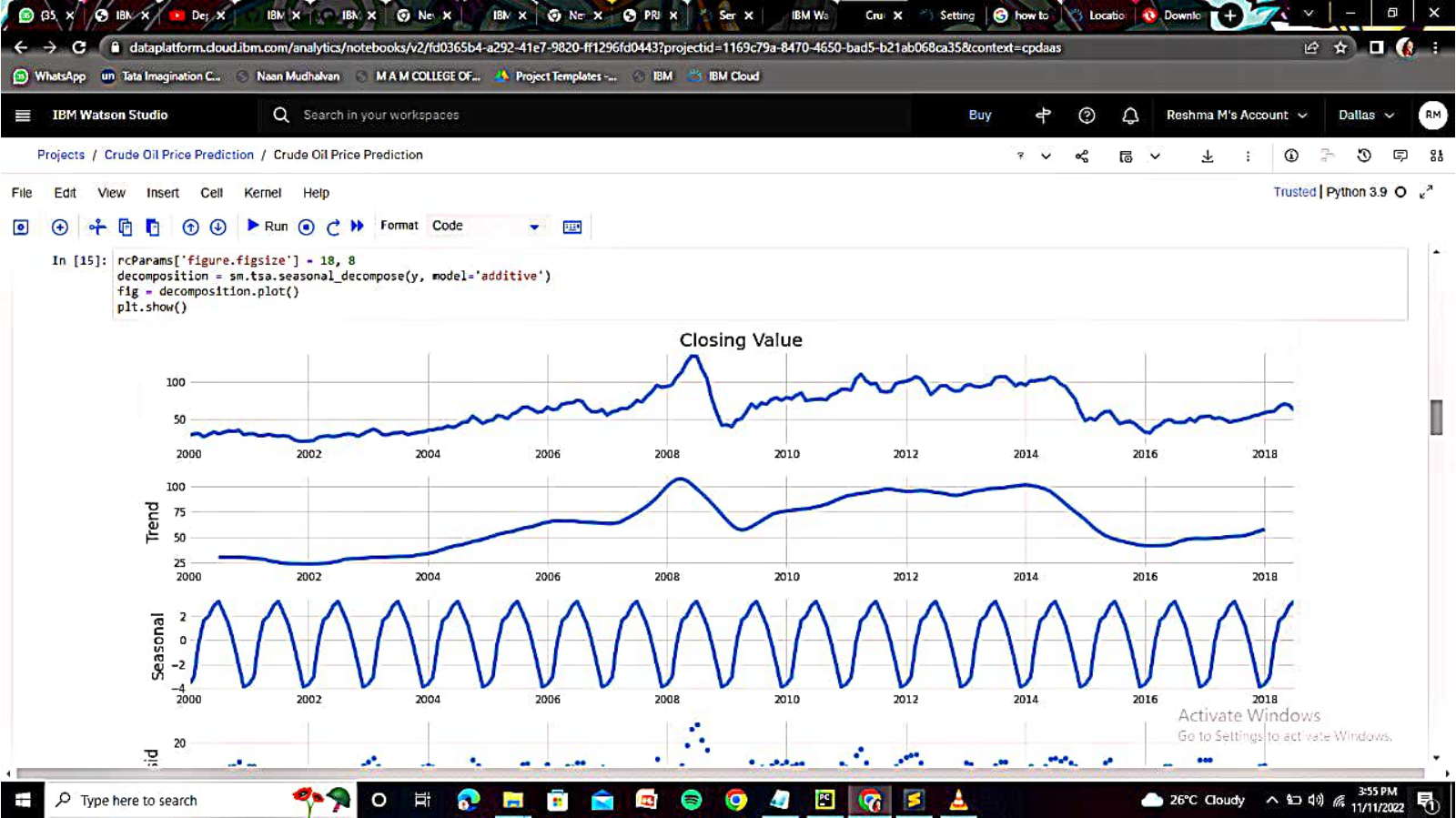
```
In [13]: y = df['Closing Value'].resample('MS').mean()
```

```
In [14]: y.plot(figsize=(15, 6))
plt.show()
```

**Activate Windows**  
Go to Settings to activate Windows.













IBM Watson Studio

Search in your workspaces

Buy

Reshma M's Account

Dallas

RM

Projects / Crude Oil Price Prediction / Crude Oil Price Prediction

Trusted | Python 3.9

File Edit View Insert Cell Kernel Help

Run

Format

Code

Epoch 9/20

212/212 [=====] - 17s 80ms/step - loss: 0.0026 - val\_loss: 0.0019 - lr: 1.0000e-04

Epoch 10/20

212/212 [=====] - 14s 68ms/step - loss: 0.0021 - val\_loss: 0.0017 - lr: 1.0000e-04

Epoch 11/20

212/212 [=====] - 15s 73ms/step - loss: 0.0019 - val\_loss: 0.0016 - lr: 1.0000e-04

Epoch 12/20

212/212 [=====] - 15s 69ms/step - loss: 0.0016 - val\_loss: 0.0015 - lr: 1.0000e-04

Epoch 13/20

212/212 [=====] - 15s 71ms/step - loss: 0.0015 - val\_loss: 0.0014 - lr: 1.0000e-04

Epoch 14/20

212/212 [=====] - 16s 74ms/step - loss: 0.0013 - val\_loss: 0.0014 - lr: 1.0000e-04

Epoch 15/20

212/212 [=====] - 15s 69ms/step - loss: 0.0012 - val\_loss: 0.0013 - lr: 1.0000e-04

Epoch 16/20

212/212 [=====] - 17s 79ms/step - loss: 0.0012 - val\_loss: 0.0013 - lr: 1.0000e-04

Epoch 17/20

212/212 [=====] - 15s 70ms/step - loss: 0.0011 - val\_loss: 0.0013 - lr: 1.0000e-04

Epoch 18/20

212/212 [=====] - 17s 78ms/step - loss: 0.0011 - val\_loss: 0.0013 - lr: 1.0000e-04

Epoch 19/20

212/212 [=====] - 18s 84ms/step - loss: 0.0011 - val\_loss: 0.0013 - lr: 1.0000e-04

Epoch 20/20

212/212 [=====] - 17s 80ms/step - loss: 0.0010 - val\_loss: 0.0013 - lr: 1.0000e-05

Model Training

In [21]: train\_predict = regressor.predict(X\_train)

test\_predict = regressor.predict(X\_test)

In [22]: train\_predict = sc.inverse\_transform(train\_predict)

Y\_train = sc.inverse\_transform(Y\_train)

Type here to search

26°C Cloudy

3:56 PM 11/11/2022