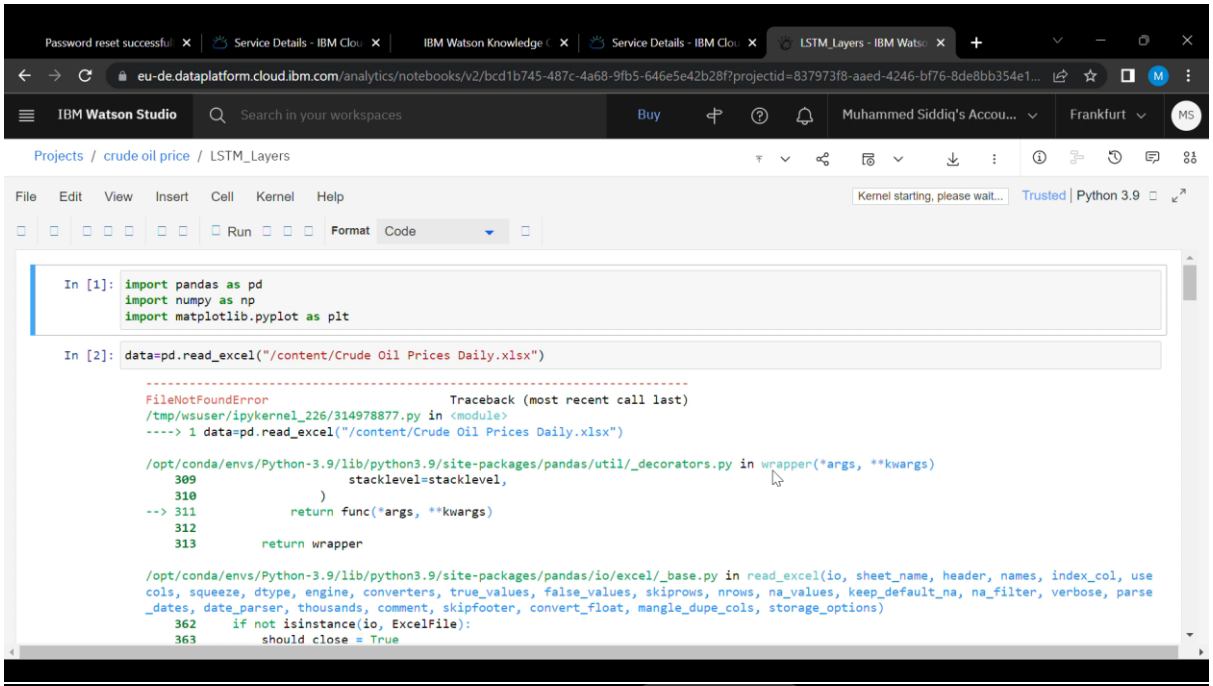
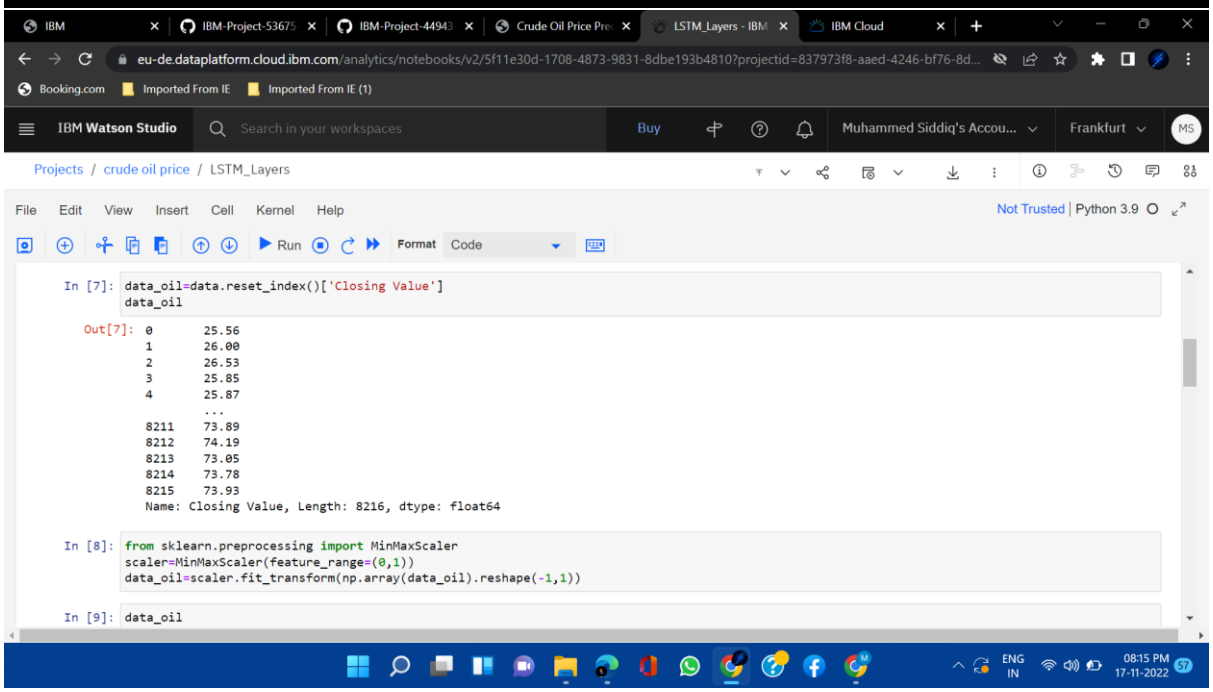

Password reset successful
Service Details - IBM Clou
IBM Watson Knowledge C
Service Details - IBM Clou
LSTM_Layers - IBM Watso
eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/bcd1b745-487c-4a68-9fb5-646e5e42b28f?projectid=837973f8-aaed-4246-bf76-8de8bb354e1...
IBM Watson Studio
Search in your workspaces
Buy
Muhammed Siddiq's Accou...
Frankfurt
MS
Projects / crude oil price / LSTM_Layers
File Edit View Insert Cell Kernel Help
Kernel starting, please wait...
Trusted | Python 3.9
Run
Format
Code
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")
FileNotFoundError Traceback (most recent call last)
/tmp/wsuser/ipykernel_226/314978877.py in <module>
----> 1 data=pd.read_excel("/content/Crude Oil Prices Daily.xlsx")
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/pandas/util/_decorators.py in wrapper(*args, **kwargs)
309 stacklevel=stacklevel,
310)
--> 311 return func(*args, **kwargs)
312
313 return wrapper
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/pandas/io/excel/_base.py in read_excel(io, sheet_name, header, names, index_col, use cols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse _dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols, storage_options)
362 if not isinstance(io, ExcelFile):
363 should_close = True


IBM
IBM-Project-53675
IBM-Project-44943
Crude Oil Price Pre
LSTM_Layers - IBM
IBM Cloud
eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/5f11e30d-1708-4873-9831-8dbe193b4810?projectid=837973f8-aaed-4246-bf76-8d...
Booking.com
Imported From IE
Imported From IE (1)
IBM Watson Studio
Search in your workspaces
Buy
Muhammed Siddiq's Accou...
Frankfurt
MS
Projects / crude oil price / LSTM_Layers
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format
Code
In [7]:
data_oil=data.reset_index()['Closing Value']
data_oil
Out[7]:
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 [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
ENG
IN
08:15 PM
17-11-2022

IBM Watson Studio

Projects / crude oil price / LSTM_Layers

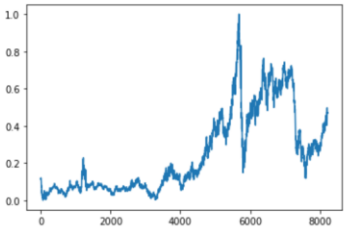
File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

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

Out[10]: [<matplotlib.lines.Line2D at 0x7f5a0c7b5850>]



In [11]:
```
training_size=int(len(data_oil)*0.65)
```

Projects / CRUDE OIL PRICE PREDICTION / PROJECT

MODEL TRAINING

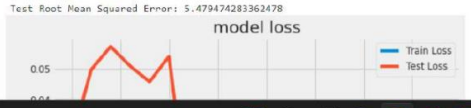
In [19]:
```
train_predict = regressor.predict(X_train)
test_predict = regressor.predict(X_test)
```

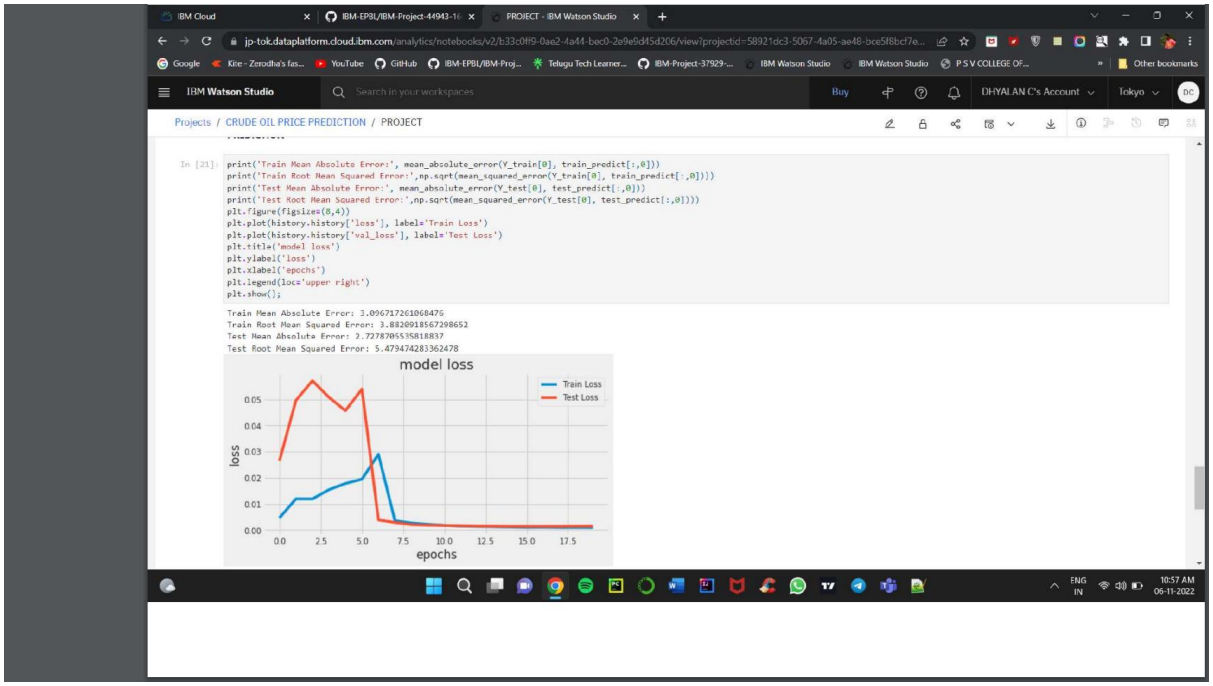
In [20]:
```
train_predict = sc.inverse_transform(train_predict)
Y_train = sc.inverse_transform([Y_train])
test_predict = sc.inverse_transform(test_predict)
Y_test = sc.inverse_transform([Y_test])
```

PREDICTION

In [21]:
```
print('Train Mean Absolute Error:', mean_absolute_error(Y_train[0], train_predict[:,0]))
print('Train Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_train[0], train_predict[:,0])))
print('Test Mean Absolute Error:', mean_absolute_error(Y_test[0], test_predict[:,0]))
print('Test Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_test[0], test_predict[:,0])))
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();
```

```
Train Mean Absolute Error: 3.096717261068476
Train Root Mean Squared Error: 3.8826918567298652
Test Mean Absolute Error: 2.7278705535818837
Test Root Mean Squared Error: 5.479474283362478
```




IBM Cloud
IBM-EPBL/IBM-Project-44943-16
PROJECT - IBM Watson Studio
jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/b33c0ff9-0ae2-4a44-bec0-2e9e9d45d206/view?projectid=58921dc3-5067-4a05-ae48-bce5f8bcf7e...
IBM Watson Studio
Search in your workspaces
Buy
DHYALAN C's Account
Tokyo
Projects / CRUDE OIL PRICE PREDICTION / PROJECT
In [21]:
print('Train Mean Absolute Error:', mean_absolute_error(Y_train[0], train_predict[:,0]))
print('Train Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_train[0], train_predict[:,0])))
print('Test Mean Absolute Error:', mean_absolute_error(Y_test[0], test_predict[:,0]))
print('Test Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_test[0], test_predict[:,0])))
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();
Train Mean Absolute Error: 3.096717261068476
Train Root Mean Squared Error: 3.8820918567298652
Test Mean Absolute Error: 2.7278705535818837
Test Root Mean Squared Error: 5.47947428362478
model loss
Train Loss
Test Loss
loss
epochs
0.00
0.01
0.02
0.03
0.04
0.05
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
ENG
IN
10:57 AM
06-11-2022