Metal_Options (1)

April 25, 2021

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split
```

[2]:
```
# Import required libraries
import pandas as pd
import numpy as np
from pandas.plotting import lag_plot
from pandas.plotting import autocorrelation_plot
from matplotlib import pyplot
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
import math as math
from scipy.stats import boxcox
from random import randrange
from random import seed
from random import random
from random import gauss
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[71]:
```
df = pd.read_csv("https://raw.githubusercontent.com/shauryashivam/
↪commodity-futures/main/Dataset/Gold5.csv?
↪token=AMF2Z3LBLLFT62EDL6NOAGDARINHK",header=0, index_col=0, parse_dates=True,
squeeze=True)
```

[4]:
```
df.drop(columns=['Open','High','Low','Adj Close','Volume'],axis=1,inplace=True)
```

[5]:
```
df.head()
```

[5]:
```
                     Close
Date
```

```
2016-01-04  1075.099976
2016-01-05  1078.400024
2016-01-06  1091.900024
2016-01-07  1107.699951
2016-01-08  1097.800049
```

[6]:
```
df.describe()
```

[6]:
```
             Close
count  1246.000000
mean   1388.680578
std     217.880280
min    1073.900024
25%    1249.000000
50%    1299.400024
75%    1482.649963
max    2051.500000
```

1 Single Lag

[7]:
```
var = pd.DataFrame(df.values)
dataframe = pd.concat([var.shift(1), var], axis=1)
dataframe.columns = ['t', 't+1']
print(dataframe.head(5))
```

```
             t          t+1
0          NaN  1075.099976
1  1075.099976  1078.400024
2  1078.400024  1091.900024
3  1091.900024  1107.699951
4  1107.699951  1097.800049
```

[8]:
```
var = pd.DataFrame(df.values)
dataframe = pd.concat([var.shift(3), var.shift(2), var.shift(1), var], axis=1)
dataframe.columns = ['t-2', 't-1', 't', 't+1']
print(dataframe.head(5))
```

```
           t-2          t-1            t          t+1
0          NaN          NaN          NaN  1075.099976
1          NaN          NaN  1075.099976  1078.400024
2          NaN  1075.099976  1078.400024  1091.900024
3  1075.099976  1078.400024  1091.900024  1107.699951
4  1078.400024  1091.900024  1107.699951  1097.800049
```

[9]:
```
var = pd.DataFrame(df.values)
shifted = var.shift(1)
```

```
window = shifted.rolling(window=2)
means = window.mean()
dataframe = pd.concat([means, var], axis=1)
dataframe.columns = ['mean(t-1,t)', 't+1']
print(dataframe.head(5))
```

```
   mean(t-1,t)          t+1
0          NaN  1075.099976
1          NaN  1078.400024
2  1076.750000  1091.900024
3  1085.150024  1107.699951
4  1099.799988  1097.800049
```

[10]:
```
var = pd.DataFrame(df.values)
window = var.expanding()
dataframe = pd.concat([window.min(), window.mean(), window.max(), var.
 ↪shift(-1)], axis=1)
dataframe.columns = ['min', 'mean', 'max', 't+1']
print(dataframe.head(5))
```

```
           min         mean          max          t+1
0  1075.099976  1075.099976  1075.099976  1078.400024
1  1075.099976  1076.750000  1078.400024  1091.900024
2  1075.099976  1081.800008  1091.900024  1107.699951
3  1075.099976  1088.274994  1107.699951  1097.800049
4  1075.099976  1090.180005  1107.699951  1096.500000
```

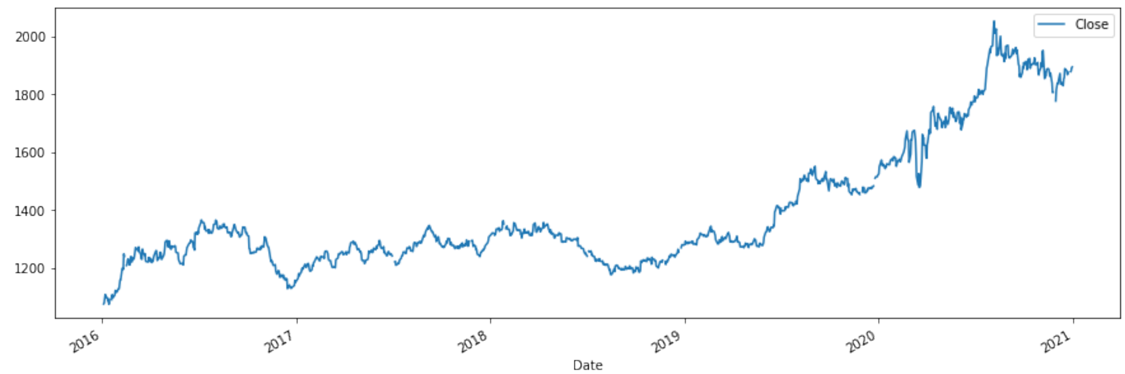
[11]:
```
dataframe = pd.DataFrame()
dataframe['month'] = [df.index[i].month for i in range(len(df))]
dataframe['day'] = [df.index[i].day for i in range(len(df))]
dataframe['Close'] = [df['Close'] for i in range(len(df))]
print(dataframe.head(5))
```

```
   month  day                                              Close
0      1    4  Date
2016-01-04    1075.099976
2016-01-05    1…
1      1    5  Date
2016-01-04    1075.099976
2016-01-05    1…
2      1    6  Date
2016-01-04    1075.099976
2016-01-05    1…
3      1    7  Date
2016-01-04    1075.099976
2016-01-05    1…
4      1    8  Date
2016-01-04    1075.099976
```

2016-01-05   1…

[12]:
```
df.plot(figsize=(15,5))
```

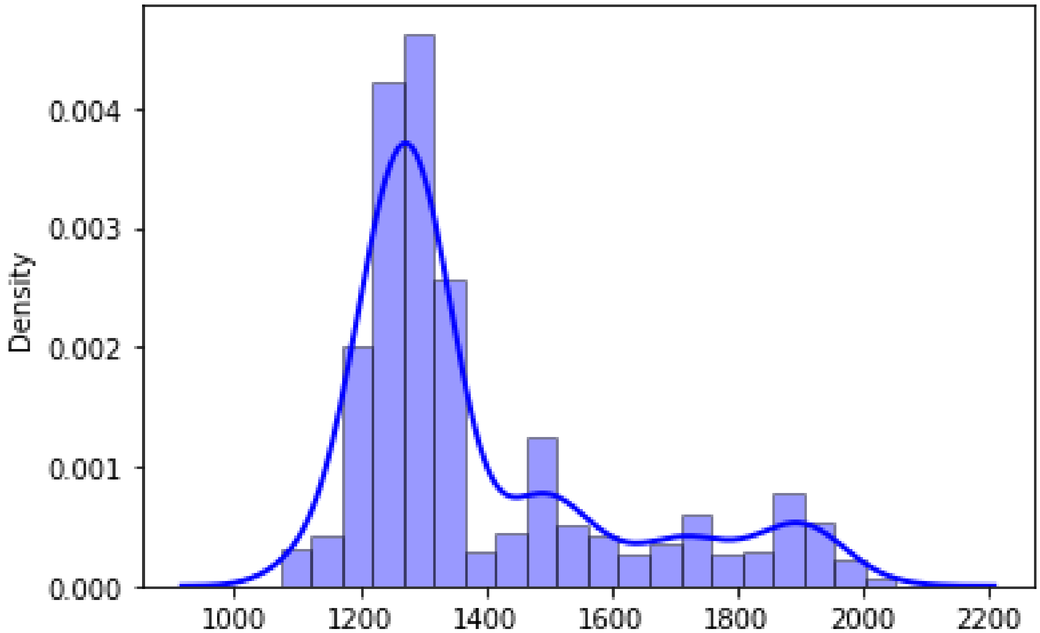
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a85a08f50>



[13]:
```
import seaborn as sns
sns.distplot(df, hist=True, kde=True,
             bins=20,
             color = 'blue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 2})
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a84bdf590>

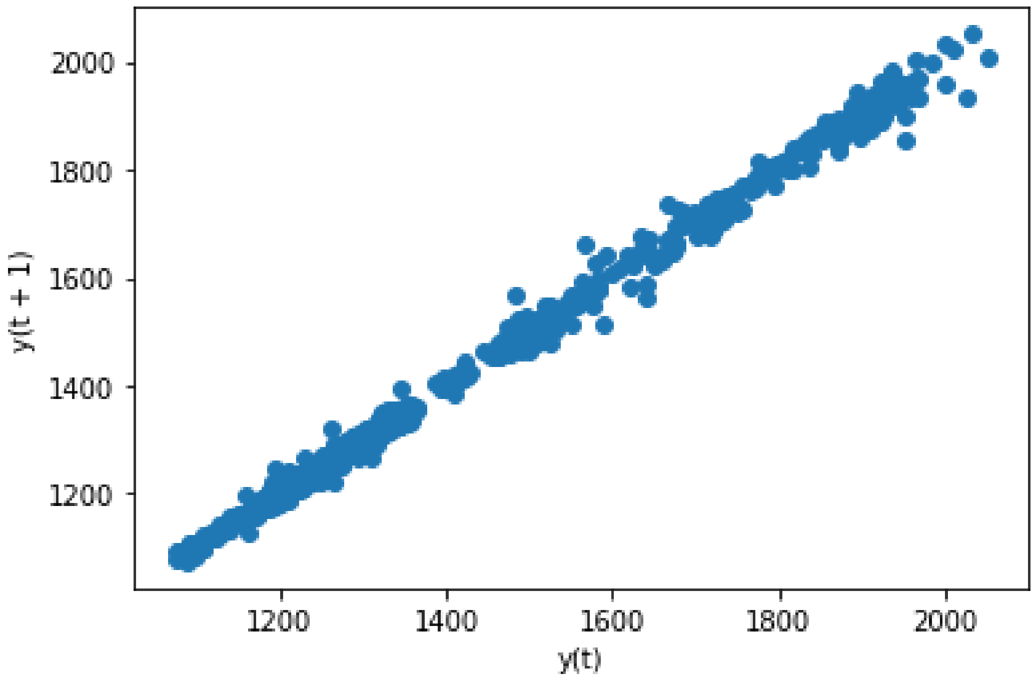


[14]:
```
df.head(
)
```

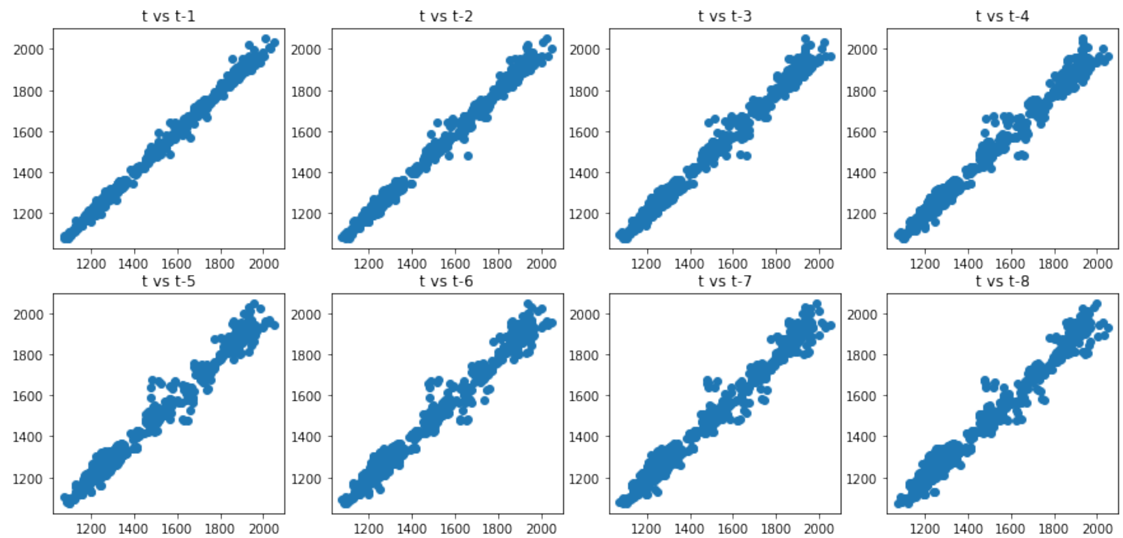
[14]:

Date	Close
2016-01-04	1075.099976
2016-01-05	1078.400024
2016-01-06	1091.900024
2016-01-07	1107.699951
2016-01-08	1097.800049

[16]:
```
lag_plot(df)
pyplot.show()
```

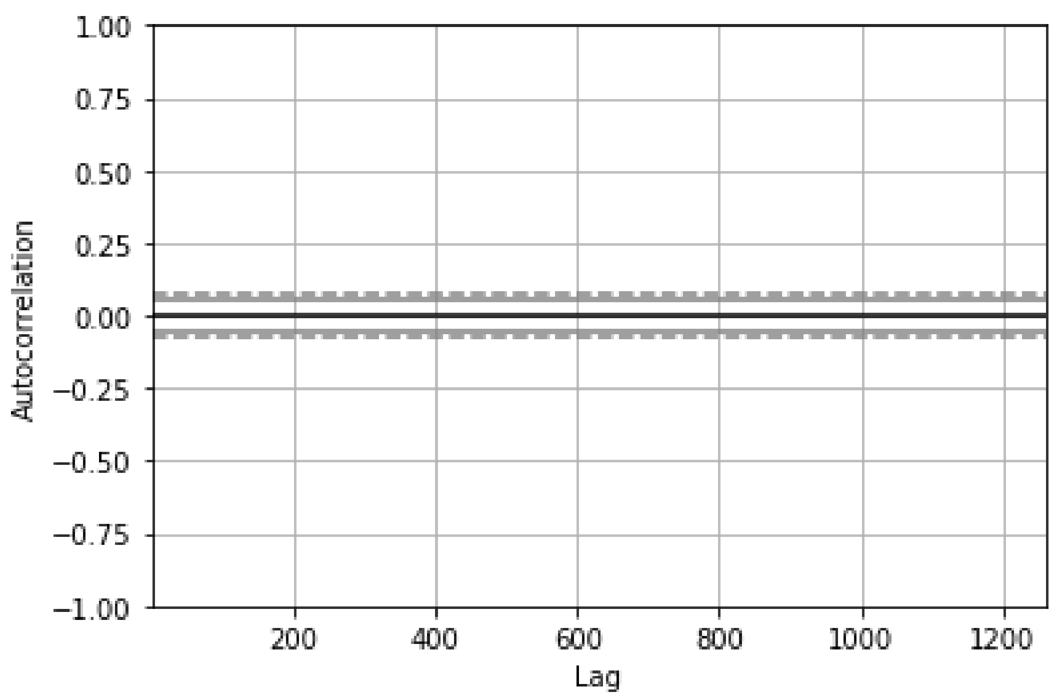


[17]:
```
values = pd.DataFrame(df.values)
lags = 8
columns = [values]
for i in range(1,(lags + 1)):
    columns.append(values.shift(i))
dataframe = pd.concat(columns, axis=1)
columns = ['t']
for i in range(1,(lags + 1)):
    columns.append('t-' + str(i))
dataframe.columns = columns
pyplot.figure(1,figsize=(15,7))
for i in range(1,(lags + 1)):
    ax = pyplot.subplot(240 + i)
    ax.set_title('t vs t-' + str(i))
    pyplot.scatter(x=dataframe['t'].values, y=dataframe['t-'+str(i)].values)
pyplot.show()
```




```
[18]: autocorrelation_plot(df)
```

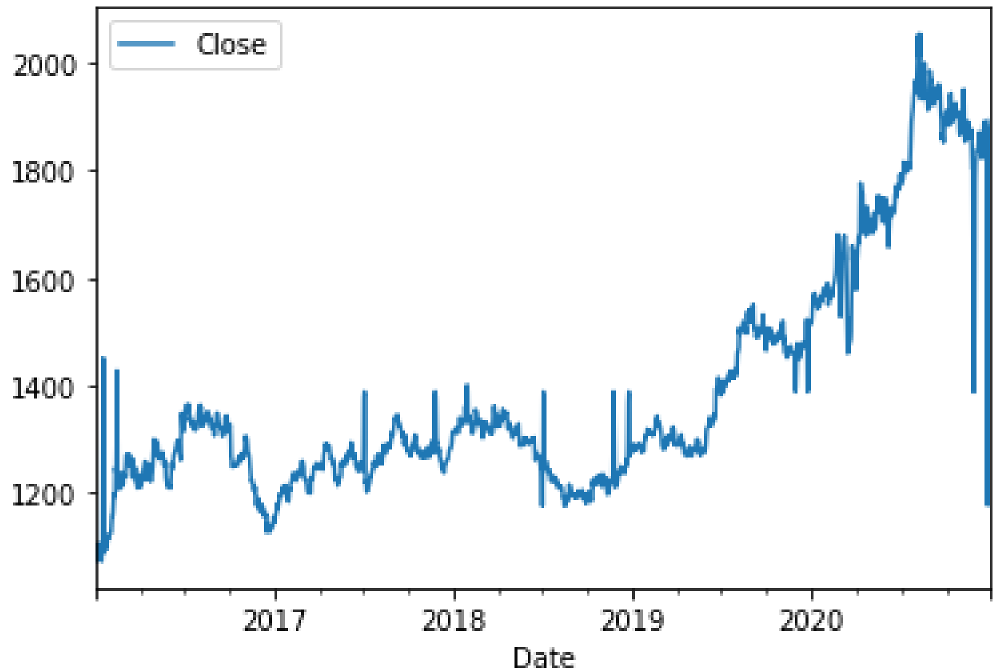
[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a81a2b810>




```
[19]: df=df.fillna(df.mean())
```

[23]:
```
upsampled = df.resample('D').mean()
interpolated = upsampled.interpolate(method='quadratic')
interpolated.plot()
```

[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a78f6e9d0>
```



[24]:
```
pip install mxnet
```

```
Collecting mxnet
  Downloading https://files.pythonhosted.org/packages/30/07/66174e78c12a30
48db9039aaa09553e35035ef3a008ba3e0ed8d2aa3c47b/mxnet-1.8.0.post0-py2.py3-none-
manylinux2014_x86_64.whl (46.9MB)
     |                  | 46.9MB 98kB/s
Collecting graphviz<0.9.0,>=0.8.1
  Downloading https://files.pythonhosted.org/packages/53/39/4ab213673844e0c004be
d8a0781a0721a3f6bb23eb8854ee75c236428892/graphviz-0.8.4-py2.py3-none-any.whl
Requirement already satisfied: requests<3,>=2.20.0 in
/usr/local/lib/python3.7/dist-packages (from mxnet) (2.23.0)
Requirement already satisfied: numpy<2.0.0,>1.16.0 in
/usr/local/lib/python3.7/dist-packages (from mxnet) (1.19.5)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet)
(1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests<3,>=2.20.0->mxnet) (2.10)
```

```
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet)
(2020.12.5)
Installing collected packages: graphviz, mxnet
  Found existing installation: graphviz 0.10.1
    Uninstalling graphviz-0.10.1:
      Successfully uninstalled graphviz-0.10.1
Successfully installed graphviz-0.8.4 mxnet-1.8.0.post0
```

[25]:
```
import time
import numpy as np

from mxnet import nd, autograd, gluon
from mxnet.gluon import nn, rnn
import mxnet as mx
import datetime
import seaborn as sns

import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.decomposition import PCA

import math

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler

import xgboost as xgb
from sklearn.metrics import accuracy_score
```

[26]:
```
df.describe()
```

[26]:

	Close
count	1259.000000
mean	1388.680578
std	216.751584
min	1073.900024
25%	1249.250000
50%	1301.500000
75%	1481.849976
max	2051.500000

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(
...     X, y, test_size=0.15, random_state=42)
```

```
[ ]: model = tf.keras.Sequential()
```

```
[ ]: model.add(tf.keras.layers.
      ↪Conv1D(kernel_size=64,filters=3*3,strides=1,activation="relu"))
     model.add(tf.keras.layers.MaxPooling1D(pool_size=2))
```

```
[ ]: model.summary()
```

2 BUILDING MODEL

2.1 LSTM-GRU

```
[66]: model = tf.keras.Sequential()
      model.add(tf.keras.layers.GRU(5,activation = 'relu', input_shape=(1,1)))
      model.add(Dense(100,activation='relu'))
      model.add(Dense(1))
      model.compile(loss='mse',optimizer='adam',metrics=['mae'])
```

2.2 LSTM CNN Attention

```
[31]: !pip install keras-attention
      !pip install keras-self-attention
```

```
Requirement already satisfied: keras-attention in /usr/local/lib/python3.7/dist-
packages (1.0.0)
Requirement already satisfied: keras in /usr/local/lib/python3.7/dist-packages
(from keras-attention) (2.4.3)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-
packages (from keras->keras-attention) (1.19.5)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-
packages (from keras->keras-attention) (1.4.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages
(from keras->keras-attention) (2.10.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from keras->keras-attention) (3.13)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from h5py->keras->keras-attention) (1.15.0)
Collecting keras-self-attention
  Downloading https://files.pythonhosted.org/packages/c3/34/e21dc6adcdab2be03781
bde78c6c5d2b2136d35a1dd3e692d7e160ba062a/keras-self-attention-0.49.0.tar.gz
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from keras-self-attention) (1.19.5)
Requirement already satisfied: Keras in /usr/local/lib/python3.7/dist-packages
(from keras-self-attention) (2.4.3)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages
(from Keras->keras-self-attention) (2.10.0)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-
```

```
packages (from Keras->keras-self-attention) (1.4.1)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from Keras->keras-self-attention) (3.13)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from h5py->Keras->keras-self-attention) (1.15.0)
Building wheels for collected packages: keras-self-attention
  Building wheel for keras-self-attention (setup.py) … done
  Created wheel for keras-self-attention:
filename=keras_self_attention-0.49.0-cp37-none-any.whl size=19468
sha256=08334fe68e10170a4bbd13359e0d4d4d6d9b2a891df99e311412a43047eb7aff
  Stored in directory: /root/.cache/pip/wheels/6f/9d/c5/26693a5092d9313daeae94db
04818fc0a2b7a48ea381989f34
Successfully built keras-self-attention
Installing collected packages: keras-self-attention
Successfully installed keras-self-attention-0.49.0
```

[67]:
```
import os
import time
import warnings
import numpy as np
import pandas as pd
import operator
from functools import reduce
import h5py
from numpy import newaxis
from keras.layers.core import Dense, Activation, Dropout
from keras.layers import Convolution1D, MaxPooling1D, Flatten, ␣
↪Embedding,Bidirectional, GRU
from keras.layers import Conv1D, GlobalMaxPooling1D, merge
from keras.layers.recurrent import LSTM
from keras.models import Sequential
from keras_self_attention import SeqSelfAttention
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, Normalizer
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
```

[47]:
```
def create_dataset(dataset, look_back=1, columns = ['Close']):
    dataX, dataY = [], []
    for i in range(len(dataset.index)):
        if i < look_back:
            continue
        a = None
        for c in columns:
            b = dataset.loc[dataset.index[i-look_back:i], c].to_numpy()
            if a is None:
```

```
                a = b
            else:
                a = np.append(a,b)
        dataX.append(a)
        dataY.append(dataset.loc[dataset.index[i-look_back], columns].
 ↪to_numpy())
    return np.array(dataX), np.array(dataY)
```

[68]:
```
look_back = 7 # 10, 13
sc = StandardScaler()
df.loc[:, 'Close'] = sc.fit_transform(df.Close.values.reshape(-1,1)) # fit.
 ↪transform()
print(df.loc[:, 'Close'])

# Create training data
#train_df = df.loc[df.index < pd.to_datetime('2010-01-01')]
train_df = df.loc[df.index < df.index[int(len(df.index)*0.8)]]
train_x, train_y = create_dataset(train_df, look_back=look_back)


# Construct the whole LSTM + CNN
model = Sequential()
# LSTM
model.add(GRU(6,input_shape = (look_back, 1), input_dim=1 , ␣
 ↪return_sequences=True))

#model.add(LSTM(input_shape = (look_back,1), input_dim=1, output_dim=6,␣
 ↪return_sequences=True))
#model.add(Dense(1))
#model.add(Activation('relu')) # ReLU : y = max(0,x)

# Attention Mechanism
model.add(SeqSelfAttention(attention_activation='sigmoid', name='Attention'))

# CNN
model.add(Convolution1D(input_shape = (look_back,1),
                        filters=64,# 32,128
                       kernel_size=2,
                       activation='relu',
                        ))
#model.add(MaxPooling1D(pool_length=2))

'''model.add(Convolution1D(input_shape = (look_back,1),
                        nb_filter=64,
                        filter_length=2,
                        border_mode='valid',
                        activation='relu',
```

```
                        subsample_length=1))'''
model.add(MaxPooling1D(pool_size=(2)))

model.add(Dropout(0.25))

#model.add(Dense(250))
#model.add(Dropout(0.25,input_shape=(2,)))
model.add(Activation('relu')) # ReLU : y = max(0,x)
model.add(Dense(1))
model.add(Activation('linear')) # Linear : y = x

# Print whole structure of the model
print(model.summary())

# training the train data with n epoch
model.compile(loss="mse", optimizer="adam") # adam, rmsprop
result = model.fit(np.atleast_3d(np.array(train_x)),
          np.atleast_3d(train_y),
          epochs=100,
          batch_size=80, verbose=1, shuffle=False)


with open('data_lstm_attention_cnn_palladium.txt','w') as f:
    f.write(str(result.history))

model.save('lstm_attention_cnn_palladium.h5')


# Make prediction and specify on the line chart
predictors = ['Close']
df['Pred'] = df.loc[df.index[0], 'Close']
for i in range(len(df.index)):
    if i < look_back:
        continue
    a = None
    for c in predictors:
        b = df.loc[df.index[i-look_back:i], c].to_numpy()
        if a is None:
            a = b
        else:
            a = np.append(a,b)
        a = a
    y = model.predict(a.reshape(1,look_back*len(predictors),1))
    df.loc[df.index[i], 'Pred']=y[0][0]

df.loc[:, 'Close'] = sc.inverse_transform(df.loc[:, 'Close'])
df.loc[:, 'Pred'] = sc.inverse_transform(df.loc[:, 'Pred'])
```

```
def mape(y_true, y_pred):
    n = len(y_true)
    mape = sum(np.abs((y_true - y_pred) / y_true)) / n * 100
    return mape

# present the line chart and some parameters like MSE, which reflects the␣
↪accuracy of the model in sample or out sample
plt.grid(ls='--')
plt.plot(df.loc[df.index < df.index[int(len(df.index)*0.8)], 'Pred'], 'orange',␣
↪label = 'Insample Prediction')
plt.plot(df.loc[df.index >= df.index[int(len(df.index)*0.8)], 'Pred'], 'g',␣
↪label = 'Outsample Prediction')
plt.plot(df.Close ,'b', label = 'Price')
plt.xlabel('Date')
plt.ylabel('Closing Price')
#print('%e'%mean_squared_error(df.loc[df.index < pd.
↪to_datetime('2010-01-01'),'Close'],df.loc[df.index < pd.
↪to_datetime('2010-01-01'),'Pred']))
#print('%e'%mean_squared_error(df.loc[df.index >= pd.
↪to_datetime('2010-01-01'),'Close'],df.loc[df.index >= pd.
↪to_datetime('2010-01-01'),'Pred']))
print('The RMSE is ','%e'%sqrt(mean_squared_error(df.loc[df.index >= df.
↪index[int(len(df.index)*0.8)], 'Close'], df.loc[df.index >= df.
↪index[int(len(df.index)*0.8)], 'Pred'])))
print('The RMAE is ','%e'%sqrt(mean_absolute_error(df.loc[df.index >= df.
↪index[int(len(df.index)*0.8)], 'Close'], df.loc[df.index >= df.
↪index[int(len(df.index)*0.8)], 'Pred'])))
print('The MAPE is ','%e'%mape(df.loc[df.index >= df.index[int(len(df.index)*0.
↪8)], 'Close'], df.loc[df.index >= df.index[int(len(df.index)*0.8)], 'Pred']))

plt.legend()
plt.savefig("lstm_attention_cnn_palladium.eps", format='eps', dpi=1000)
plt.show()
```

```
Date
2016-01-04   -1.447303e+00
2016-01-05   -1.432072e+00
2016-01-06   -1.369764e+00
2016-01-07   -1.296841e+00
2016-01-08   -1.342533e+00
                  …
2020-12-24   -8.203802e-15
2020-12-28    2.254717e+00
2020-12-29    2.266256e+00
2020-12-30    2.318410e+00
2020-12-31    2.328102e+00
```

```
Name: Close, Length: 1259, dtype: float64
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_1 (GRU)                  (None, 7, 6)              162
_________________________________________________________________
Attention (SeqSelfAttention) (None, None, 6)           449
_________________________________________________________________
conv1d_6 (Conv1D)            (None, None, 64)          832
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, None, 64)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, None, 64)          0
_________________________________________________________________
activation_8 (Activation)    (None, None, 64)          0
_________________________________________________________________
dense_6 (Dense)              (None, None, 1)           65
_________________________________________________________________
activation_9 (Activation)    (None, None, 1)           0
=================================================================
Total params: 1,508
Trainable params: 1,508
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/100
13/13 [==============================] - 3s 8ms/step - loss: 0.5087
Epoch 2/100
13/13 [==============================] - 0s 9ms/step - loss: 0.2243
Epoch 3/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0800
Epoch 4/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0439
Epoch 5/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0343
Epoch 6/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0306
Epoch 7/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0317
Epoch 8/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0299
Epoch 9/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0266
Epoch 10/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0237
Epoch 11/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0260
```

```
Epoch 12/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0235
Epoch 13/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0232
Epoch 14/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0227
Epoch 15/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0229
Epoch 16/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0233
Epoch 17/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0213
Epoch 18/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0227
Epoch 19/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0223
Epoch 20/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0210
Epoch 21/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0196
Epoch 22/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0222
Epoch 23/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0201
Epoch 24/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0214
Epoch 25/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0189
Epoch 26/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0218
Epoch 27/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0183
Epoch 28/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0176
Epoch 29/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0183
Epoch 30/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0182
Epoch 31/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0187
Epoch 32/100
13/13 [==============================] - 0s 11ms/step - loss: 0.0187
Epoch 33/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0200
Epoch 34/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0178
Epoch 35/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0186
```

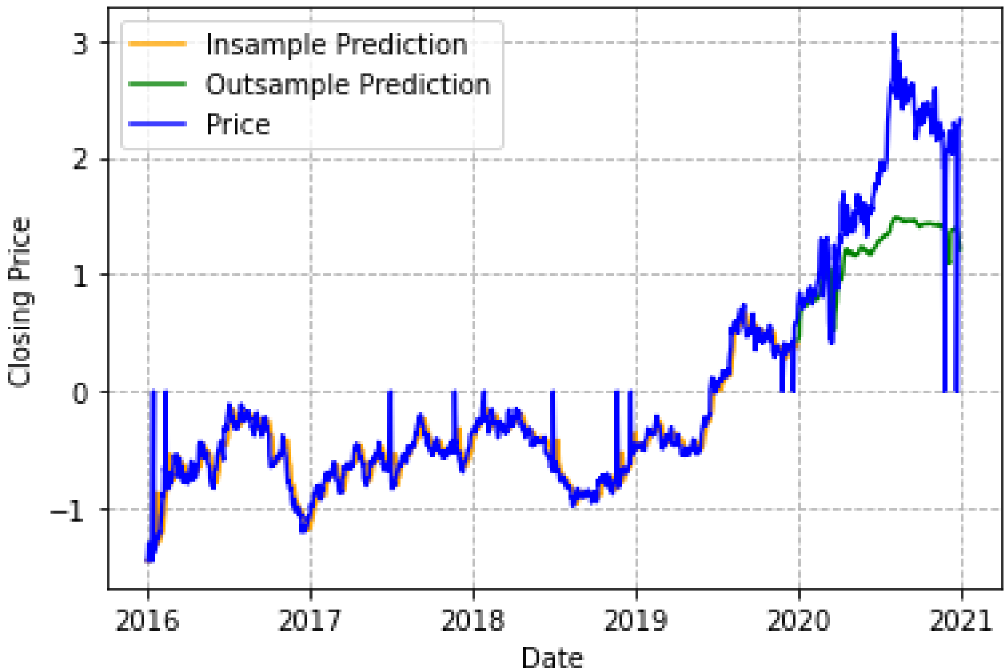
```
Epoch 36/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0170
Epoch 37/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0175
Epoch 38/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0181
Epoch 39/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0165
Epoch 40/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0173
Epoch 41/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0176
Epoch 42/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0168
Epoch 43/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0157
Epoch 44/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0165
Epoch 45/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0166
Epoch 46/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0152
Epoch 47/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0152
Epoch 48/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0152
Epoch 49/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0145
Epoch 50/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0136
Epoch 51/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0156
Epoch 52/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0141
Epoch 53/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0153
Epoch 54/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0146
Epoch 55/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0160
Epoch 56/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0144
Epoch 57/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0134
Epoch 58/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0143
Epoch 59/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0145
```

```
Epoch 60/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0129
Epoch 61/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0154
Epoch 62/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0144
Epoch 63/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0143
Epoch 64/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0144
Epoch 65/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0136
Epoch 66/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0130
Epoch 67/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0132
Epoch 68/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0136
Epoch 69/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0131
Epoch 70/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0131
Epoch 71/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0131
Epoch 72/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0136
Epoch 73/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0125
Epoch 74/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0131
Epoch 75/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0126
Epoch 76/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0132
Epoch 77/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0134
Epoch 78/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0122
Epoch 79/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0128
Epoch 80/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0130
Epoch 81/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0141
Epoch 82/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0137
Epoch 83/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0136
```

```
Epoch 84/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0151
Epoch 85/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0135
Epoch 86/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0131
Epoch 87/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0127
Epoch 88/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0123
Epoch 89/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0136
Epoch 90/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0133
Epoch 91/100
13/13 [==============================] - 0s 8ms/step - loss: 0.0128
Epoch 92/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0128
Epoch 93/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0133
Epoch 94/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0116
Epoch 95/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0122
Epoch 96/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0127
Epoch 97/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0125
Epoch 98/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0125
Epoch 99/100
13/13 [==============================] - 0s 10ms/step - loss: 0.0128
Epoch 100/100
13/13 [==============================] - 0s 9ms/step - loss: 0.0119
```

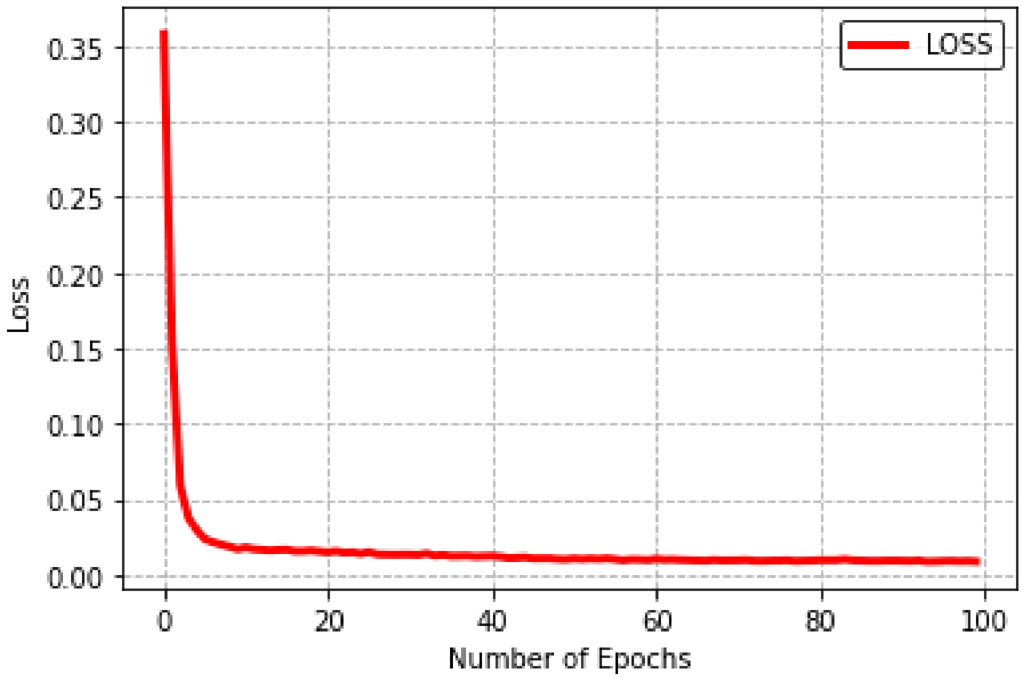
```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```

```
The RMSE is  7.236283e-01
The RMAE is  7.819518e-01
The MAPE is  1.353704e+14
```



[69]:
```
# sketch loss
#plt.cla() # clear the axis
plt.grid(ls='--')
plt.plot(result.epoch,result.history['loss'],label='LOSS',c='r',lw=3)
#plt.scatter(result.epoch,result.history['loss'],s=15,c='r')
plt.xlabel('Number of Epochs')
plt.ylabel('Loss')
plt.legend(loc='upper right', frameon=True, edgecolor='black')
plt.savefig("LC_loss.eps", format='eps', dpi=1000)
plt. close(0)
```

```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```


LOSS
Loss
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
Number of Epochs

[69]: