Copy of stock price predection

April 24, 2022

0.1 Importing Libraries

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

[2]:
```
np.random.seed(42)
```

0.2 Loading Our Dataset

[3]:
```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

[4]:
```
df=pd.read_csv("gdrive/My Drive/QCOM-4.csv")
df.head()
```

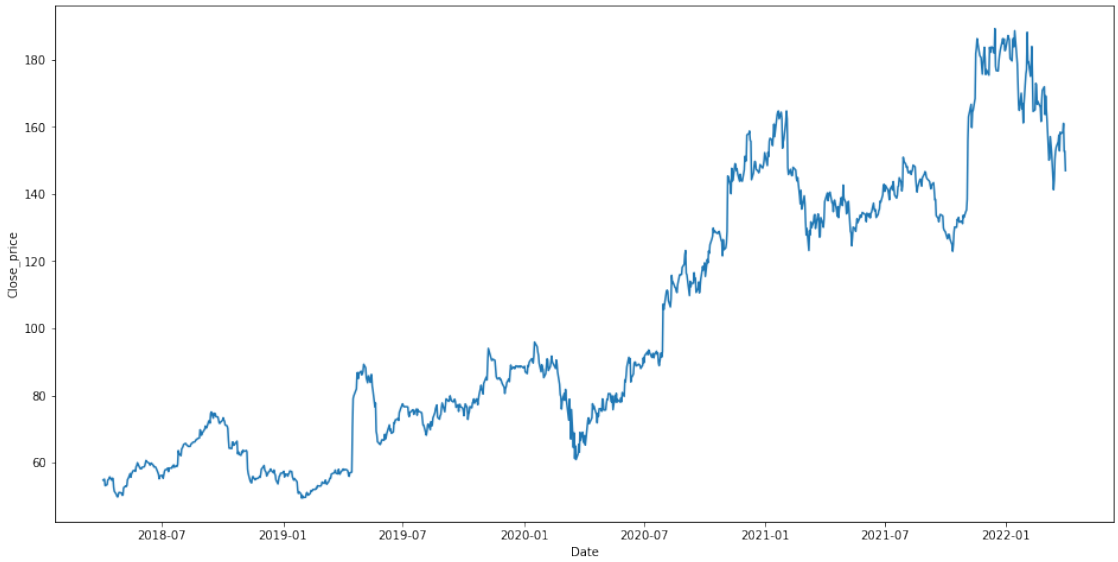
[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-04-03	54.290001	55.049999	53.619999	54.779999	48.340557	7930700
1	2018-04-04	53.799999	55.119999	53.419998	54.990002	48.525875	7496700
2	2018-04-05	55.490002	55.500000	54.439999	55.040001	48.569996	5823300
3	2018-04-06	54.419998	54.770000	53.110001	53.119999	46.875687	8322200
4	2018-04-09	53.520000	54.889999	53.340000	53.430000	47.149254	7608500

[5]:
```
df["Date"]=pd.to_datetime(df.Date,format="%Y-%m-%d")
df.index=df['Date']

plt.figure(figsize=(16,8))
plt.plot(df["Close"],label='Close Price history')
plt.xlabel('Date')
plt.ylabel('Close_price')
```

[5]:
```
Text(0, 0.5, 'Close_price')
```



0.3 Feature Extraction

The number of the trading days and the columns:

[6]:
```
df.shape
```

[6]: (1010, 7)

[7]:
```
df = df['Close'].values
df = df.reshape(-1, 1)
```

After extracting one column:

[8]:
```
df.shape
```

[8]: (1010, 1)

[9]:
```
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential, load_model
from keras.layers import LSTM, Dense, Dropout
```

[10]:
```
from sklearn import neighbors
from sklearn import linear_model
from sklearn import metrics
```

[11]:
```
from sklearn import model_selection
from sklearn import pipeline
from sklearn import preprocessing
```

[12]:
```
max_lags=81
normalizer=preprocessing.MinMaxScaler()
df=normalizer.fit_transform(df)
```

[13]:
```
TSS=model_selection.TimeSeriesSplit(n_splits=5)
Df = pd.DataFrame([])
r2_scores=np.zeros([max_lags-1])
for n_lag in range(20,max_lags,20):
   scores=[]
   x = []
   y = []
   for i in range(n_lag, df.shape[0]):
     x.append(df[i-n_lag:i, 0])
     y.append(df[i, 0])
   x = np.array(x)
   y = np.array(y)
   for train_ndx,valid_ndx in TSS.split(x,y):
     valid_ndx_adjusted=valid_ndx
     x_train=x[train_ndx]
     x_test=x[valid_ndx_adjusted]
     #x_train=normalizer.fit_transform(x_train)
     #x_test=normalizer.fit_transform(x_test)
     x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
     x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
     model = Sequential()
     model.add(LSTM(units=96, return_sequences=True, input_shape=(x_train.
 ↪shape[1], 1)))
     model.add(Dropout(0.2))
     model.add(LSTM(units=96, return_sequences=True))
     model.add(Dropout(0.2))
     model.add(LSTM(units=96, return_sequences=True))
     model.add(Dropout(0.2))
     model.add(LSTM(units=96))
     model.add(Dropout(0.2))
     model.add(Dense(units=1))
     model.compile(loss='mean_squared_error', optimizer='adam')
     model.fit(x_train,y[train_ndx],epochs=5, batch_size=32)
     score=(metrics.mean_squared_error\
          (y[valid_ndx_adjusted],model.predict(x_test)))
     scores.append(score)
   mse=np.mean(scores)
   Df=Df.append(pd.DataFrame({'lags' : n_lag,'mse' : mse}, index=[0]))
```

```
Epoch 1/5
6/6 [==============================] - 11s 68ms/step - loss: 0.0046
Epoch 2/5
6/6 [==============================] - 0s 68ms/step - loss: 0.0017
```

```
Epoch 3/5
6/6 [==============================] - 0s 72ms/step - loss: 0.0011
Epoch 4/5
6/6 [==============================] - 0s 66ms/step - loss: 9.9314e-04
Epoch 5/5
6/6 [==============================] - 0s 70ms/step - loss: 9.9990e-04
Epoch 1/5
11/11 [==============================] - 7s 70ms/step - loss: 0.0056
Epoch 2/5
11/11 [==============================] - 1s 72ms/step - loss: 0.0026
Epoch 3/5
11/11 [==============================] - 1s 70ms/step - loss: 0.0020
Epoch 4/5
11/11 [==============================] - 1s 69ms/step - loss: 0.0019
Epoch 5/5
11/11 [==============================] - 1s 70ms/step - loss: 0.0017
Epoch 1/5
16/16 [==============================] - 8s 70ms/step - loss: 0.0069
Epoch 2/5
16/16 [==============================] - 1s 72ms/step - loss: 0.0029
Epoch 3/5
16/16 [==============================] - 1s 71ms/step - loss: 0.0028
Epoch 4/5
16/16 [==============================] - 1s 71ms/step - loss: 0.0021
Epoch 5/5
16/16 [==============================] - 1s 72ms/step - loss: 0.0020
Epoch 1/5
21/21 [==============================] - 8s 71ms/step - loss: 0.0160
Epoch 2/5
21/21 [==============================] - 2s 72ms/step - loss: 0.0045
Epoch 3/5
21/21 [==============================] - 2s 72ms/step - loss: 0.0032
Epoch 4/5
21/21 [==============================] - 2s 72ms/step - loss: 0.0028
Epoch 5/5
21/21 [==============================] - 1s 71ms/step - loss: 0.0027
Epoch 1/5
26/26 [==============================] - 9s 73ms/step - loss: 0.0209
Epoch 2/5
26/26 [==============================] - 2s 72ms/step - loss: 0.0039
Epoch 3/5
26/26 [==============================] - 2s 72ms/step - loss: 0.0036
Epoch 4/5
26/26 [==============================] - 2s 73ms/step - loss: 0.0032
Epoch 5/5
26/26 [==============================] - 2s 72ms/step - loss: 0.0031
Epoch 1/5
6/6 [==============================] - 8s 130ms/step - loss: 0.0038
```

```
Epoch 2/5
6/6 [==============================] - 1s 129ms/step - loss: 0.0021
Epoch 3/5
6/6 [==============================] - 1s 127ms/step - loss: 0.0016
Epoch 4/5
6/6 [==============================] - 1s 247ms/step - loss: 0.0014
Epoch 5/5
6/6 [==============================] - 1s 127ms/step - loss: 0.0011
Epoch 1/5
11/11 [==============================] - 8s 134ms/step - loss: 0.0049
Epoch 2/5
11/11 [==============================] - 1s 134ms/step - loss: 0.0029
Epoch 3/5
11/11 [==============================] - 2s 136ms/step - loss: 0.0024
Epoch 4/5
11/11 [==============================] - 1s 132ms/step - loss: 0.0019
Epoch 5/5
11/11 [==============================] - 1s 135ms/step - loss: 0.0017
Epoch 1/5
16/16 [==============================] - 9s 136ms/step - loss: 0.0075
Epoch 2/5
16/16 [==============================] - 2s 136ms/step - loss: 0.0026
Epoch 3/5
16/16 [==============================] - 2s 136ms/step - loss: 0.0021
Epoch 4/5
16/16 [==============================] - 2s 136ms/step - loss: 0.0017
Epoch 5/5
16/16 [==============================] - 2s 137ms/step - loss: 0.0016
Epoch 1/5
21/21 [==============================] - 10s 135ms/step - loss: 0.0134
Epoch 2/5
21/21 [==============================] - 3s 135ms/step - loss: 0.0040
Epoch 3/5
21/21 [==============================] - 3s 135ms/step - loss: 0.0036
Epoch 4/5
21/21 [==============================] - 3s 136ms/step - loss: 0.0027
Epoch 5/5
21/21 [==============================] - 3s 135ms/step - loss: 0.0030
Epoch 1/5
26/26 [==============================] - 10s 137ms/step - loss: 0.0264
Epoch 2/5
26/26 [==============================] - 4s 138ms/step - loss: 0.0045
Epoch 3/5
26/26 [==============================] - 4s 137ms/step - loss: 0.0038
Epoch 4/5
26/26 [==============================] - 4s 137ms/step - loss: 0.0032
Epoch 5/5
26/26 [==============================] - 4s 138ms/step - loss: 0.0032
```

```
Epoch 1/5
5/5 [==============================] - 7s 185ms/step - loss: 0.0040
Epoch 2/5
5/5 [==============================] - 1s 185ms/step - loss: 0.0019
Epoch 3/5
5/5 [==============================] - 1s 301ms/step - loss: 0.0016
Epoch 4/5
5/5 [==============================] - 2s 316ms/step - loss: 0.0012
Epoch 5/5
5/5 [==============================] - 1s 185ms/step - loss: 8.2222e-04
Epoch 1/5
10/10 [==============================] - 9s 197ms/step - loss: 0.0066
Epoch 2/5
10/10 [==============================] - 2s 197ms/step - loss: 0.0034
Epoch 3/5
10/10 [==============================] - 2s 196ms/step - loss: 0.0027
Epoch 4/5
10/10 [==============================] - 2s 198ms/step - loss: 0.0022
Epoch 5/5
10/10 [==============================] - 2s 198ms/step - loss: 0.0018
Epoch 1/5
15/15 [==============================] - 10s 198ms/step - loss: 0.0093
Epoch 2/5
15/15 [==============================] - 3s 195ms/step - loss: 0.0035
Epoch 3/5
15/15 [==============================] - 3s 196ms/step - loss: 0.0027
Epoch 4/5
15/15 [==============================] - 3s 196ms/step - loss: 0.0022
Epoch 5/5
15/15 [==============================] - 3s 195ms/step - loss: 0.0021
Epoch 1/5
20/20 [==============================] - 11s 196ms/step - loss: 0.0137
Epoch 2/5
20/20 [==============================] - 4s 196ms/step - loss: 0.0053
Epoch 3/5
20/20 [==============================] - 4s 196ms/step - loss: 0.0037
Epoch 4/5
20/20 [==============================] - 4s 196ms/step - loss: 0.0027
Epoch 5/5
20/20 [==============================] - 4s 200ms/step - loss: 0.0029
Epoch 1/5
25/25 [==============================] - 12s 196ms/step - loss: 0.0228
Epoch 2/5
25/25 [==============================] - 5s 198ms/step - loss: 0.0044
Epoch 3/5
25/25 [==============================] - 5s 196ms/step - loss: 0.0034
Epoch 4/5
25/25 [==============================] - 5s 195ms/step - loss: 0.0036
```

```
Epoch 5/5
25/25 [==============================] - 5s 194ms/step - loss: 0.0031
Epoch 1/5
5/5 [==============================] - 9s 259ms/step - loss: 0.0042
Epoch 2/5
5/5 [==============================] - 1s 254ms/step - loss: 0.0022
Epoch 3/5
5/5 [==============================] - 1s 252ms/step - loss: 0.0022
Epoch 4/5
5/5 [==============================] - 1s 255ms/step - loss: 0.0015
Epoch 5/5
5/5 [==============================] - 1s 255ms/step - loss: 0.0010
Epoch 1/5
10/10 [==============================] - 10s 261ms/step - loss: 0.0064
Epoch 2/5
10/10 [==============================] - 3s 257ms/step - loss: 0.0032
Epoch 3/5
10/10 [==============================] - 3s 255ms/step - loss: 0.0027
Epoch 4/5
10/10 [==============================] - 3s 260ms/step - loss: 0.0023
Epoch 5/5
10/10 [==============================] - 3s 256ms/step - loss: 0.0019
Epoch 1/5
15/15 [==============================] - 12s 257ms/step - loss: 0.0092
Epoch 2/5
15/15 [==============================] - 4s 259ms/step - loss: 0.0032
Epoch 3/5
15/15 [==============================] - 4s 256ms/step - loss: 0.0023
Epoch 4/5
15/15 [==============================] - 4s 254ms/step - loss: 0.0022
Epoch 5/5
15/15 [==============================] - 4s 256ms/step - loss: 0.0019
Epoch 1/5
20/20 [==============================] - 13s 257ms/step - loss: 0.0167
Epoch 2/5
20/20 [==============================] - 5s 259ms/step - loss: 0.0037
Epoch 3/5
20/20 [==============================] - 5s 255ms/step - loss: 0.0039
Epoch 4/5
20/20 [==============================] - 5s 254ms/step - loss: 0.0038
Epoch 5/5
20/20 [==============================] - 5s 254ms/step - loss: 0.0031
Epoch 1/5
25/25 [==============================] - 14s 257ms/step - loss: 0.0178
Epoch 2/5
25/25 [==============================] - 6s 256ms/step - loss: 0.0048
Epoch 3/5
25/25 [==============================] - 6s 254ms/step - loss: 0.0045
```

```
Epoch 4/5
25/25 [==============================] - 6s 255ms/step - loss: 0.0036
Epoch 5/5
25/25 [==============================] - 6s 256ms/step - loss: 0.0031
```

[15]:
```
Df[Df['mse'] == np.amin(Df['mse'])]
```

[15]:

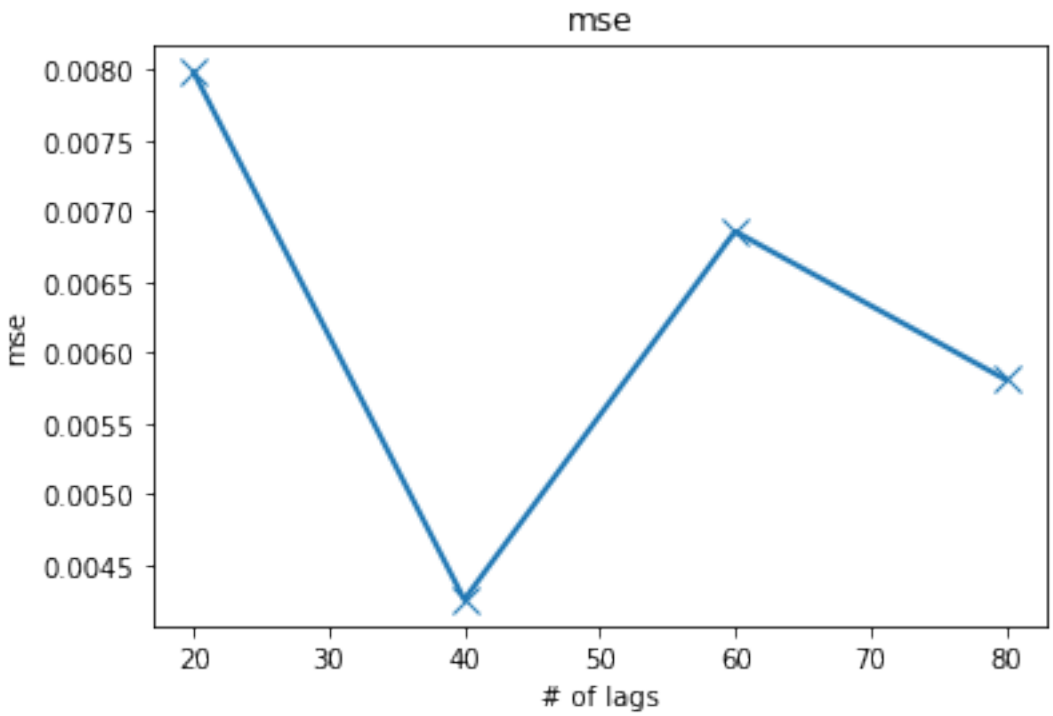
	lags	mse
0	40	0.004256

[72]:
```
Df
```

[72]:

	lags	mse
0	20	0.007981
0	40	0.004256
0	60	0.006856
0	80	0.005810

[73]:
```
fig=plt.figure()
plt.plot(Df['lags'],Df['mse'],'x-',lw=2,ms=10)
plt.xlabel('# of lags')
plt.ylabel('mse')
plt.title('mse')
plt.show()
```



[ ]:
```
model = load_model('stock_prediction.h5')
```

0.4 Results visualization

[65]:
```
def create_dataset(df):
    x = []
    y = []
    for i in range(40, df.shape[0]):
        x.append(df[i-40:i, 0])
        y.append(df[i, 0])
    x = np.array(x)
    y = np.array(y)
    return x,y
```

[66]:
```
dataset_train = np.array(df[:int(df.shape[0]*0.8)])
dataset_test = np.array(df[int(df.shape[0]*0.8):])
```

[67]:
```
x_train, y_train=create_dataset(dataset_train)
x_test, y_test=create_dataset(dataset_test)
```

[68]:
```
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

[69]:
```
    model = Sequential()
    model.add(LSTM(units=96, return_sequences=True, input_shape=(x_train.
↪shape[1], 1)))
    model.add(Dropout(0.2))
    model.add(LSTM(units=96, return_sequences=True))
    model.add(Dropout(0.2))
    model.add(LSTM(units=96, return_sequences=True))
    model.add(Dropout(0.2))
    model.add(LSTM(units=96))
    model.add(Dropout(0.2))
    model.add(Dense(units=1))
    model.compile(loss='mean_squared_error', optimizer='adam')
```

[70]:
```
model.fit(x_train, y_train, epochs=50, batch_size=32)
model.save('stock_prediction.h5')
```

```
Epoch 1/50
24/24 [==============================] - 9s 126ms/step - loss: 0.0242
Epoch 2/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0048
Epoch 3/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0036
Epoch 4/50
```

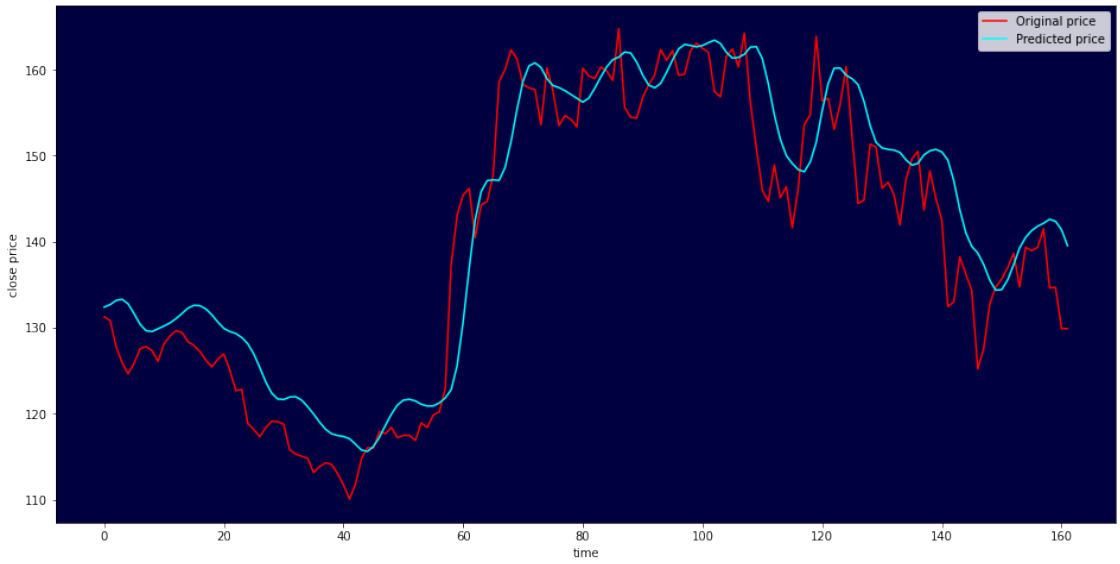
```
24/24 [==============================] - 3s 128ms/step - loss: 0.0032
Epoch 5/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0034
Epoch 6/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0033
Epoch 7/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0027
Epoch 8/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0027
Epoch 9/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0026
Epoch 10/50
24/24 [==============================] - 3s 124ms/step - loss: 0.0027
Epoch 11/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0024
Epoch 12/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0030
Epoch 13/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0025
Epoch 14/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0023
Epoch 15/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0021
Epoch 16/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0022
Epoch 17/50
24/24 [==============================] - 3s 124ms/step - loss: 0.0022
Epoch 18/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0021
Epoch 19/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0021
Epoch 20/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0022
Epoch 21/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0019
Epoch 22/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0019
Epoch 23/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0019
Epoch 24/50
24/24 [==============================] - 3s 128ms/step - loss: 0.0018
Epoch 25/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0018
Epoch 26/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0021
Epoch 27/50
24/24 [==============================] - 3s 124ms/step - loss: 0.0018
Epoch 28/50
```

```
24/24 [==============================] - 3s 126ms/step - loss: 0.0018
Epoch 29/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0017
Epoch 30/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0015
Epoch 31/50
24/24 [==============================] - 4s 161ms/step - loss: 0.0018
Epoch 32/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0018
Epoch 33/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0015
Epoch 34/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0018
Epoch 35/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0015
Epoch 36/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0014
Epoch 37/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0017
Epoch 38/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0015
Epoch 39/50
24/24 [==============================] - 3s 128ms/step - loss: 0.0015
Epoch 40/50
24/24 [==============================] - 3s 127ms/step - loss: 0.0016
Epoch 41/50
24/24 [==============================] - 3s 128ms/step - loss: 0.0017
Epoch 42/50
24/24 [==============================] - 3s 128ms/step - loss: 0.0014
Epoch 43/50
24/24 [==============================] - 3s 128ms/step - loss: 0.0016
Epoch 44/50
24/24 [==============================] - 4s 147ms/step - loss: 0.0015
Epoch 45/50
24/24 [==============================] - 4s 168ms/step - loss: 0.0013
Epoch 46/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0014
Epoch 47/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0015
Epoch 48/50
24/24 [==============================] - 3s 126ms/step - loss: 0.0013
Epoch 49/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0012
Epoch 50/50
24/24 [==============================] - 3s 125ms/step - loss: 0.0011
```

[71]:
```
predictions = model.predict(x_test)
prediction = normalizer.inverse_transform(predictions)
y_test_scaled = normalizer.inverse_transform(y_test.reshape(-1, 1))

fig, ax = plt.subplots(figsize=(16,8))
ax.set_facecolor('#000041')
ax.plot(y_test_scaled, color='red', label='Original price')
plt.plot(prediction, color='cyan', label='Predicted price')
plt.xlabel('time')
plt.ylabel('close price')
plt.legend()
```

[71]:
```
<matplotlib.legend.Legend at 0x7f9041621f10>
```



[ ]:
```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('Copy of stock price predection.ipynb')
```

```
File ‘colab_pdf.py’ already there; not retrieving.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```