stockive

February 6, 2019

```
In [67]: import numpy as np
         from matplotlib import pyplot as plt
         import pandas as pd
         from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import LSTM
         from keras.layers import Dropout
```

0.1 Create Dataset

```
In [68]: stock_dataset = pd.read_csv("AMD.csv", index_col = "Date", parse_dates=True)

In [69]: stock_dataset.head()
```

Out[69]:

Date	Open	High	Low	Close	Adj Close	Volume
2009-05-22	4.43	4.43	4.25	4.26	4.26	8274300
2009-05-26	4.26	4.57	4.23	4.53	4.53	16094300
2009-05-27	4.57	4.80	4.55	4.71	4.71	21512600
2009-05-28	4.75	4.84	4.54	4.70	4.70	18383900
2009-05-29	4.71	4.78	4.38	4.54	4.54	24539700

```
In [70]: stock_dataset.describe()
```

Out[70]:

	Open	High	Low	Close	Adj Close	Volume
count	2335.000000	2335.000000	2335.000000	2335.000000	2335.000000	2.335000e+03
mean	6.434732	6.563961	6.300887	6.433221	6.433221	3.107491e+07
std	3.872391	3.954999	3.792914	3.879124	3.879124	2.791149e+07
min	1.620000	1.690000	1.610000	1.620000	1.620000	
25%	3.480000	3.550000	3.405000	3.465000	3.465000	
50%	5.580000	5.720000	5.450000	5.600000	5.600000	
75%	8.555000	8.710000	8.375000	8.565000	8.565000	
max	25.510000	27.299999	24.629999	25.260000	25.260000	

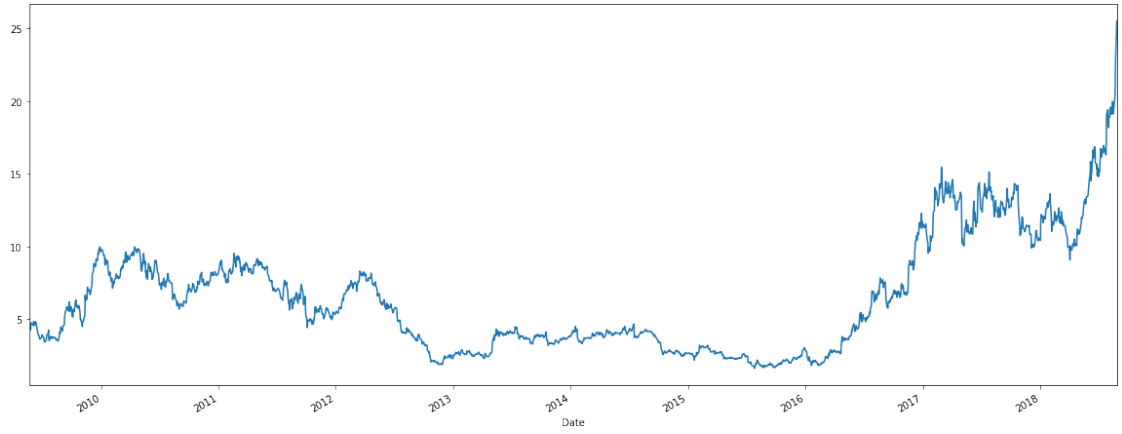
```
min     0.000000e+00
25%     1.421670e+07
50%     2.228020e+07
75%     3.776905e+07
max     3.250584e+08
```

```
In [71]: stock_dataset.dtypes
```

```
Out[71]: Open          float64
         High          float64
         Low           float64
         Close         float64
         Adj Close     float64
         Volume          int64
         dtype: object
```

```
In [72]: stock_dataset['Open'].plot(figsize = (20,8))
```

```
Out[72]: <matplotlib.axes._subplots.AxesSubplot at 0x13c388b70>
```




```
In [22]: open_dataset = stock_dataset["Open"]
         mean = open_dataset.mean()
         std = open_dataset.std()
         mean,std
```

```
Out[22]: (6.4347323361884365, 3.872390825060707)
```

```
In [73]: open_dataset = open_dataset.apply(lambda x : (x - mean) / std)
         open_dataset.head()
```

```
Out[73]: Date
         2009-05-22   -1.795385
         2009-05-26   -1.806721
```

```
2009-05-27   -1.786049
2009-05-28   -1.774045
2009-05-29   -1.776712
Name: Open, dtype: float64
```

```
In [74]: open_dataset.shape
```

```
Out[74]: (2335,)
```

```
In [75]: features = []
         y = []

         window_size = 50
         predicted_size = 1

         for i in range(open_dataset.shape[0] - window_size - predicted_size):
             features.append(open_dataset[i:i+window_size])
             y.append(open_dataset[i+ window_size])
         features, y = np.array(features), np.array(y)
         features = np.reshape(features, (features.shape[0], features.shape[1], 1))
```

0.2 Train test split (20%)

```
In [76]: split_index = int(features.shape[0]*(1-0.2))
         X_train, X_test = features[0:split_index],features[split_index:]
         y_train,y_test = y[0:split_index],y[split_index:]

         X_train.shape, y_train.shape
```

```
Out[76]: ((1827, 50, 1), (1827,))
```

0.3 Create Neural Network

```
In [78]: model = Sequential()

         model.add(LSTM(units = 50, return_sequences = True, input_shape = (window_size, 1)))
         model.add(Dropout(0.2))

         model.add(LSTM(units = 100, return_sequences = True))
         model.add(Dropout(0.2))

         model.add(LSTM(units = 100, return_sequences = True))
         model.add(Dropout(0.2))

         model.add(LSTM(units = 50), return_sequences = False)
         model.add(Dropout(0.2))

         model.add(Dense(units = 1))

         model.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

0.4 Fit model

```
In [85]: model.fit(
             features,
             y,
             epochs = 50,
             batch_size = 32
         )
```

```
Epoch 1/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0221
Epoch 2/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0201
Epoch 3/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0216
Epoch 4/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0197
Epoch 5/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0180
Epoch 6/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0178
Epoch 7/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0187
Epoch 8/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0186
Epoch 9/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0182
Epoch 10/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0177
Epoch 11/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0167
Epoch 12/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0160
Epoch 13/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0176
Epoch 14/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0157
Epoch 15/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0144
Epoch 16/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0164
Epoch 17/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0157
Epoch 18/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0172
Epoch 19/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0152
Epoch 20/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0173
```

```
Epoch 21/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0151
Epoch 22/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0135
Epoch 23/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0153
Epoch 24/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0155
Epoch 25/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0148
Epoch 26/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0139
Epoch 27/50
2284/2284 [==============================] - 18s 8ms/step - loss: 0.0138
Epoch 28/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0137
Epoch 29/50
2284/2284 [==============================] - 16s 7ms/step - loss: 0.0132
Epoch 30/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0131
Epoch 31/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0130
Epoch 32/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0143
Epoch 33/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0124
Epoch 34/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0122
Epoch 35/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0123
Epoch 36/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0119
Epoch 37/50
2284/2284 [==============================] - 17s 7ms/step - loss: 0.0121
Epoch 38/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0115
Epoch 39/50
2284/2284 [==============================] - 16s 7ms/step - loss: 0.0118
Epoch 40/50
2284/2284 [==============================] - 16s 7ms/step - loss: 0.0116
Epoch 41/50
2284/2284 [==============================] - 15s 7ms/step - loss: 0.0106
Epoch 42/50
2284/2284 [==============================] - 15s 6ms/step - loss: 0.0113
Epoch 43/50
2284/2284 [==============================] - 15s 7ms/step - loss: 0.0125
Epoch 44/50
2284/2284 [==============================] - 16s 7ms/step - loss: 0.0099
```

```
Epoch 45/50
2284/2284 [==============================] - 15s 6ms/step - loss: 0.0107
Epoch 46/50
2284/2284 [==============================] - 15s 7ms/step - loss: 0.0104
Epoch 47/50
2284/2284 [==============================] - 16s 7ms/step - loss: 0.0104
Epoch 48/50
2284/2284 [==============================] - 15s 6ms/step - loss: 0.0098
Epoch 49/50
2284/2284 [==============================] - 17s 8ms/step - loss: 0.0099
Epoch 50/50
2284/2284 [==============================] - 19s 8ms/step - loss: 0.0097
```

```
Out[85]: <keras.callbacks.History at 0x13f06f160>
```

```
In [86]: predict_stcok = model.predict(X_test)
```

```
In [89]: plt.figure(figsize = (20,4))
         plt.plot(predict_stcok,color='red', label='prediction')
         plt.plot(y_test,color='blue', label='y_test')
         plt.legend(loc='upper left')
         plt.show()
```