Facebook Stock👀

Import modules

[1]
```
import quandl as ql
import plotly as py
import plotly.graph_objs as go
import pandas as pd
# import numpy as np
from keras.models import Sequential
from keras.layers import LSTM,Dense
from sklearn.preprocessing import MinMaxScaler
#import matplotlib.pyplot as plot
#import tensorflow as tf
scaler = MinMaxScaler()
```

```
D:\Tools\Anaconda3\lib\site-packages\h5py\__init__.py:36: FutureWarning:

Conversion of the second argument of issubdtype from `float` to
`np.floating` is deprecated. In future, it will be treated as `np.float64
== np.dtype(float).type`.

Using TensorFlow backend.
```

Plotly and Quandl config

[2]
```
py.offline.init_notebook_mode(connected=True)
ql.ApiConfig.api_key = "b6y7-mew_t8z5yGJijFv"
```

Retriving, Slicing and Dicing

Use to fetch data from quandl

[3]
```
# data = ql.get("WIKI/fb")
# data.to_csv("Facebook", encoding='utf-8')
```

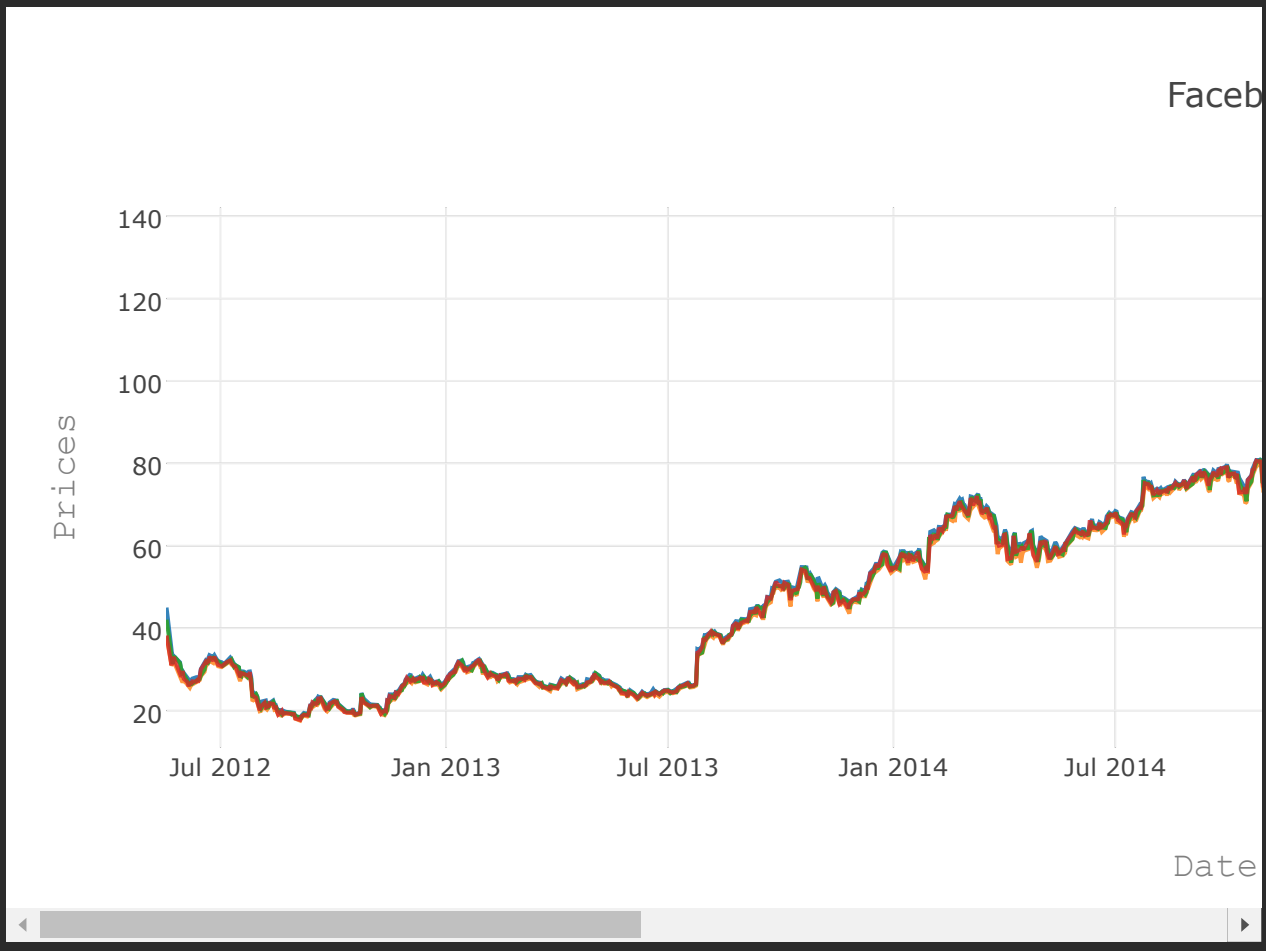
[4]
```
df = pd.read_csv("Facebook")[['Date','Open','Close','High','Low']]
train_limit = int(df.shape[0]*0.8)
train_data = df[:train_limit]
df[:5]
```

	Date	Open	Close	High	Low
0	2012-05-18	42.05	38.2318	45.00	38.00
1	2012-05-21	36.53	34.0300	36.66	33.00
2	2012-05-22	32.61	31.0000	33.59	30.94
3	2012-05-23	31.37	32.0000	32.50	31.36
4	2012-05-24	32.95	33.0300	33.21	31.77

Plotting as a TimeSeries

[5]
```
trace_high = go.Scatter(
    x=train_data['Date'],
    y=train_data['High'],
    name = "High",
#     line = dict(color = '#17BECF'),
    opacity = 0.9)

trace_low = go.Scatter(
    x=train_data['Date'],
    y=train_data['Low'],
    name = "Low",
#     line = dict(color = '#7F7F7F'),
    opacity = 0.8)

trace_open = go.Scatter(
    x=train_data['Date'],
    y=train_data['Open'],
    name = "Open",
#     line = dict(color = '#17BECF'),
    opacity = 0.9)

trace_close = go.Scatter(
    x=train_data['Date'],
    y=train_data['Close'],
    name = "Close",
#     line = dict(color = '#7F7F7F'),
    opacity = 0.8)

plot_data = [ trace_high, trace_low, trace_open, trace_close]
```

```
layout = go.Layout(
    title='Facebook',
    xaxis=dict(
        title='Date',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    ),
    yaxis=dict(
        title='Prices',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

fig = dict(data=plot_data, layout=layout)
py.offline.iplot(fig, filename = "Facebook Plot")
```



The fun part aka some lstm magic!!

Scaling

Warning! DataFrames will be changed after this...

[6]
```
df[['Open','Close','High','Low']] = scaler.fit_transform(df[['Open','Clos
df = df.drop( ['Date'], axis = 1)
train_data = df[:train_limit]
test_data = df[train_limit:]
train_data[:5]
# df[['Open','Close','High','Low']] = scaler.inverse_transform(df[['Open'
# df
```

	Open	Close	High	Low
0	0.137790	0.116918	0.150974	0.118599
1	0.106059	0.092957	0.103869	0.089602
2	0.083525	0.075678	0.086529	0.077655
3	0.076397	0.081381	0.080373	0.080090
4	0.085479	0.087254	0.084383	0.082468

[7]