overbought-oversold

September 29, 2021

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import matplotlib.ticker as mticker
import matplotlib
from mpl_finance import candlestick_ohlc
from datetime import datetime
import seaborn as sns
sns.set()
```

[2]:
```
df = pd.read_csv('TSLA.csv')
df.head()
```

[2]:

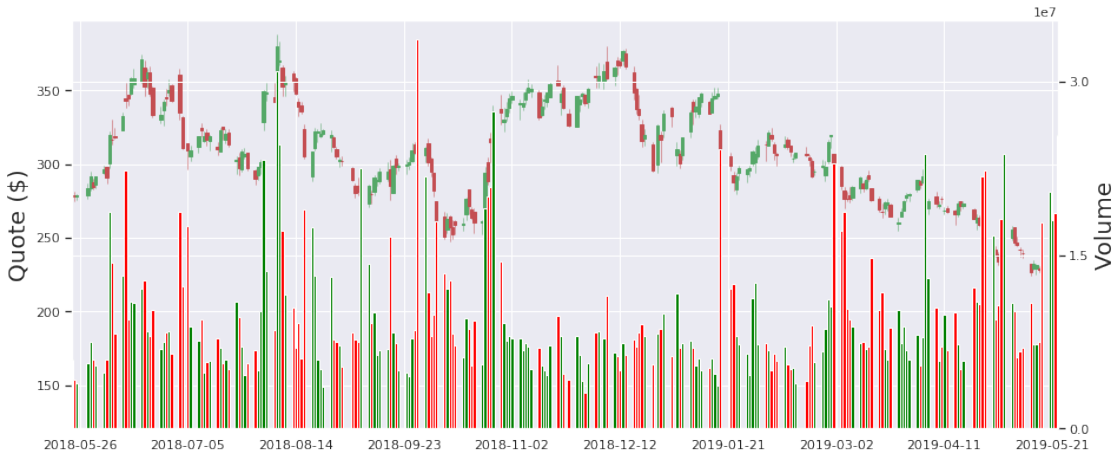
	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-05-23	277.760010	279.910004	274.000000	279.070007	279.070007	5953100
1	2018-05-24	278.399994	281.109985	274.890015	277.850006	277.850006	4176700
2	2018-05-25	277.630005	279.640015	275.609985	278.850006	278.850006	3875100
3	2018-05-29	278.510010	286.500000	276.149994	283.760010	283.760010	5666600
4	2018-05-30	283.290009	295.010010	281.600006	291.720001	291.720001	7489700

[3]:
```
date = [datetime.strptime(d, '%Y-%m-%d') for d in df['Date']]
```

[4]:
```
candlesticks = list(zip(mdates.date2num(date),df['Open'],
                  df['High'],df['Low'],df['Close'],df['Volume']))
```

[5]:
```
fig = plt.figure(figsize = (15, 15))
ax = fig.add_subplot(1,1,1)
ax.set_ylabel('Quote ($)', size=20)
```

```
dates = [x[0] for x in candlesticks]
dates = np.asarray(dates)
volume = [x[5] for x in candlesticks]
volume = np.asarray(volume)

candlestick_ohlc(ax, candlesticks, width=1,
                 colorup='g', colordown='r')
pad = 0.25
yl = ax.get_ylim()
ax.set_ylim(yl[0]-(yl[1]-yl[0])*pad,yl[1])
ax2 = ax.twinx()

ax2.set_position(matplotlib.transforms.Bbox([[0.125,0],[0.9,0.32]]))

pos = df['Open'] - df['Close']<0
neg = df['Open'] - df['Close']>0
ax2.bar(dates[pos],volume[pos],color='green',width=1,align='center')
ax2.bar(dates[neg],volume[neg],color='red',width=1,align='center')

ax2.set_xlim(min(dates),max(dates))
yticks = ax2.get_yticks()
ax2.set_yticks(yticks[::3])

ax2.yaxis.set_label_position("right")
ax2.set_ylabel('Volume', size=20)

ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
ax.xaxis.set_major_locator(mticker.MaxNLocator(10))

plt.show()
```



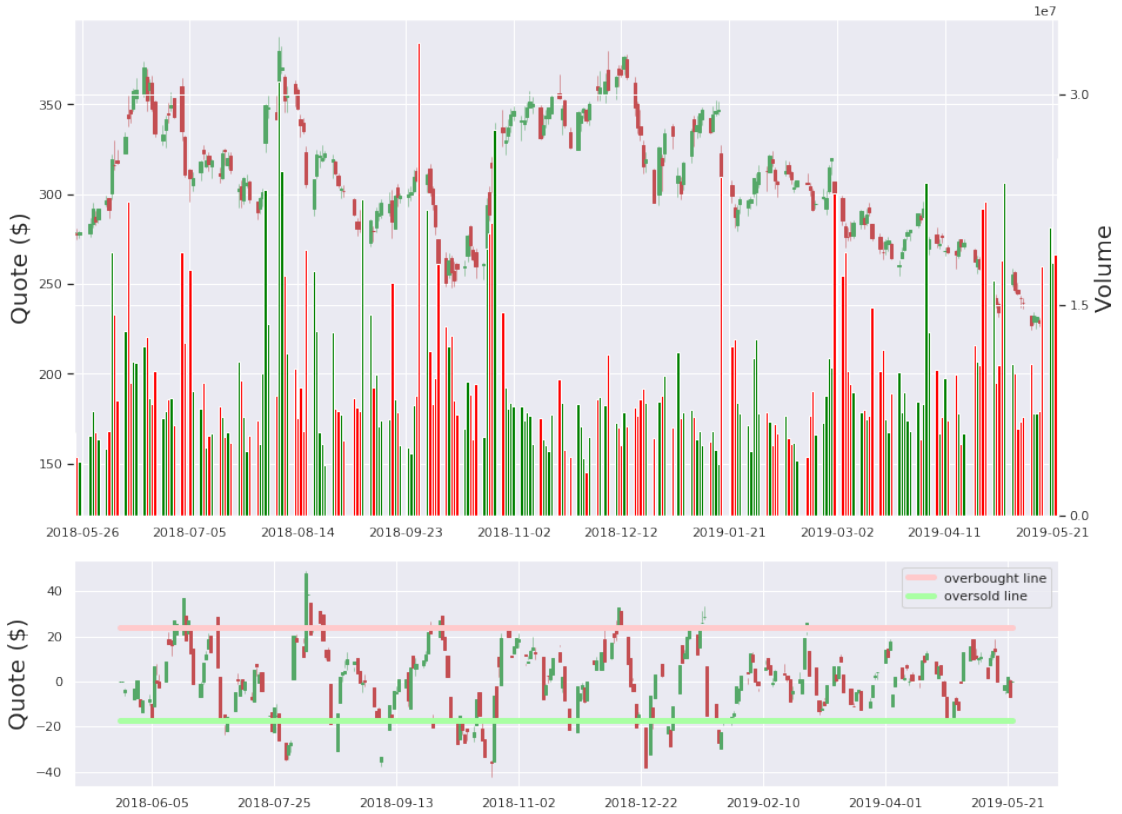
[6]:
```
def removal(signal, repeat):
    copy_signal = np.copy(signal)
    for j in range(repeat):
        for i in range(3, len(signal)):
            copy_signal[i - 1] = (copy_signal[i - 2] + copy_signal[i]) / 2
    return copy_signal

def get(original_signal, removed_signal):
    buffer = []
    for i in range(len(removed_signal)):
        buffer.append(original_signal[i] - removed_signal[i])
    return np.array(buffer)

signal = np.copy(df.Open.values)
removed_signal = removal(signal, 30)
noise_open = get(signal, removed_signal)

signal = np.copy(df.High.values)
removed_signal = removal(signal, 30)
noise_high = get(signal, removed_signal)

signal = np.copy(df.Low.values)
removed_signal = removal(signal, 30)
noise_low = get(signal, removed_signal)

signal = np.copy(df.Close.values)
removed_signal = removal(signal, 30)
noise_close = get(signal, removed_signal)
```

[7]:
```
noise_candlesticks = list(zip(mdates.date2num(date),noise_open,
                              noise_high,noise_low,noise_close))

fig = plt.figure(figsize = (15, 5))
ax = fig.add_subplot(1,1,1)
ax.set_ylabel('Quote ($)', size=20)

candlestick_ohlc(ax, noise_candlesticks, width=1,
                 colorup='g', colordown='r')
ax.plot(dates, [np.percentile(noise_close, 95)] * len(noise_candlesticks),␣
↪color = (1.0, 0.792156862745098, 0.8, 0.7),
       linewidth=10.0, label = 'overbought line')

ax.plot(dates, [np.percentile(noise_close, 10)] * len(noise_candlesticks),
        color = (0.6627450980392157, 1.0, 0.6392156862745098, 0.7),
       linewidth=10.0, label = 'oversold line')

ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
```

```
ax.xaxis.set_major_locator(mticker.MaxNLocator(10))

plt.legend()
plt.show()
```

[8]:
```
fig = plt.figure(figsize = (15, 12))
ax1 = plt.subplot2grid((3, 1), (0, 0), rowspan=2)

ax1.set_ylabel('Quote ($)', size=20)

dates = [x[0] for x in candlesticks]
dates = np.asarray(dates)
volume = [x[5] for x in candlesticks]
volume = np.asarray(volume)

candlestick_ohlc(ax1, candlesticks, width=1,
                 colorup='g', colordown='r')
pad = 0.25
yl = ax1.get_ylim()
ax1.set_ylim(yl[0]-(yl[1]-yl[0])*pad,yl[1])
ax2 = ax1.twinx()

pos = df['Open'] - df['Close']<0
neg = df['Open'] - df['Close']>0
ax2.bar(dates[pos],volume[pos],color='green',width=1,align='center')
ax2.bar(dates[neg],volume[neg],color='red',width=1,align='center')

ax2.set_xlim(min(dates),max(dates))
yticks = ax2.get_yticks()
ax2.set_yticks(yticks[::3])

ax2.yaxis.set_label_position("right")
ax2.set_ylabel('Volume', size=20)
```

```
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
ax1.xaxis.set_major_locator(mticker.MaxNLocator(10))

ax2 = plt.subplot2grid((3, 1), (2, 0))

ax2.set_ylabel('Quote ($)', size=20)

candlestick_ohlc(ax2, noise_candlesticks, width=1,
                 colorup='g', colordown='r')
ax2.plot(dates, [np.percentile(noise_close, 95)] * len(noise_candlesticks),␣
↪color = (1.0, 0.792156862745098, 0.8, 1.0),
       linewidth=5.0, label = 'overbought line')

ax2.plot(dates, [np.percentile(noise_close, 10)] * len(noise_candlesticks),
        color = (0.6627450980392157, 1.0, 0.6392156862745098, 1.0),
       linewidth=5.0, label = 'oversold line')

ax2.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
ax2.xaxis.set_major_locator(mticker.MaxNLocator(10))

plt.legend()
plt.show()
```



[ ]: