Learning-Python-AlpaVantage-ML

December 11, 2020

0.1 Stock Analysis and ML

0.1.1 Stock Analysis

[3]:
```
import pandas as pd
import yfinance as yf
```

[4]:
```
tickerSymbol = 'HD'
```

[5]:
```
hd_df = yf.Ticker(tickerSymbol)
```

[6]:
```
dt_range = hd_df.history(period='1d', start='2020-12-3', end='2020-12-4')
```

[7]:
```
hd_df.recommendations
```

[7]:
```
                                Firm    To Grade  From Grade Action
Date

2012-02-22 09:06:00      Jefferies      Hold            main
2012-02-22 11:12:00            UBS       Buy            main
2012-03-20 06:30:00  Deutsche Bank      Hold            main
2012-03-21 06:10:00  Credit Suisse Outperform           main
2012-03-26 08:48:00 Canaccord Genuity   Hold            init
…                                …         …         …    …
2020-08-19 13:58:34 B of A Securities    Buy    Neutral   up
2020-09-18 15:11:12    Oppenheimer   Perform Outperform down
2020-10-07 11:33:40 Morgan Stanley Overweight           main
2020-11-12 12:30:33 Gordon Haskett       Buy Accumulate   up
2020-12-04 14:55:12 Morgan Stanley Overweight           main
[217 rows x 4 columns]
```

[6]:
```
# quandl api key = Quandl_API_KEY
# alpha_vantage key = ALPHA_API_KEY
```

[8]:
```
import pandas as pd
from alpha_vantage.timeseries import TimeSeries
import time
```

[9]:
```
api_key = 'Y6OSP0UFVK3CU8DP'
```

[10]:
```
ts = TimeSeries(key=api_key, output_format='json')
```

[11]:
```
data, meta_data = ts.get_intraday(symbol='HD', interval = '1min', outputsize =
↪'full')
```

[12]:
```
print(data)
```

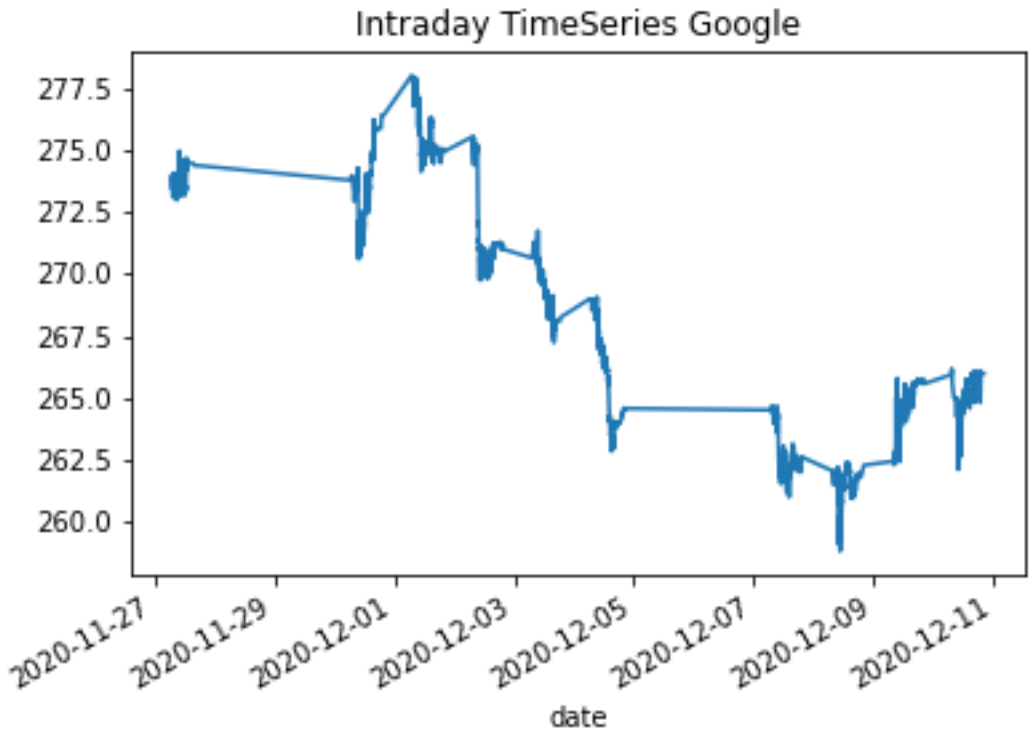
```
{'2020-12-10 20:00:00': {'1. open': '266.0000', '2. high':
'266.0000', '3. low': '266.0000', '4. close': '266.0000', '5.
volume': '204'}, '2020-12-10 19:25:00':
{'1. open': '266.0000', '2. high': '266.0000', '3. low': '266.0000',
'4. close': '266.0000', '5. volume': '200'}, '2020-12-10 18:50:00':
{'1. open': '265.9000', '2. high': '265.9000', '3. low': '265.9000',
'4. close': '265.9000', '5. volume': '183'}, '2020-12-10 18:35:00':
{'1. open': '264.8100', '2. high': '264.8100', '3. low': '264.8100',
'4. close': '264.8100', '5. volume': '13955'}, '2020-12-10 18:26:00':
{'1. open': '266.0900', '2. high': '266.1000', '3. low': '266.0900',
'4. close': '266.1000', '5. volume': '903'}, '2020-12-10 18:00:00':
{'1. open': '265.9400', '2. high': '265.9400', '3. low': '265.9400',
'4. close': '265.9400', '5. volume': '202'}, '2020-12-10 17:16:00':
{'1. open': '265.9700', '2. high': '265.9700', '3. low': '265.9700',

 . . . Truncated . . .

{'1. open': '273.4768', '2. high': '273.4868', '3. low': '273.4768',
'4. close': '273.4868', '5. volume': '492'}, '2020-11-27 08:32:00':
{'1. open': '273.4867', '2. high': '273.4867', '3. low': '273.4867',
'4. close': '273.4867', '5. volume': '520'}, '2020-11-27 08:27:00':
{'1. open': '273.3873', '2. high':
'273.3873', '3. low': '273.3873', '4. close': '273.3873', '5.
volume': '510'},
'2020-11-27 08:01:00': {'1. open': '273.4768', '2. high': '274.0735',
'3. low':
'273.4768', '4. close': '274.0735', '5. volume': '2670'}, '2020-11-27
07:29:00': {'1. open': '273.2481', '2. high': '273.2481', '3. low':
'273.0890', '4. close': '273.0890', '5. volume': '806'}, '2020-11-27
07:05:00': {'1. open': '273.4768', '2. high': '273.9840', '3. low':
'273.4768', '4. close': '273.9840', '5. volume': '520'}, '2020-11-27
07:01:00': {'1. open': '273.4868', '2. high': '273.4868', '3. low':
'273.4868', '4. close': '273.4868', '5. volume': '634'}}
```

[13]:
```
import pandas as pd
from alpha_vantage.timeseries import TimeSeries
import time
import matplotlib.pyplot as plt
```

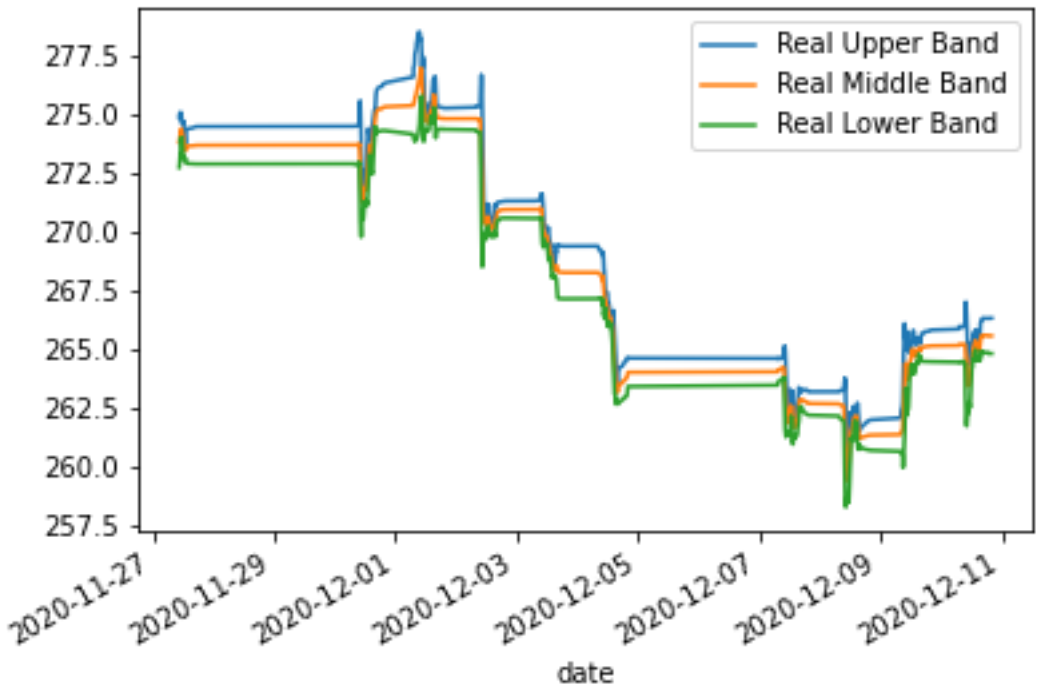
[14]:
```
from alpha_vantage.timeseries import TimeSeries
# Please Contact for Professional Services
print(data)
```

```
                     1. open   2. high    3. low  4. close  5. volume
date
2020-12-10 20:00:00 266.0000 266.0000 266.0000          204.0
266.0000
2020-12-10 19:25:00 266.0000 266.0000 266.0000          200.0
266.0000
2020-12-10 18:50:00 265.9000 265.9000 265.9000          183.0
265.9000
2020-12-10 18:35:00 264.8100 264.8100 264.8100        13955.0
264.8100
2020-12-10 18:26:00 266.0900 266.1000 266.0900          903.0
266.1000
…                        …       …       …       …        …
2020-11-27 08:27:00 273.3873 273.3873 273.3873          510.0
273.3873
2020-11-27 08:01:00 273.4768 274.0735 273.4768         2670.0
274.0735
2020-11-27 07:29:00 273.2481 273.2481 273.0890          806.0
273.0890
2020-11-27 07:05:00 273.4768 273.9840 273.4768          520.0
273.9840
2020-11-27 07:01:00 273.4868 273.4868 273.4868          634.0
273.4868
[4129 rows x 5 columns]
```

[15]:
```
data['4. close'].plot()
plt.title('Intraday TimeSeries Google')
plt.show()
```



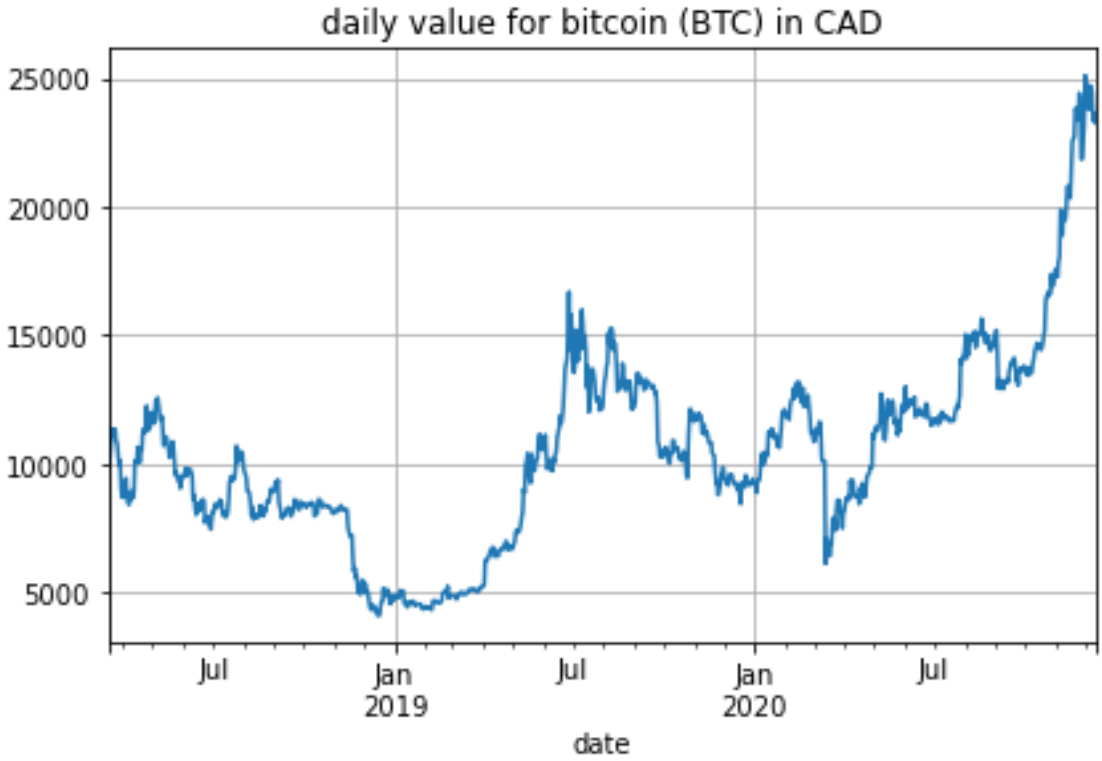
[17]:
```
from alpha_vantage.techindicators import TechIndicators
# Please Contact for Professional Services
plt.show()
```



[18]:
```
from alpha_vantage.cryptocurrencies import CryptoCurrencies
# Please Contact for Professional
Services plt.show()
```

```
      1a. open (CAD) 1b. open (USD) 2a. high (CAD) 2b. high (USD) \
date
2020-12-11  23267.763374      18254.81  23316.096585      18292.73
2020-12-10  23632.913647      18541.29  23653.345645      18557.32
2020-12-09  23356.093847      18324.11  23758.182318      18639.57
2020-12-08  24430.322409      19166.90  24593.396012      19294.84
2020-12-07  24674.754369      19358.67  24754.086095      19420.91
...                  ...           ...           ...           ...
2018-03-22  11324.680420       8884.82  11598.951000       9100.00
2018-03-21  11356.724116       8909.96  11697.108716       9177.01
2018-03-20  10955.285696       8595.01  11535.220500       9050.00
2018-03-19  10437.781290       8189.00  11095.773210       8705.23
2018-03-18   9972.561386       7824.01  10601.441214       8317.40

      3a. low (CAD) 3b. low (USD) 4a. close (CAD) 4b. close (USD) \
date
2020-12-11 22879.249500     17950.00   22950.602168      18005.98
2020-12-10 22829.692663     17911.12   23267.533944      18254.63
2020-12-09 22496.866500     17650.00   23632.900901      18541.28
2020-12-08 23197.902000     18200.00   23356.093847      18324.11
```

```
 2020-12-07 24093.799877    18902.88    24430.322409      19166.90
 …                   …           …           …              …
2018-03-22 10789.701111      8465.10   11118.295569        8722.90
2018-03-21 11153.602266      8750.60   11324.909850        8885.00
2018-03-20 10553.770800      8280.00   11356.749608        8909.98
2018-03-19 10309.555524      8088.40   10961.646000        8600.00
2018-03-18  9332.694420      7322.00   10439.043154        8189.99

              5. volume 6. market cap (USD)
 date
 2020-12-11 3892.503080 3892.503080 2020-12-10
 52890.675094 52890.675094
2020-12-09 79585.553801      79585.553801
2020-12-08 61626.947614      61626.947614
2020-12-07 41372.296293      41372.296293
 …                   …                 …
2018-03-22 40617.556809      40617.556809
2018-03-21 39972.405371      39972.405371
2018-03-20 44865.105835      44865.105835
2018-03-19 55297.084942      55297.084942
2018-03-18 59488.231711      59488.231711

 [1000 rows x 10 columns]
```



[19]:
```
from alpha_vantage.timeseries import TimeSeries
# Please Contact for Professional Services
print(data)
```

```
                     1. open  2. high  3. low 4. close 5. volume
date
2020-12-10 20:00:00 266.0000 266.0000 266.0000          204.0
266.0000
2020-12-10 19:25:00 266.0000 266.0000 266.0000          200.0
266.0000
2020-12-10 18:50:00 265.9000 265.9000 265.9000          183.0
265.9000
2020-12-10 18:35:00 264.8100 264.8100 264.8100        13955.0
264.8100
2020-12-10 18:26:00 266.0900 266.1000 266.0900          903.0
266.1000
…                         …       …       …       …        …
2020-11-27 08:27:00 273.3873 273.3873 273.3873          510.0
273.3873
2020-11-27 08:01:00 273.4768 274.0735 273.4768         2670.0
274.0735
2020-11-27 07:29:00 273.2481 273.2481 273.0890          806.0
273.0890
2020-11-27 07:05:00 273.4768 273.9840 273.4768          520.0
273.9840
2020-11-27 07:01:00 273.4868 273.4868 273.4868          634.0
273.4868
[4129 rows x 5 columns]
```

[20]:
```
# Please Contact for Professional Services
print(percentage_change)
```

```
date
2020-12-10              NaN
20:00:00
2020-12-10         0.000000
19:25:00
2020-12-10        -
18:50:00          0.000376
2020-12-10        -
18:35:00          0.004099
2020-12-10         0.004871
18:26:00             …
2020-11-27        -
08:27:00          0.000363
2020-11-27         0.002510
08:01:00
```

```
2020-11-27          -
07:29:00            0.003592
2020-11-27           0.003277
07:05:00
2020-11-27          -
07:01:00            0.001815
Name: 4. close, Length: 4129, dtype: float64
```

[21]:
```
change = percentage_change[-1]
```

[22]:
```
print(change)
```

```
-0.001814704508292353
```

0.1.2 Stock ML

[24]:
```
# Please Contact for Professional Services
```

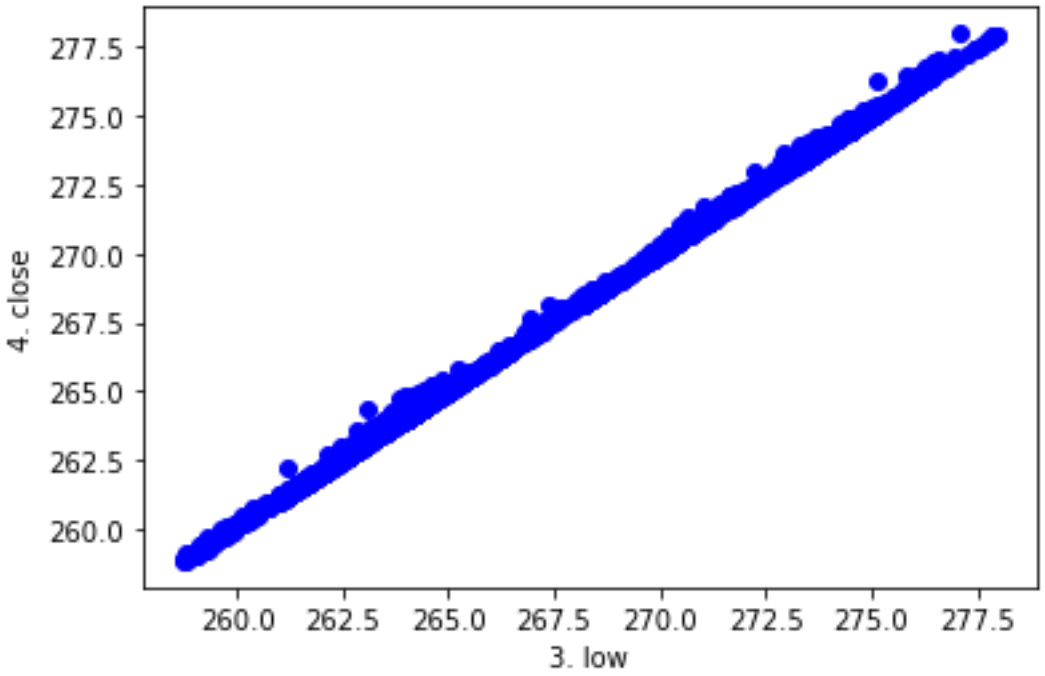
[25]:
```
print(dataML)
```

```
                     1. open  2. high  3. low 4. close 5. volume
date
2020-12-10 20:00:00 266.0000 266.0000 266.0000          204.0
266.0000
2020-12-10 19:25:00 266.0000 266.0000 266.0000          200.0
266.0000
2020-12-10 18:50:00 265.9000 265.9000 265.9000          183.0
265.9000
2020-12-10 18:35:00 264.8100 264.8100 264.8100        13955.0
264.8100
2020-12-10 18:26:00 266.0900 266.1000 266.0900          903.0
266.1000
…                         …       …       …       …       …
2020-11-27 08:27:00 273.3873 273.3873 273.3873          510.0
273.3873
2020-11-27 08:01:00 273.4768 274.0735 273.4768         2670.0
274.0735
2020-11-27 07:29:00 273.2481 273.2481 273.0890          806.0
273.0890
2020-11-27 07:05:00 273.4768 273.9840 273.4768          520.0
273.9840
2020-11-27 07:01:00 273.4868 273.4868 273.4868          634.0
273.4868
[4129 rows x 5 columns]
```

[26]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

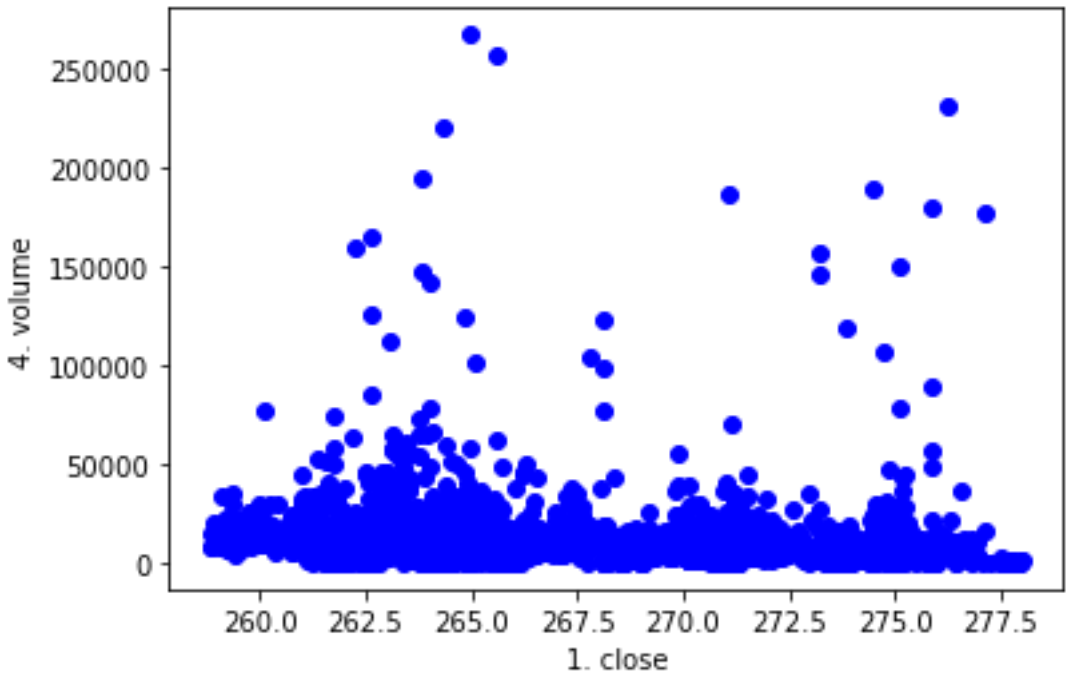
[60]:
```
datML = data[['2. high','4. close']]
```

[62]:
```
# Please Contact for Professional Services
plt.show()
```



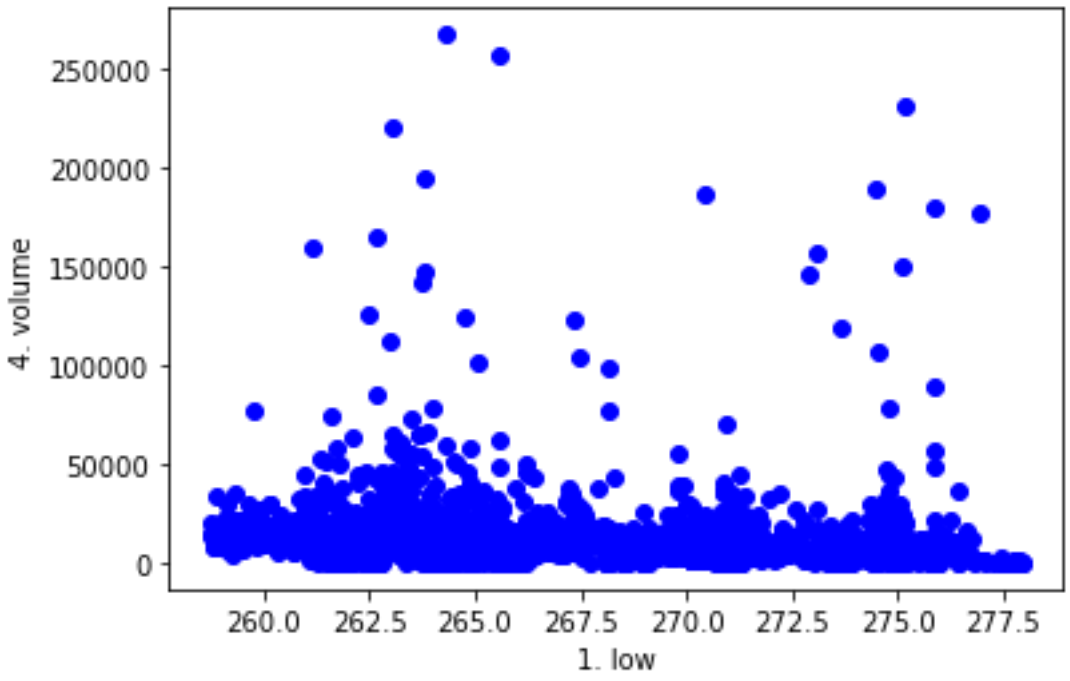
[33]:
```
datML2 = data[['4. close','5. volume']]
```

[38]:
```
# Please Contact for Professional Services
plt.show()
```



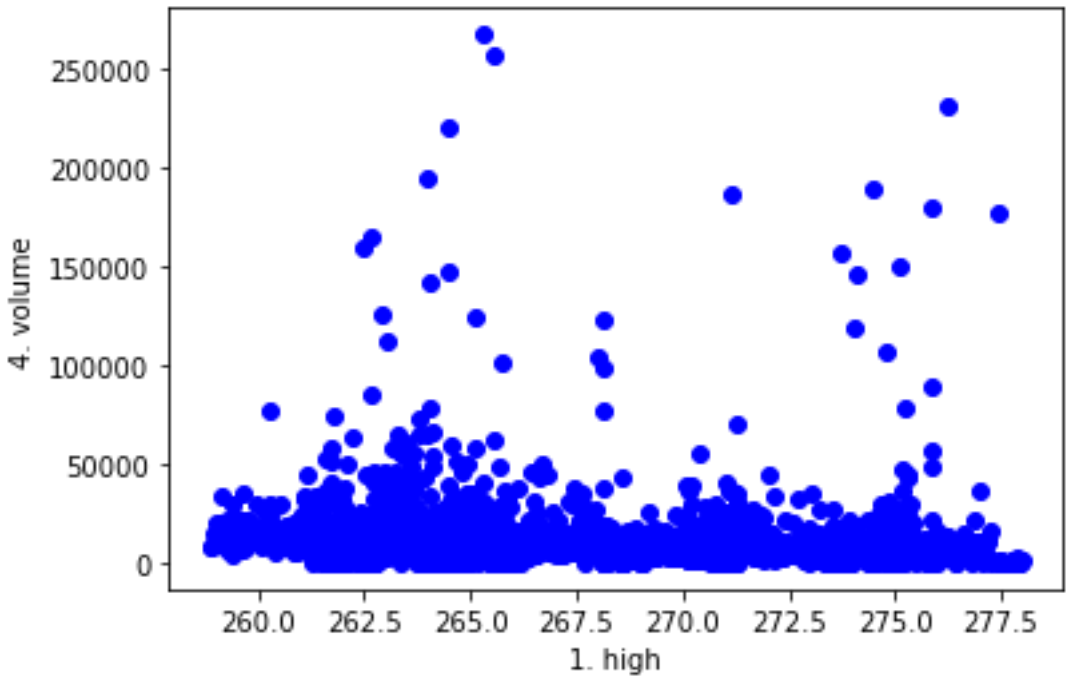
[35]:
```
datML3 = data[['3. low','5. volume']]
```

[37]:
```
# Please Contact for Professional Services
plt.show()
```



[39]:
```
datML4 = data[['2. high','5. volume']]
```

[40]:
```
# Please Contact for Professional Services
plt.show()
```



[41]:
```
train = datML3[:(int((len(data)*0.7)))]
test = datML3[(int((len(data)*0.3))):]
```

[44]:
```
# Please Contact for Professional Services
regr.fit(train_x,train_y)
```

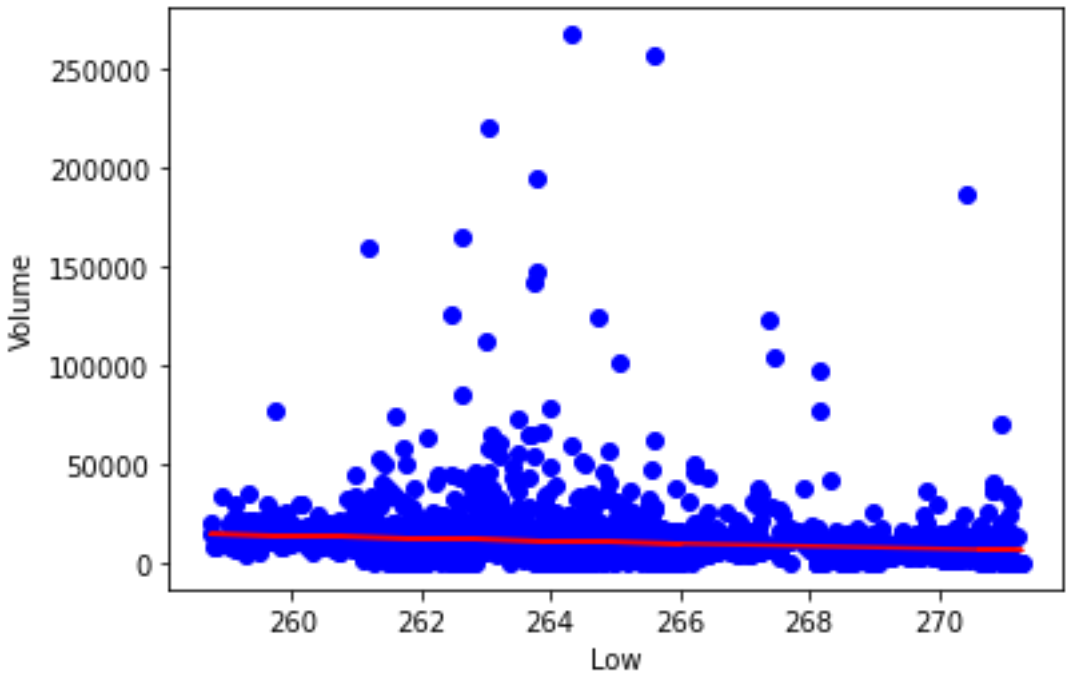
[44]:
```
LinearRegression()
```

[46]:
```
print ('coefficients : ',regr.coef_)
#Slope print ('Intercept :
',regr.intercept_) #Intercept
```

```
coefficients : [[-649.34138257]]
Intercept : [182458.55257284]
```

[50]:
```
plt.scatter(train['3. low'], train['5. volume'],
color="blue") plt.plot(train_x, regr.coef_*train_x +
regr.intercept_, "-r") plt.xlabel('Low')
plt.ylabel('Volume')
```

[50]:
```
Text(0, 0.5, 'Volume')
```



[51]:
```
# Please Contact for Professional Services
return predicted_values
```

[54]:
```
# Please Contact for Professional
Services print ('Estimated Volume
:',estimatd_volume)
```

```
Estimated Volume : -142212.13871063487
```

[55]:
```
# Please Contact for Professional
Services test_y_ = regr.predict(test_x)
```

[59]:
```
print('Mean absolute error: %.2f' % np.mean(np.absolute(test_y_ -
test_y))) print('Mean sum of squares (MSE): %.2f' % np.mean((test_y_
- test_y) ** 2)) print('R2-score: %.2f' % r2_score(test_y_ , test_y)
)
```

```
Mean absolute error: 6104.30
Mean sum of squares (MSE): 237280186.53
R2-score: -24.77
```

[ ]:
```

```