CRYPTO_MODEL_AUTOGEN

June 1, 2020

1 FROM TERMINAL, RUN THIS LINE:

jupyter nbconvert –execute –to pdf CRYPTO_MODEL_AUTOGEN.ipynb

[1]:
```
%%javascript
IPython.OutputArea.auto_scroll_threshold = 9999
```

```
<IPython.core.display.Javascript object>
```

[2]:
```
from model import MODEL
import utils, plotting, statistics
import pandas as pd
```

[3]:
```
start_date = '2020-01-01' #your start date for the model.
day_hour = 18 #your intra-day time at which to evaluate the model.
tickers = 'BTC-USD'
```

[4]:
```
model = MODEL(tickers=tickers)
model.get_data(start=start_date, interval='60m')
model.data.tail(2)
```

```
[**********************100%**********************]  1 of 1 completed
```

[4]:
```
                                BTC-USD
Datetime
2020-06-01 21:00:00+01:00  9584.559570
2020-06-01 21:04:34+01:00  9584.041016
```

[5]:
```
start_date_range = pd.Timestamp(2020, 1, 1, day_hour)
date_range = utils.create_date_range(start_date=start_date_range)
model.apply_date_filter(date_range, force_apply=True)
model.data.tail()
```

```
[INFO]: filter applied.
```

[5]:
```
                                BTC-USD
Datetime
```

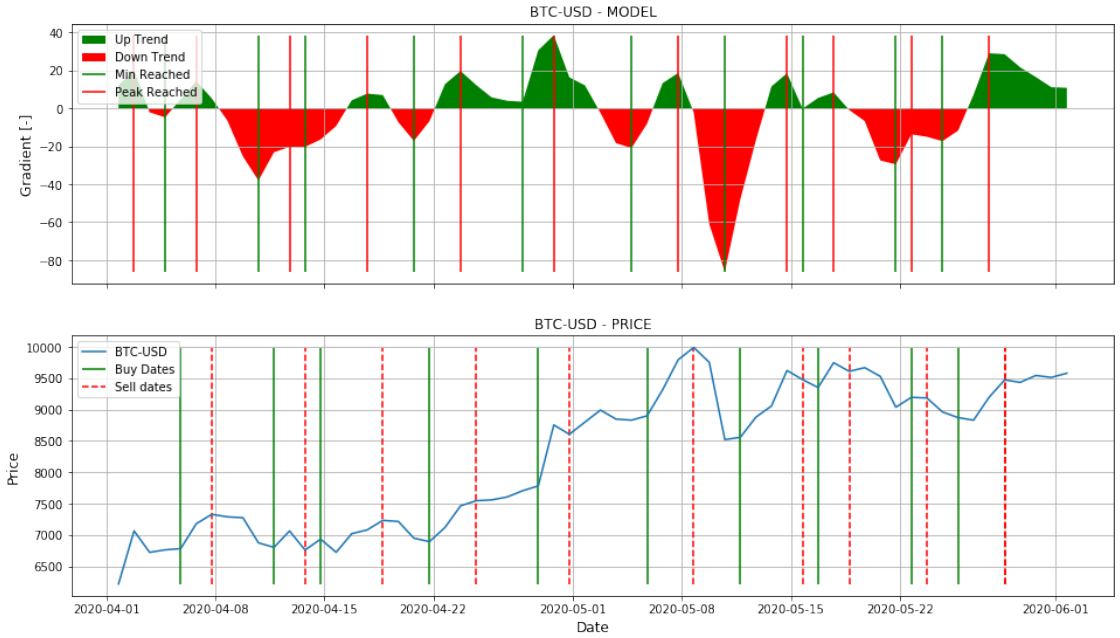
```
2020-05-28 18:00:00+01:00  9472.532227
2020-05-29 18:00:00+01:00  9433.040039
2020-05-30 18:00:00+01:00  9543.111328
2020-05-31 18:00:00+01:00  9512.514648
2020-06-01 18:00:00+01:00  9578.904297
```

[6]:
```
model.eval_model()
```

```
================================================================================
|                                PRICE MODEL                                   |
|                                Version 0.3                                   |
|                  Authors: Marco Rosenzweig & Patrick Lorenz                  |
--------------------------------------------------------------------------------
|                            ticker = ['BTC-USD']                              |
|                           start date = 01/01/20                              |
|                            end date = 06/01/20                               |
|                    initial investment per ticker = 200                       |
================================================================================
[INIT]: Initialising model for tickers: ['BTC-USD']
[INIT]: Successfully initialized model.
********************************************************************************
[TICKER]: BTC-USD
[INFO]: No tax paid.
--------------------------------------------------------------------------------
[SUMMARY]: Average trade win: 6.3764488874%
[SUMMARY]: Average trade loss: -1.0746106601%
[SUMMARY]: Efficiency: 80.77%
[SUMMARY]: NET WIN: 477.26
================================================================================
```

[7]:
```
plotting.plot_model(model, tickers='BTC-USD', plot_from_date='2020-04-01')
```

```
[INFO]: New sell signal was detected for last value: 9578.904296875.
```

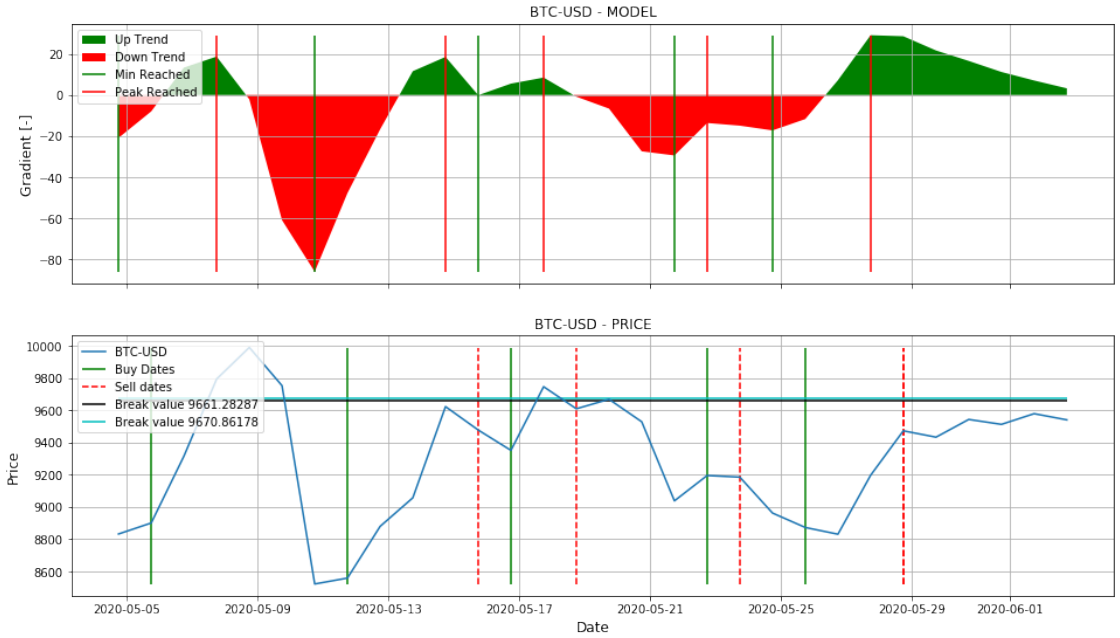


[8]:
```
imag_model = model.copy_model()
imag_model.append_timedelta(timedelta=1)
imag_model.comp_break_values(tickers='all', parallel_computing=True)
imag_model._init_model()
```

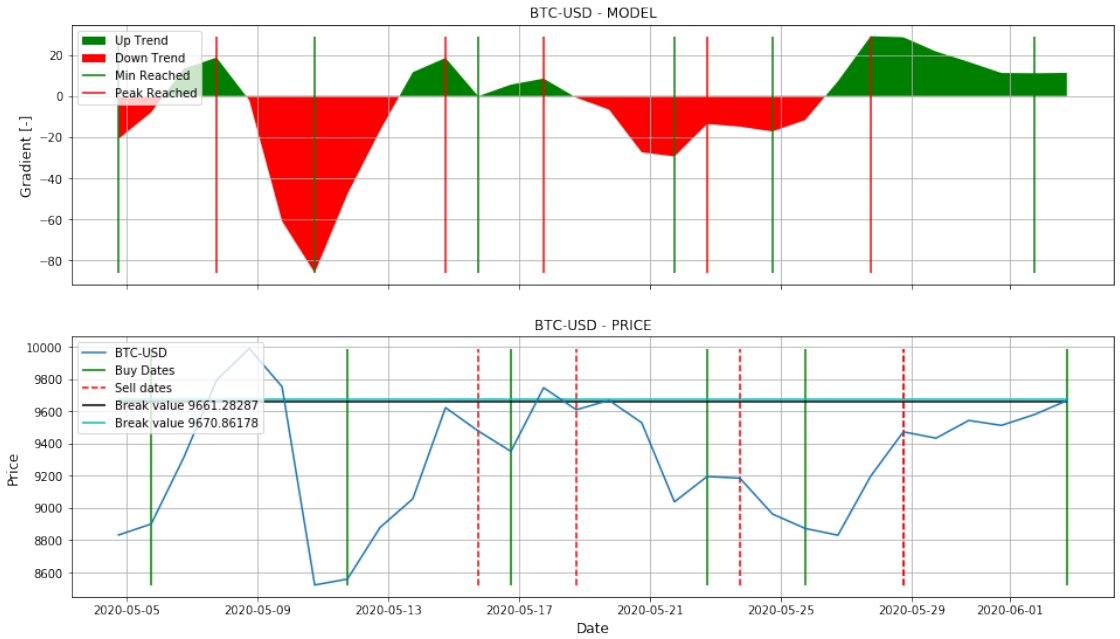
```
[INFO]: New data was appended.
[INFO]: Compute break values with 30.00% deviation
[INFO]: Current ticker: BTC-USD
[INFO]: Using 10 processes.
[INFO]: Current values: {'BTC-USD': 9578.904296875}
[INFO]: Break values: {'BTC-USD': array([9661.28287383, 9670.86177813])}
[INFO]: Tolerances: {'BTC-USD': array([82.37857695, 91.95748125])}
[INIT]: Initialising model for tickers: ['BTC-USD']
[INIT]: Successfully initialized model.
**********************************************************************************
```

[9]:
```
imag_model.show_possibilities(plot_from_date='2020-05-04', switch_axes=False)
```

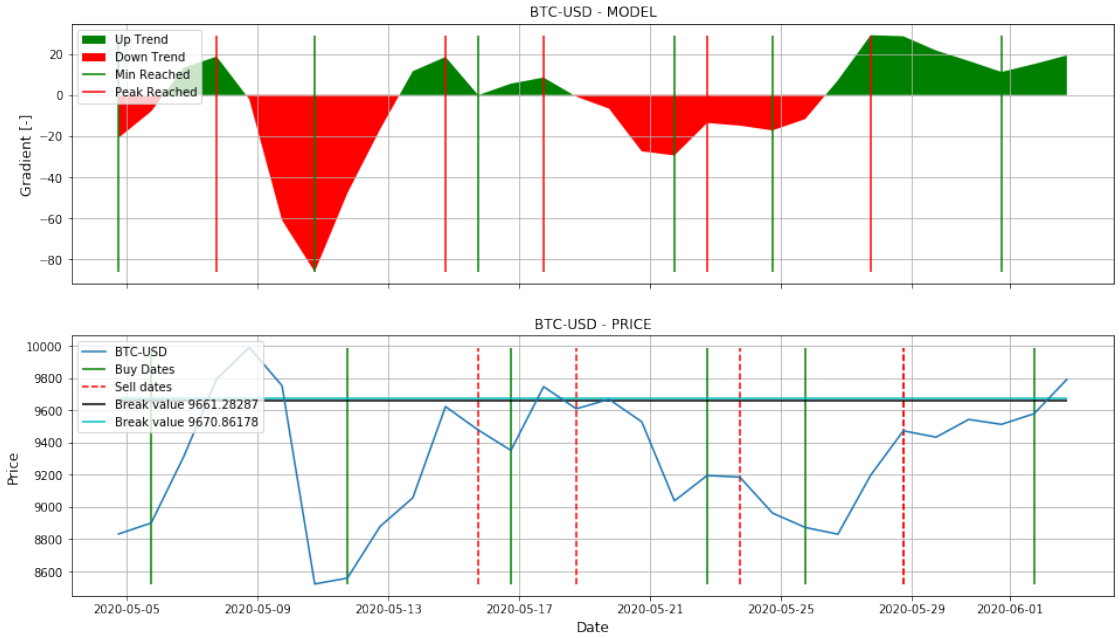
```
==================================================================================
[INFO]: Current ticker: BTC-USD
----------------------------------------------------------------------------------
[INFO]: Result for value: 9540.516837905952
[INFO]: New sell signal was detected for last value: 9540.516837905952.
```



[INFO]: Result for value: 9666.07232597725

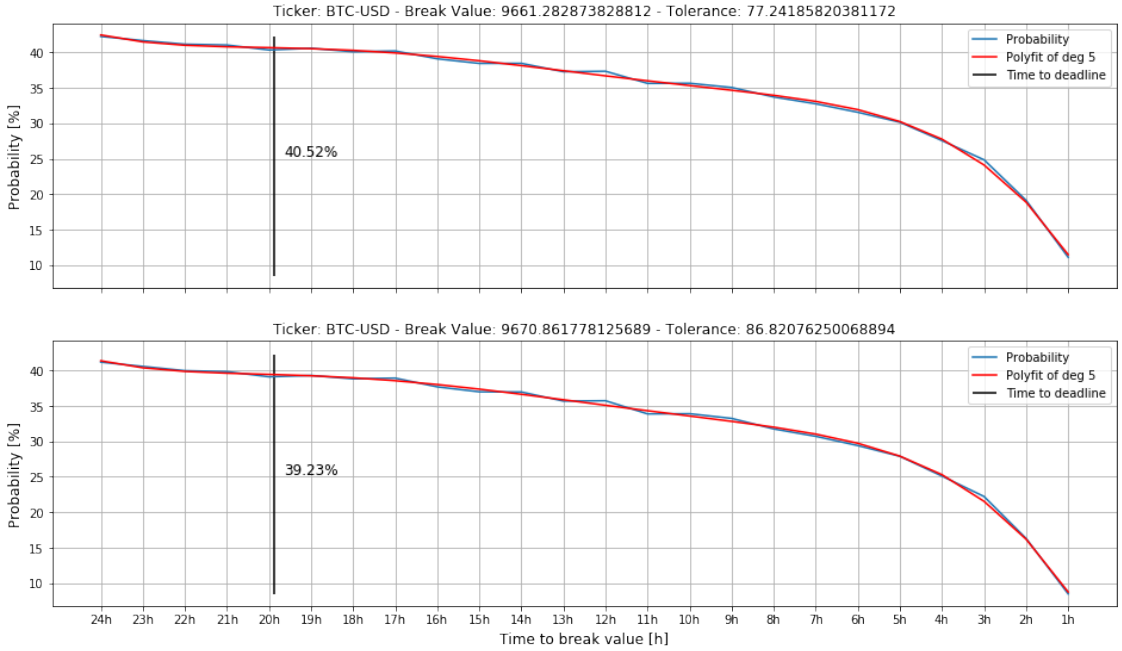


[INFO]: Result for value: 9791.74755035226



[10]:
```
statistics.calc_probs(model=imag_model, tickers='all',␣
↪auto_update_tolerances=True)
```

```
================================================================================
[INFO]: Current ticker: BTC-USD
[**********************100%**********************]  1 of 1 completed
[STATS-INFO]: Auto update of tolerances!
[**********************100%**********************]  1 of 1 completed
[STATS-INFO]: Current value: 9584.041015625!
[STATS-INFO]: New tolerances: [77.2418582 86.8207625]!
[STATS-EVAL]: Probability for tol=77.24186: 40.52%
[STATS-EVAL]: Probability for tol=86.82076: 39.23%
[STATS-EVAL]: Probability between: 1.29%
```


Ticker: BTC-USD - Break Value: 9661.282873828812 - Tolerance: 77.24185820381172
Probability
Polyfit of deg 5
Time to deadline
40.52%
Probability [%]
40
35
30
25
20
15
10

Ticker: BTC-USD - Break Value: 9670.861778125689 - Tolerance: 86.82076250068894
Probability
Polyfit of deg 5
Time to deadline
39.23%
Probability [%]
40
35
30
25
20
15
10
24h
23h
22h
21h
20h
19h
18h
17h
16h
15h
14h
13h
12h
11h
10h
9h
8h
7h
6h
5h
4h
3h
2h
1h
Time to break value [h]