CRYPTO_MODEL_AUTOGEN

June 2, 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
2020-06-02 18:00:00+01:00  9494.106445
2020-06-02 18:29:34+01:00  9491.486328
```

[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
```

```
2020-05-29 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
2020-06-02 18:00:00+01:00  9494.106445
```

[6]:

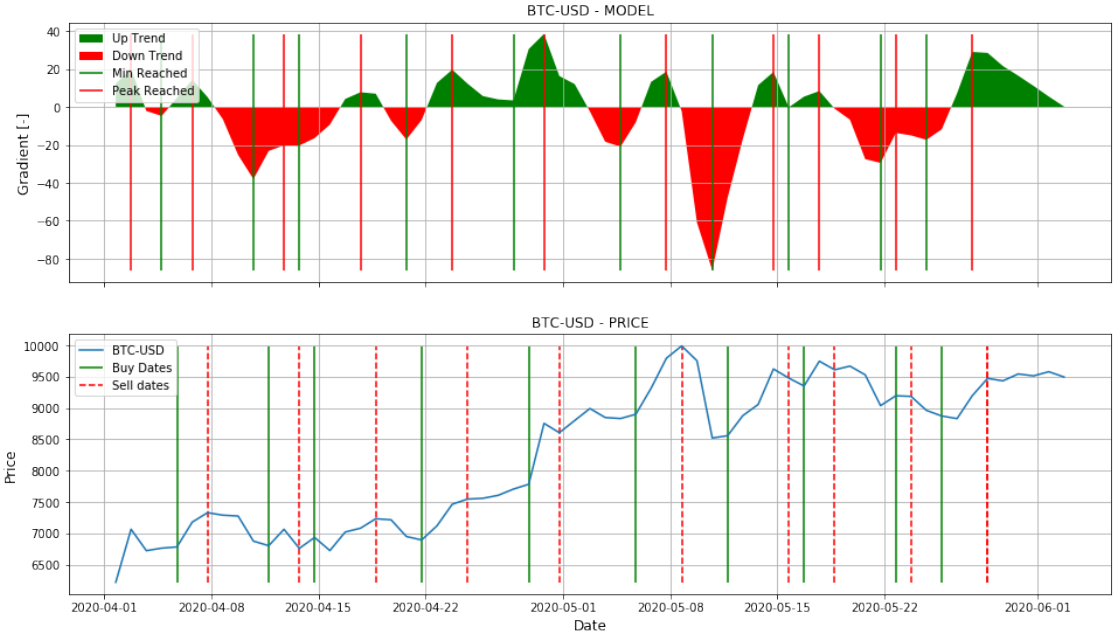
```
model.eval_model()
```

```
================================================================================
|                                 PRICE MODEL                                  |
|                                 Version 0.3                                  |
|                  Authors: Marco Rosenzweig & Patrick Lorenz                  |
--------------------------------------------------------------------------------
|                             ticker = ['BTC-USD']                             |
|                            start date = 01/01/20                             |
|                             end date = 06/02/20                              |
|                     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: 9494.1064453125.
```



[8]:

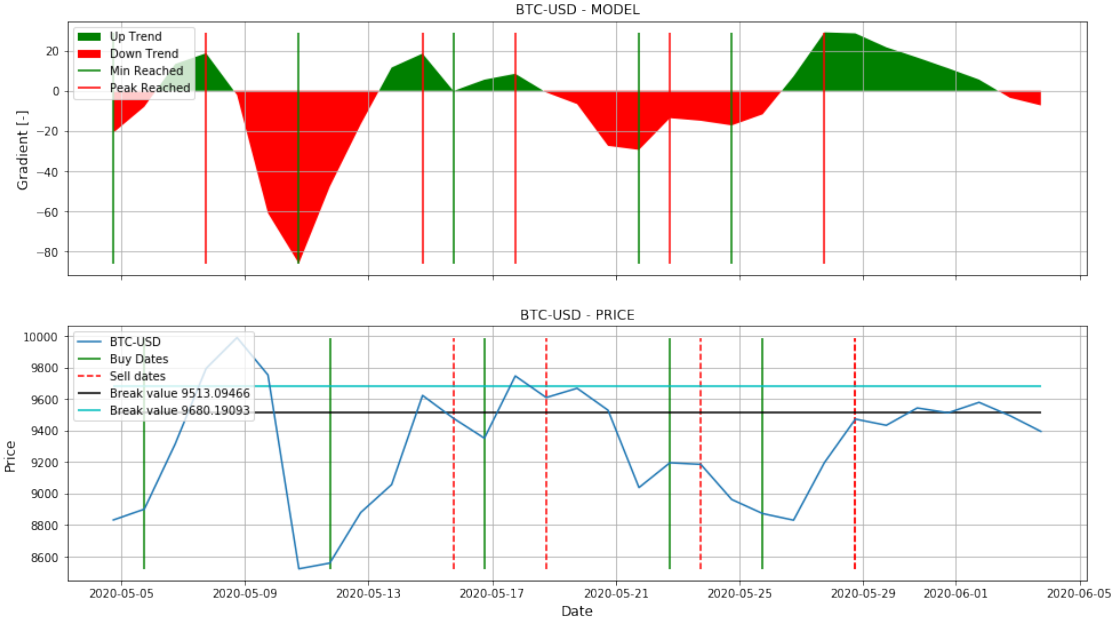
```
imag_model = model.copy_model()
imag_model.append_timedelta(timedelta=1)
imag_model.comp_break_values(tickers='all', parallel_computing=True)
imag_model._init_model()
```

```
[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': 9494.1064453125}
[INFO]: Break values: {'BTC-USD': array([9513.0946582 , 9680.19093164])}
[INFO]: Tolerances: {'BTC-USD': array([ 18.98821289, 186.08448633])}
[INIT]: Initialising model for tickers: ['BTC-USD']
[INIT]: Successfully initialized model.
********************************************************************************
```

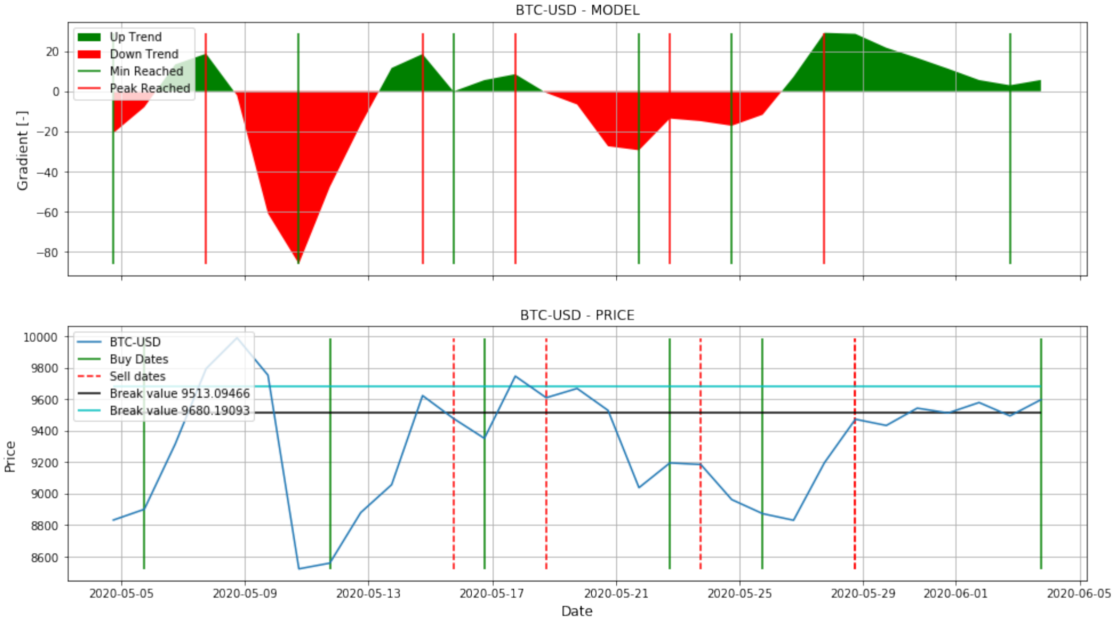
[9]:

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

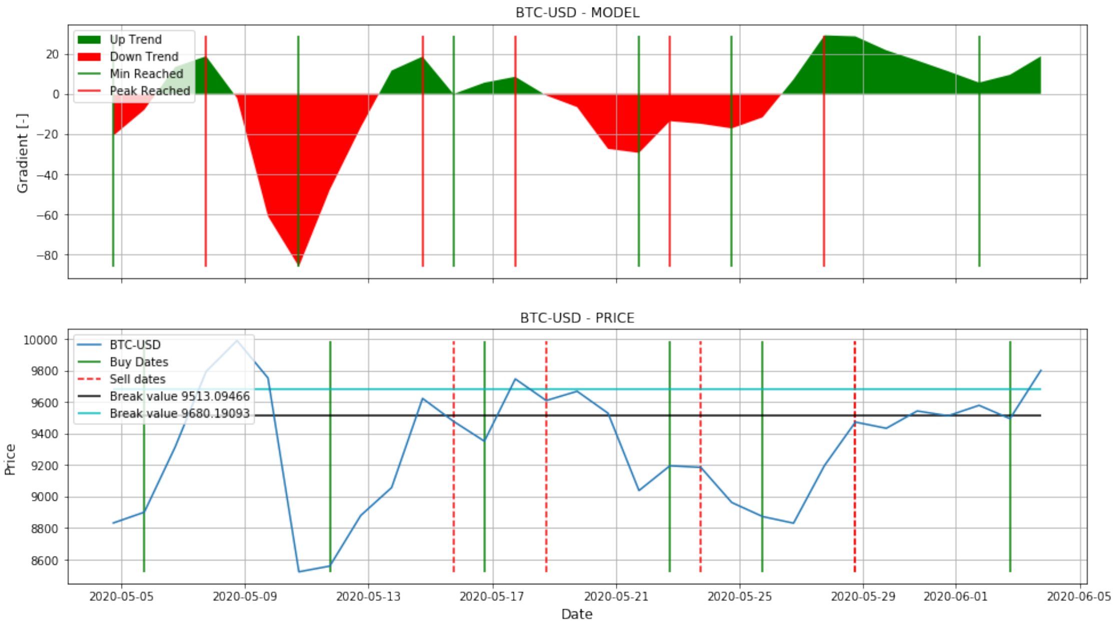
```
================================================================================
[INFO]: Current ticker: BTC-USD
--------------------------------------------------------------------------------
[INFO]: Result for value: 9394.180974975814
[INFO]: New sell signal was detected for last value: 9394.180974975814.
```



```
--------------------------------------------------------------------------------
[INFO]: Result for value: 9596.642794922112
```

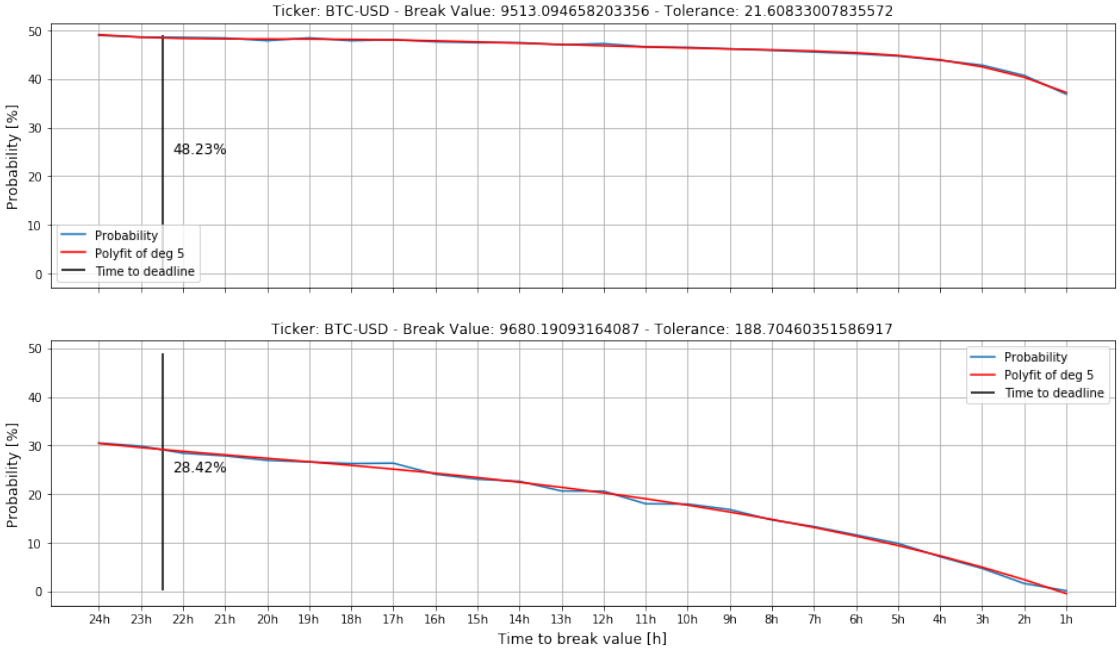


```
--------------------------------------------------------------------------------
[INFO]: Result for value: 9801.19331828638
```



```
[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: 9491.486328125!
[STATS-INFO]: New tolerances: [ 21.60833008 188.70460352]!
[STATS-EVAL]: Probability for tol=21.60833: 48.23%
[STATS-EVAL]: Probability for tol=188.70460: 28.42%
[STATS-EVAL]: Probability between: 19.82%
```


Ticker: BTC-USD - Break Value: 9513.094658203356 - Tolerance: 21.60833007835572
Probability [%]
0
10
20
30
40
50
48.23%
Probability
Polyfit of deg 5
Time to deadline
Ticker: BTC-USD - Break Value: 9680.19093164087 - Tolerance: 188.70460351586917
Probability [%]
0
10
20
30
40
50
28.42%
Probability
Polyfit of deg 5
Time to deadline
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]