Bot_crypto_trading

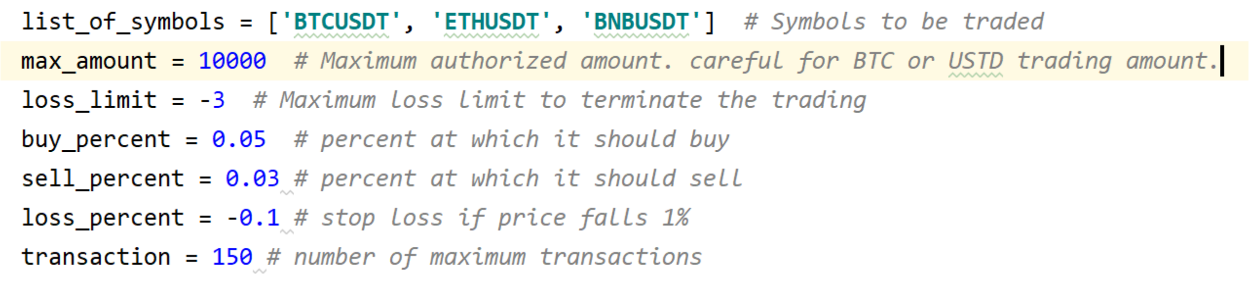
As per the requirement doc file, all the points have been addressed.

To run actual trading bot: python3 trading_bot_real.py

To run test bot: python3 trading_bot_test.py

Different test case results are saved in the excel sheet.

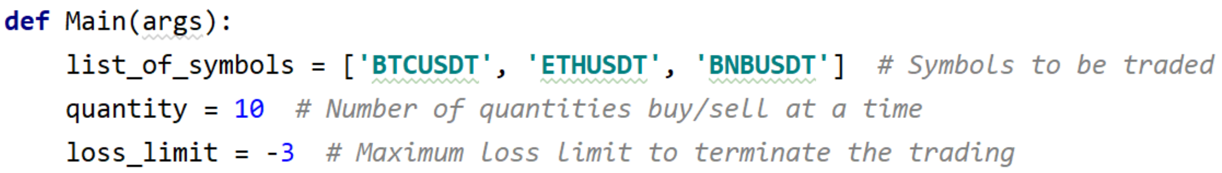
- Max_profit_30$.csv records 30 $ of profit with following conditions



```
list_of_symbols = ['BTCUSDT', 'ETHUSDT', 'BNBUSDT']  # Symbols to be traded
max_amount = 10000  # Maximum authorized amount. careful for BTC or USTD trading amount.
loss_limit = -3  # Maximum loss limit to terminate the trading
buy_percent = 0.05  # percent at which it should buy
sell_percent = 0.03 # percent at which it should sell
loss_percent = -0.1 # stop loss if price falls 1%
transaction = 150 # number of maximum transactions
```

- fifteen transaction test.csv records 12 $ profit in 15 transactions. With buy percent = 0.5 and sell_percent = 0.2 and loss_percent = -1

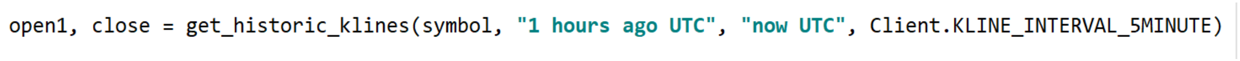
1) About point 1, the bot should analyze the selected currency



```
def Main(args):
    list_of_symbols = ['BTCUSDT', 'ETHUSDT', 'BNBUSDT']  # Symbols to be traded
    quantity = 10  # Number of quantities buy/sell at a time
    loss_limit = -3  # Maximum loss limit to terminate the trading
```

It will continuously analyze the listed symbols for trade opportunities.

2) About point 2, the user can edit the candle frame using

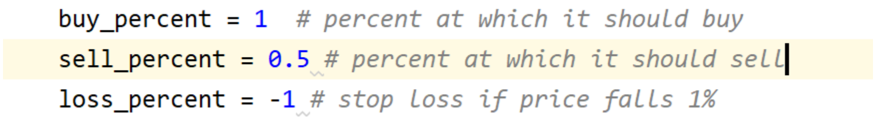


```
open1, close = get_historic_klines(symbol, "1 hours ago UTC", "now UTC", Client.KLINE_INTERVAL_5MINUTE)
```

`Edit Client.KLINE_INTERVAL_5MINUTE to Client.KLINE_INTERVAL_1MINUTE or Client.KLINE_INTERVAL_10MINUTE`

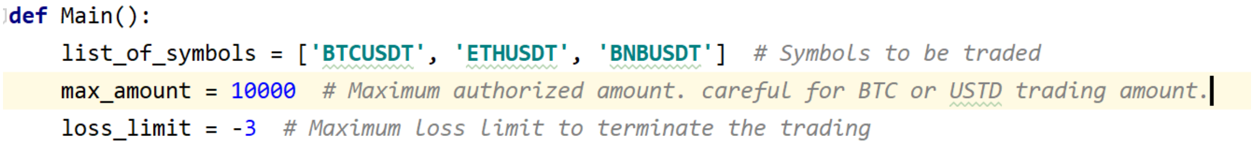
`Refer to comments of get_historical_klines function in the code for more details.`

3) About point 3, change the following parameters as desired for buy, sell and stop_loss limit



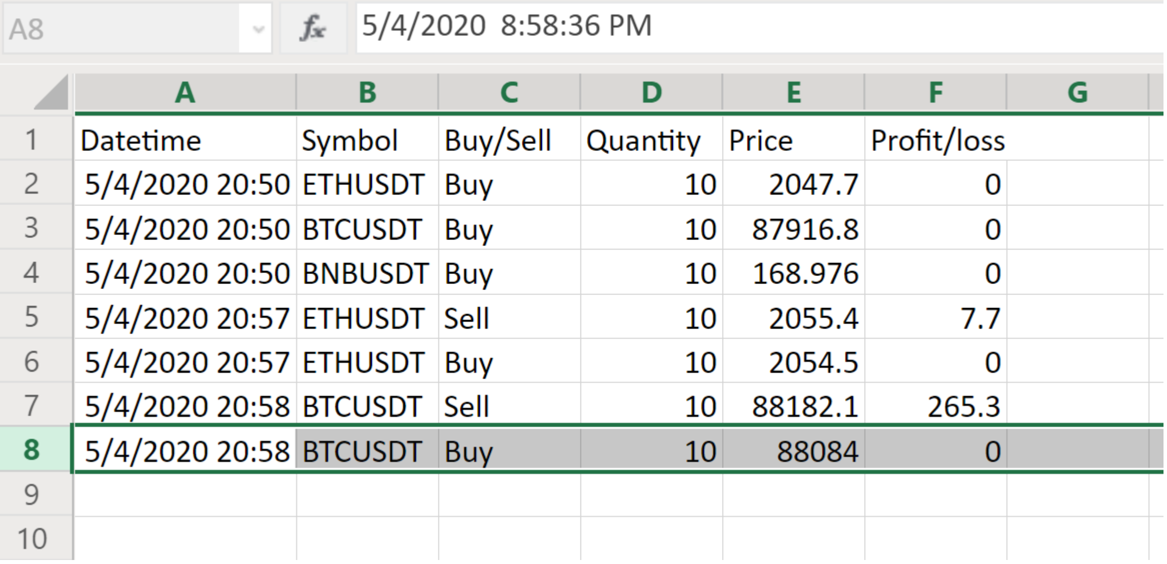
```
buy_percent = 1  # percent at which it should buy
sell_percent = 0.5 # percent at which it should sell
loss_percent = -1 # stop loss if price falls 1%
```

4) About point 4, Amount bot is authorized to trade when trading multiple symbols together.



```
def Main():
    list_of_symbols = ['BTCUSDT', 'ETHUSDT', 'BNBUSDT']  # Symbols to be traded
    max_amount = 10000  # Maximum authorized amount. careful for BTC or USTD trading amount.
    loss_limit = -3  # Maximum loss limit to terminate the trading
```

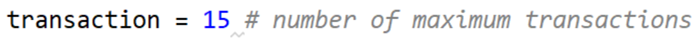
5) About point 5, excel file will be saved with these columns



A8 | 5/4/2020 8:58:36 PM

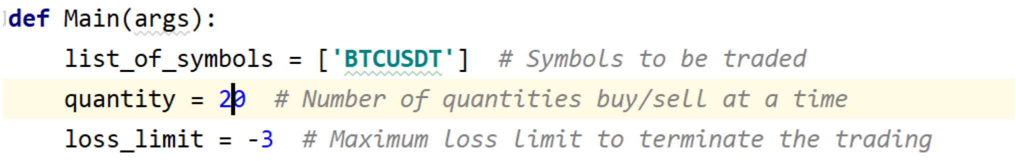
	A	B	C	D	E	F	G
1	Datetime	Symbol	Buy/Sell	Quantity	Price	Profit/loss	
2	5/4/2020 20:50	ETHUSDT	Buy	10	2047.7	0	
3	5/4/2020 20:50	BTCUSDT	Buy	10	87916.8	0	
4	5/4/2020 20:50	BNBUSDT	Buy	10	168.976	0	
5	5/4/2020 20:57	ETHUSDT	Sell	10	2055.4	7.7	
6	5/4/2020 20:57	ETHUSDT	Buy	10	2054.5	0	
7	5/4/2020 20:58	BTCUSDT	Sell	10	88182.1	265.3	
8	5/4/2020 20:58	BTCUSDT	Buy	10	88084	0	
9							
10							

6) About point 6, user-defined maximum transaction limit



```
transaction = 15 # number of maximum transactions
```

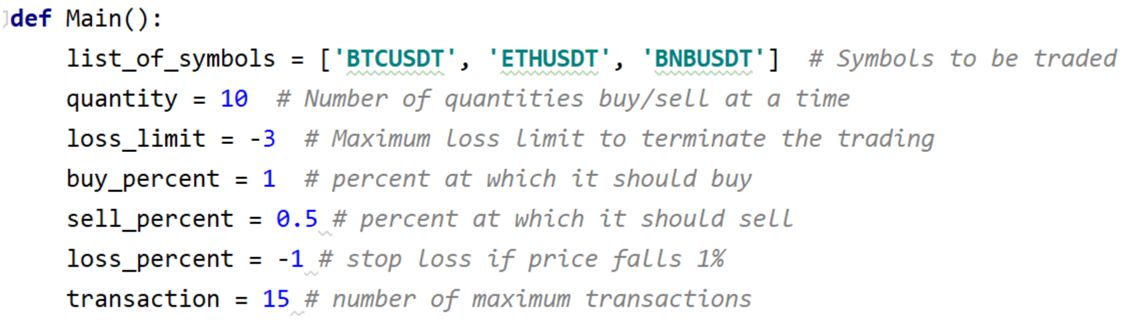
7) About point 7, bot to stop trading when the loss reaches -3%



```
def Main(args):
    list_of_symbols = ['BTCUSDT']  # Symbols to be traded
    quantity = 20  # Number of quantities buy/sell at a time
    loss_limit = -3  # Maximum loss limit to terminate the trading
```

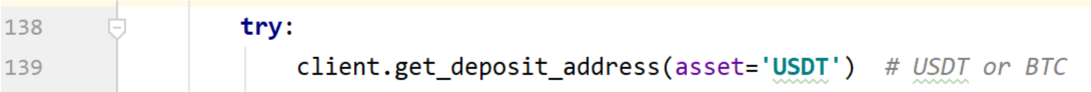
Change loss_limit to your custom loss limit.

8) About point 8, the user has a full customization option here as well as at other places of the code.



```
def Main():
    list_of_symbols = ['BTCUSDT', 'ETHUSDT', 'BNBUSDT']  # Symbols to be traded
    quantity = 10  # Number of quantities buy/sell at a time
    loss_limit = -3  # Maximum loss limit to terminate the trading
    buy_percent = 1  # percent at which it should buy
    sell_percent = 0.5 # percent at which it should sell
    loss_percent = -1 # stop loss if price falls 1%
    transaction = 15 # number of maximum transactions
```

9) About point 9, the user canchoose between USTD or BTC or other



```
try:
    client.get_deposit_address(asset='USDT')  # USDT or BTC
```