CRYPTO TRADING BOT

Fetch Live Data:

The trading bot starts trading by fetching the live data from Binance using the python library and stores in a DataFrame to create buy and sell signals.

This python library is used to connect to the Binance API which provide access to live and historical data (OHLC). It provides quick access to market data for analysis, visualization, indicator development, algorithmic trading, strategy backtesting etc.

Strategy:

The trading bot works with strategy file which is a python file where we define own strategy with the help of a python library to define and use various indicators to create buy and sell rules.

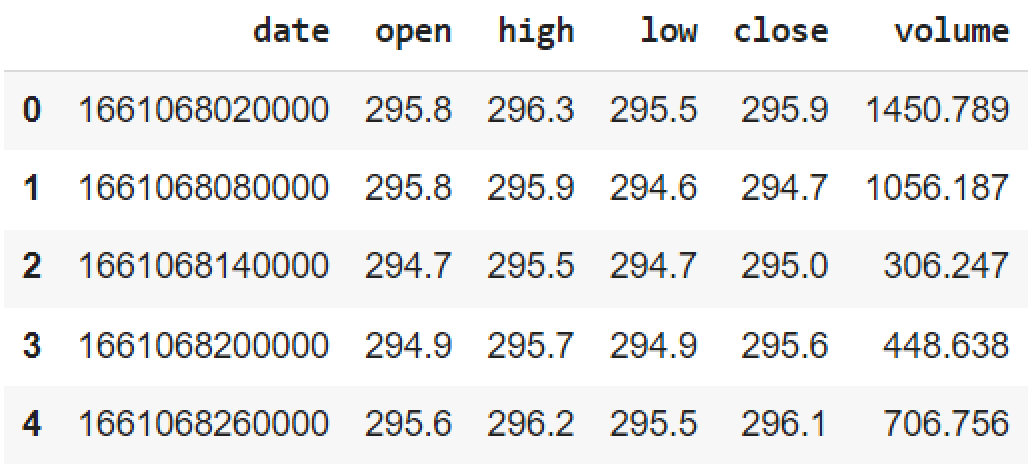
A strategy file contains all the information needed to build a good strategy:

- Indicators
- Entry strategy rules
- Exit strategy rules
- Minimal ROI
- Stoploss

Dataframe:

The trading bot uses pandas to store the candlestick data. Each row in a dataframe corresponds to one candle on a chart, with the latest candle always being the last in the dataframe as the data is sorted by date.

EXAMPLE:



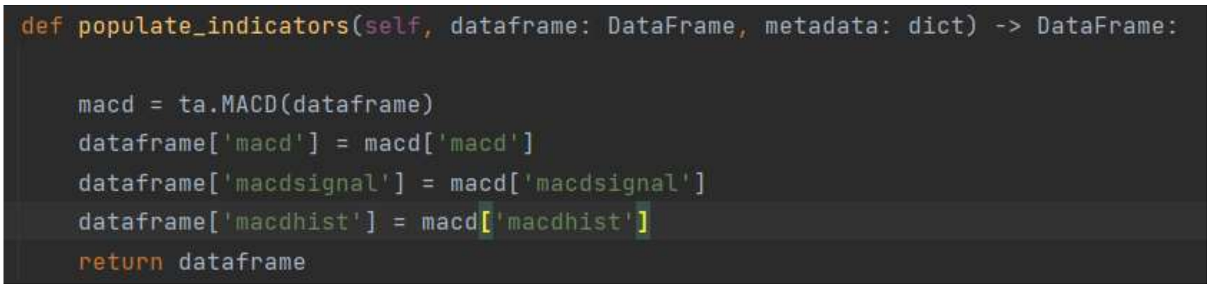
	date	open	high	low	close	volume
0	1661068020000	295.8	296.3	295.5	295.9	1450.789
1	1661068080000	295.8	295.9	294.6	294.7	1056.187
2	1661068140000	294.7	295.5	294.7	295.0	306.247
3	1661068200000	294.9	295.7	294.9	295.6	448.638
4	1661068260000	295.6	296.2	295.5	296.1	706.756

Indicators:

Buy and Sell signals need indicators. we can use predefined indicators with the help of a python library or can add more indicators by extending the list contained in the method populate_indicators() from strategy file.

Populate Indicators:

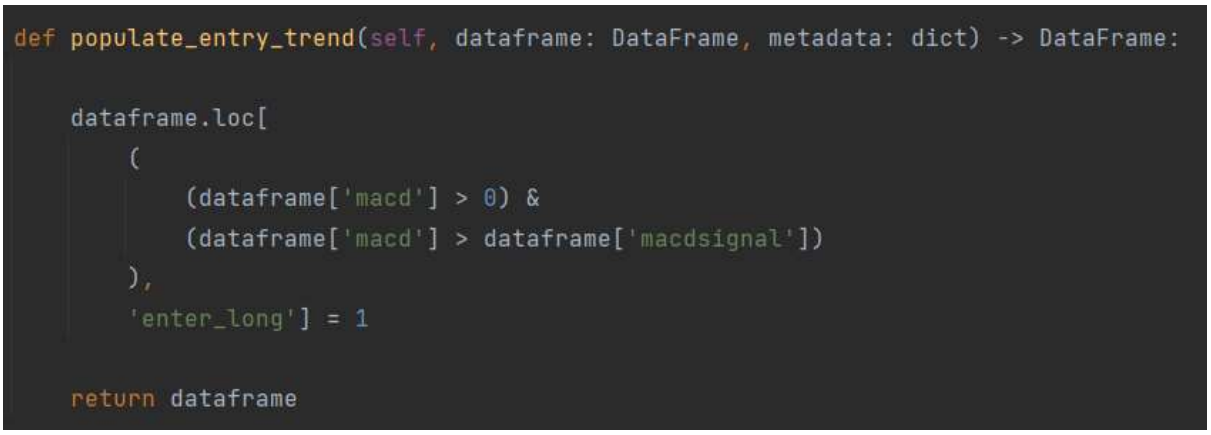
This function is used to add several different indicators to the given DataFrame to initialize and to use for creating buying and selling rules. We can add our indicators in this function.



```python
def populate_indicators(self, dataframe: DataFrame, metadata: dict) -> DataFrame:

    macd = ta.MACD(dataframe)
    dataframe['macd'] = macd['macd']
    dataframe['macdsignal'] = macd['macdsignal']
    dataframe['macdhist'] = macd['macdhist']
    return dataframe
```

Populate entry trend:

Based on the indicators, populates the buy signal for the given dataframe i.e., create rules for bot to take entry and start buying according to the provided conditions. It takes dataframe as input and returns dataframe with the conditions included.

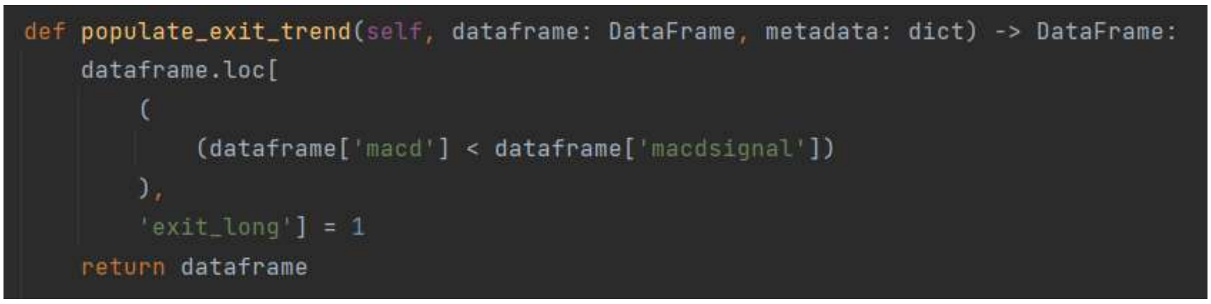


```
def populate_entry_trend(self, dataframe: DataFrame, metadata: dict) -> DataFrame:

    dataframe.loc[
        (
            (dataframe['macd'] > 0) &
            (dataframe['macd'] > dataframe['macdsignal'])
        ),
        'enter_long'] = 1

    return dataframe
```

As, for above screen shot of python script the bot will enter the market when macd line above the zero-line signal. The bot will place buy order when the macd line cut the signal line from below (see in the below live data time series screen shot from Binance).

Populate exit trend:

Based on the indicators, populates the sell signal for the given dataframe i.e., creates rules for bot for exit or sell the coins. It also takes dataframe as input and returns dataframe with the sell conditions included (see in the below live data time series screen shot from Binance).



```
def populate_exit_trend(self, dataframe: DataFrame, metadata: dict) -> DataFrame:
    dataframe.loc[
        (
            (dataframe['macd'] < dataframe['macdsignal'])
        ),
        'exit_long'] = 1
    return dataframe
```

Below I have given a live example of above Moving Average Convergence and divergence (MACD) for the live market on LUNC/BUSD Coin.

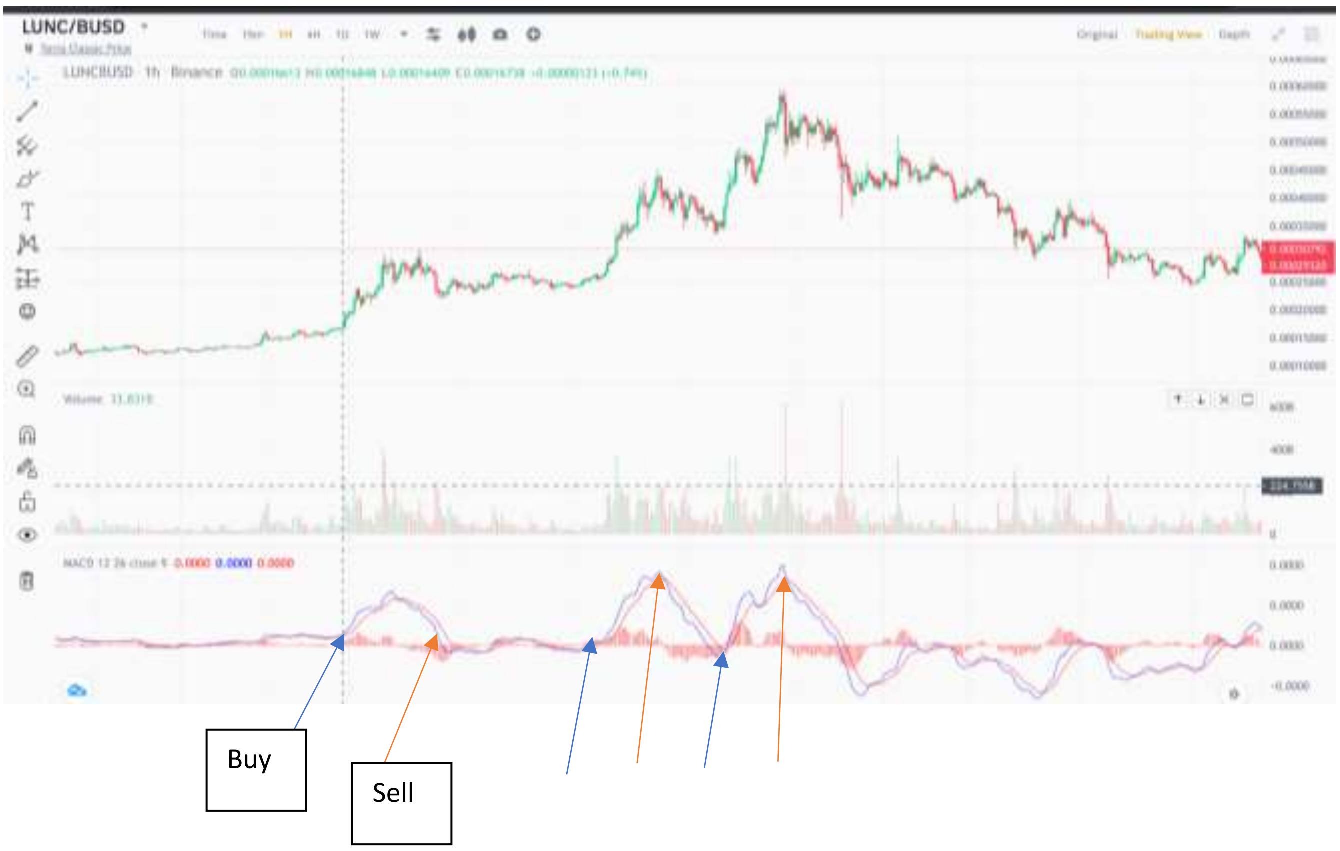
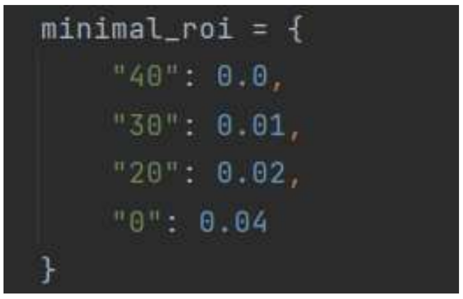


Figure: Blue arrow for buy signal and red is for sell signal respectively.

Minimal ROI

This dictionary defines the minimal Return on Investment (ROI) a trade should reach before exiting, independent from the exit signal.

It is of the following format, with the dictionary key (left side of the colon) being the minutes passed since the trade opened, and the value (right side of the colon) being the percentage.



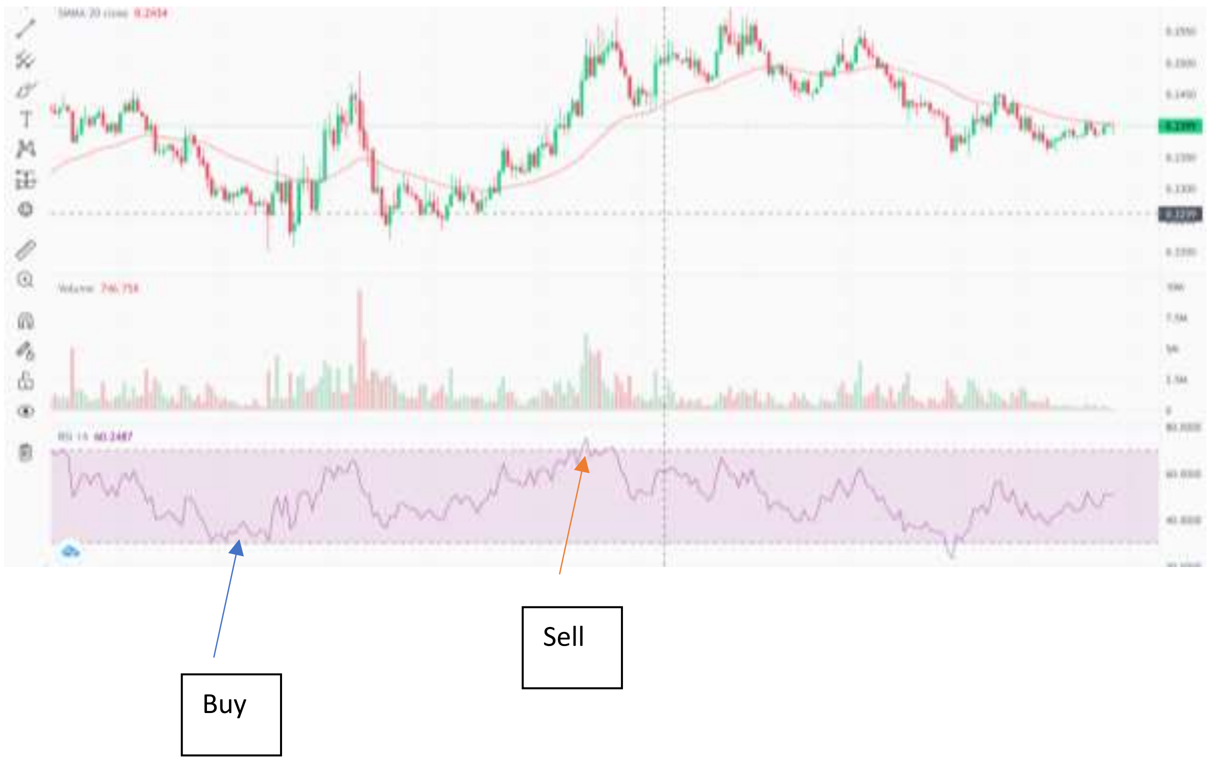
```
minimal_roi = {
    "40": 0.0,
    "30": 0.01,
    "20": 0.02,
    "0": 0.04
}
```

The above configuration would therefore mean:

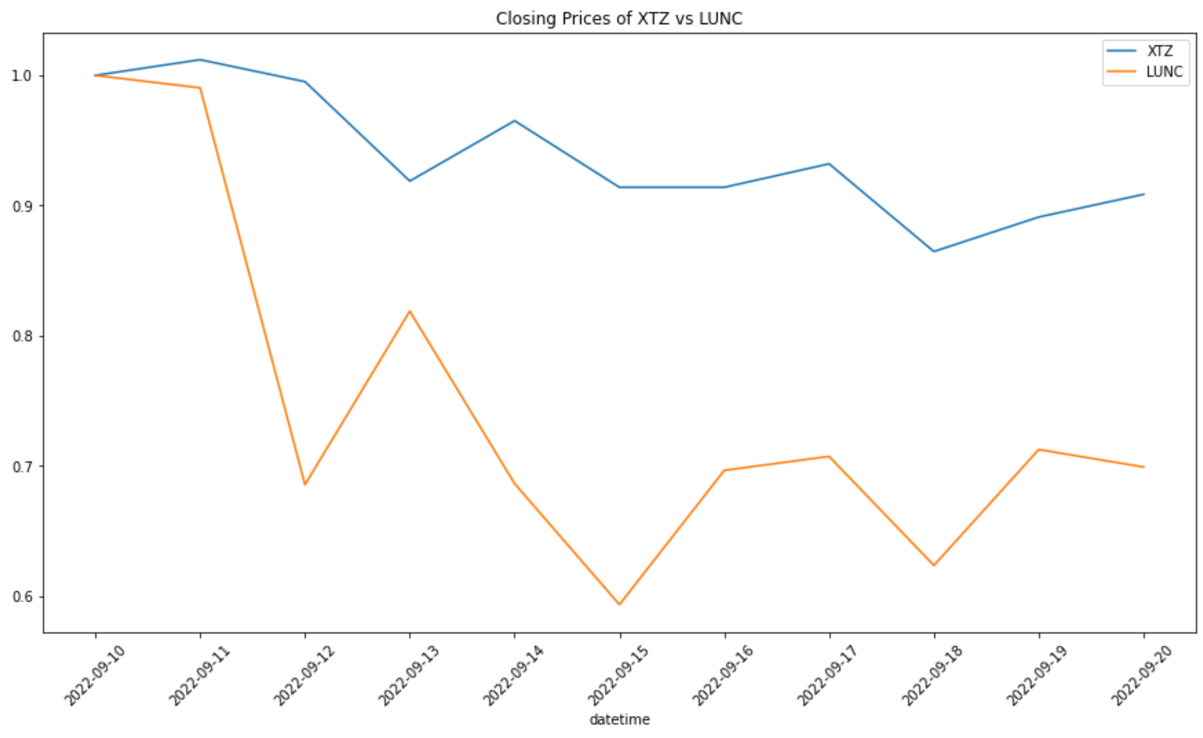
- Exit whenever 5% profit was reached
- Exit when 2% profit was reached (in effect after 20 minutes)
- Exit when 1% profit was reached (in effect after 30 minutes)

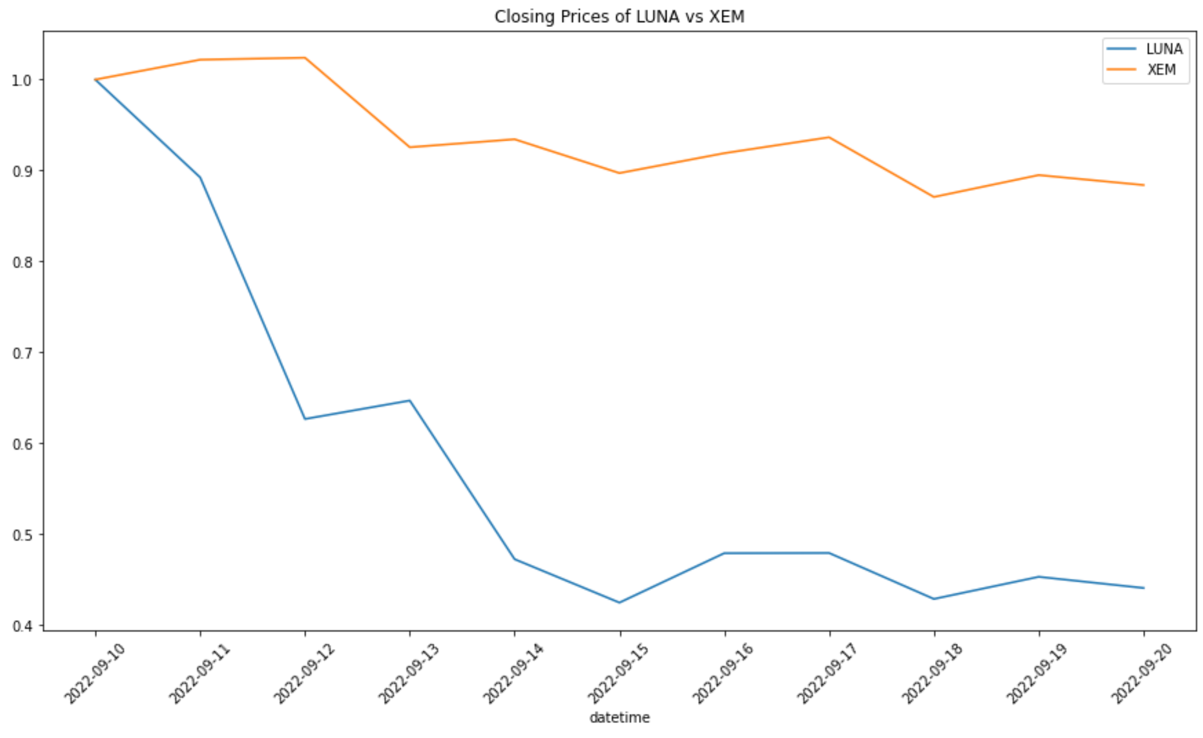
Implementation of Diversification concept of Harry Markowitz in Cryptocurrencies.

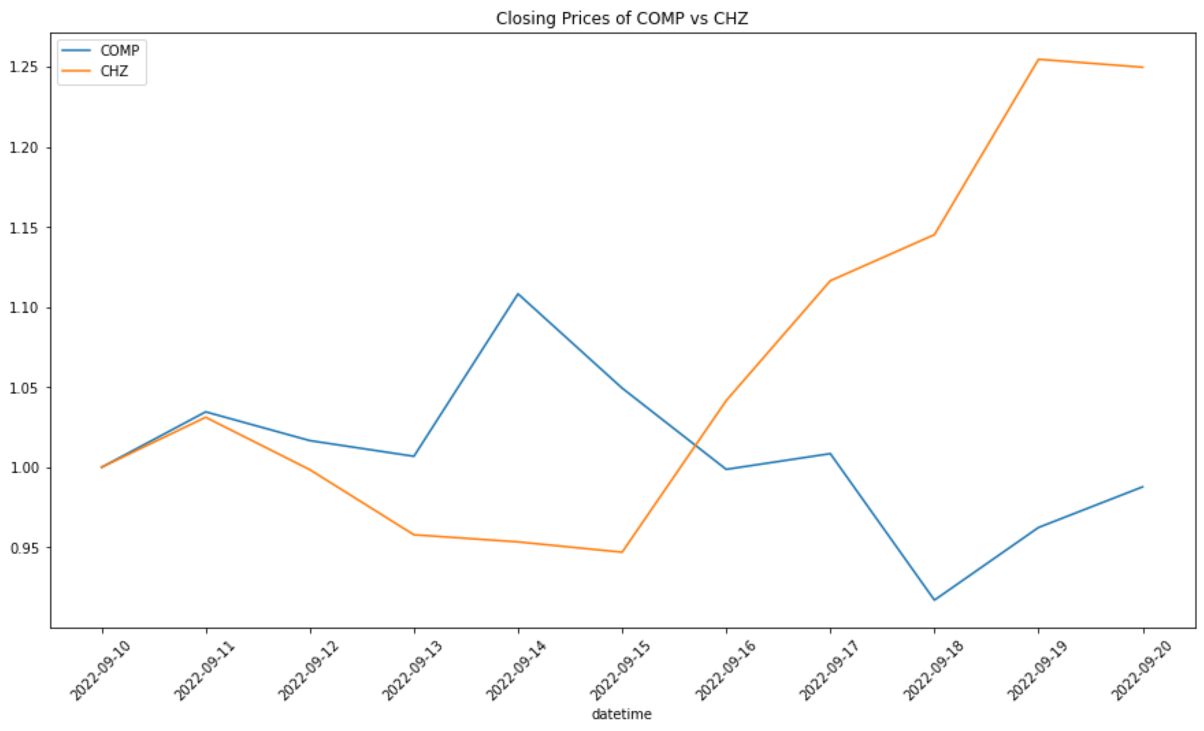
Economist Harry Markowitz introduced Modern portfolio Theory in a 1952, we apply the concept of the diversification on the cryptocurrency. Diversification is a portfolio allocation strategy that are not perfectly positively correlated. We can easily find the corelation matrix of different coins using python. In a simple meaning if a coin goes up then other will go in down. Therefore, we can buy low and sell high. It is clearer from the live market example.

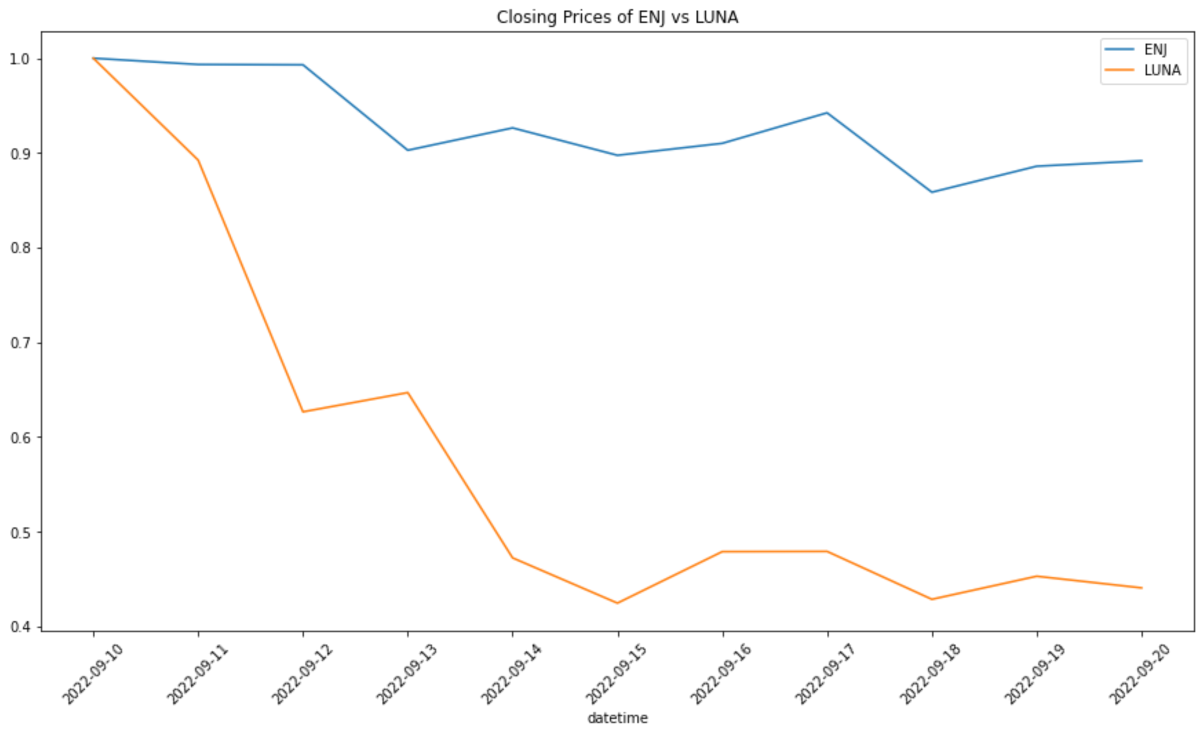


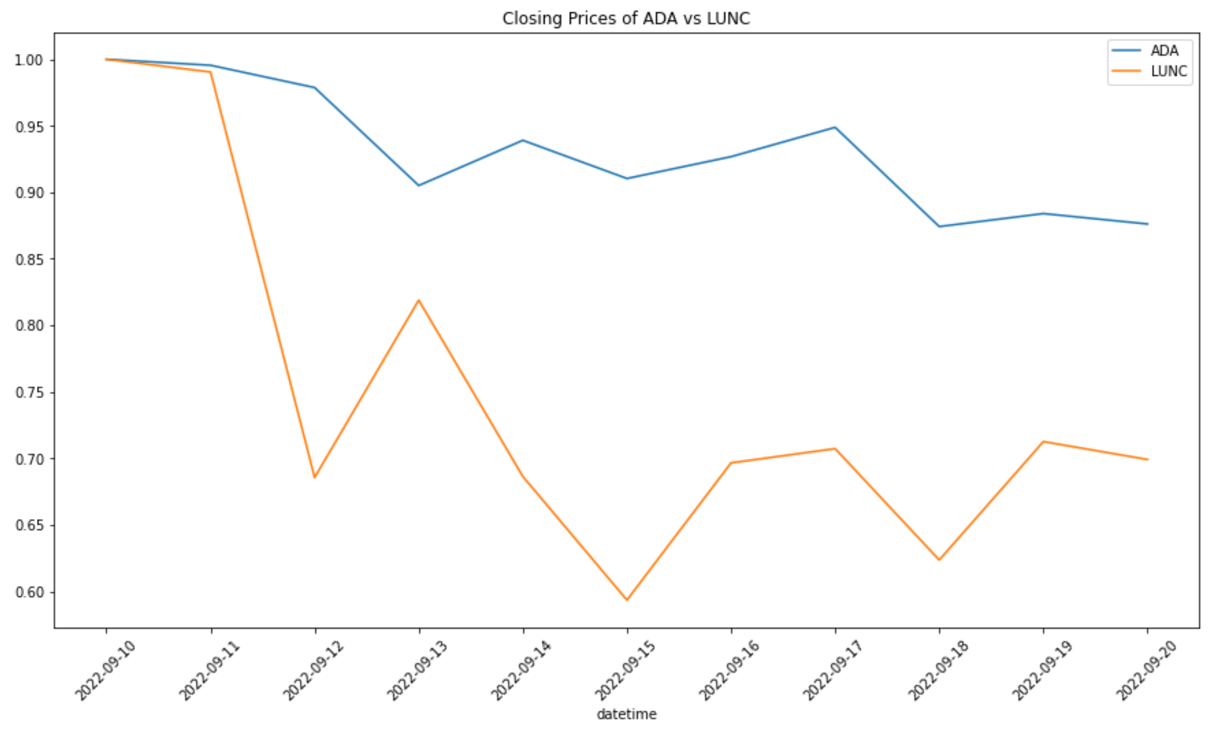
Below I have given some plots of coins which are moving opposite to each other so my idea is to buy low and sell high and this we can do easily. Buy the coins which has RSI less than 30 and sell the coins which has RSI grater than 80. To get more optimum result we can optimize the RSI value for the individual coins.

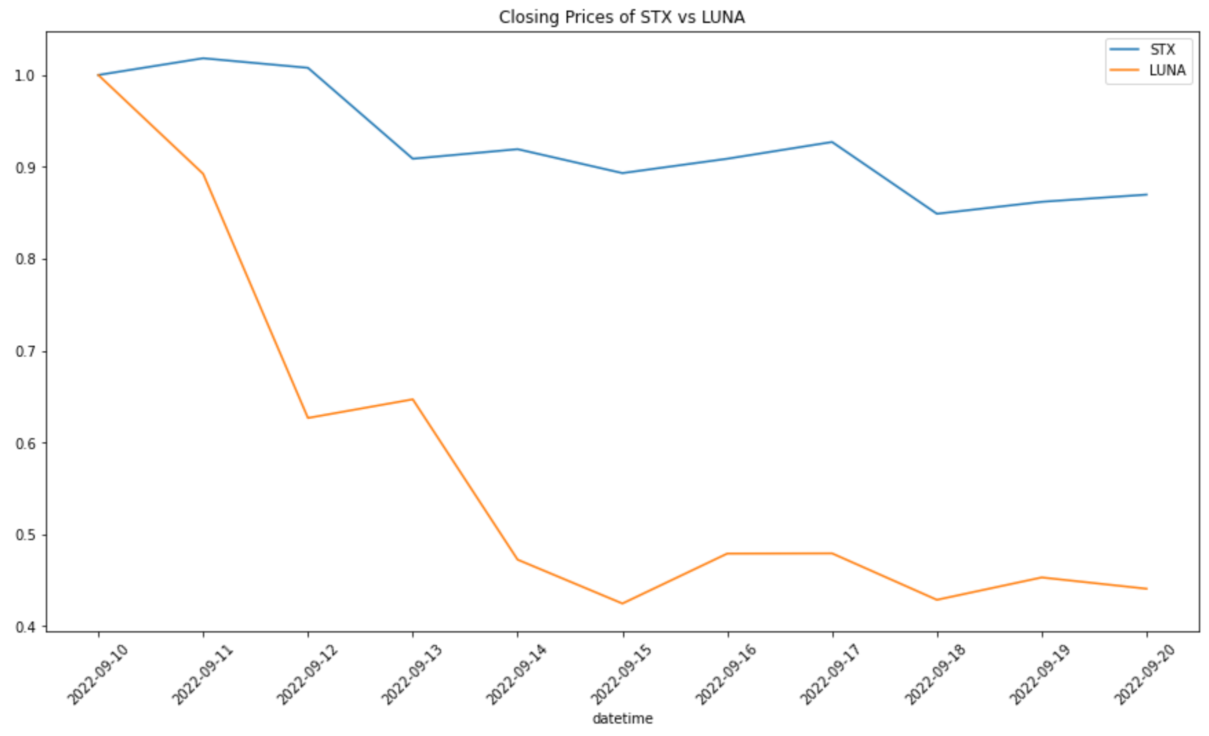

Closing Prices of XTZ vs LUNC
XTZ
LUNC
1.0
0.9
0.8
0.7
0.6
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime

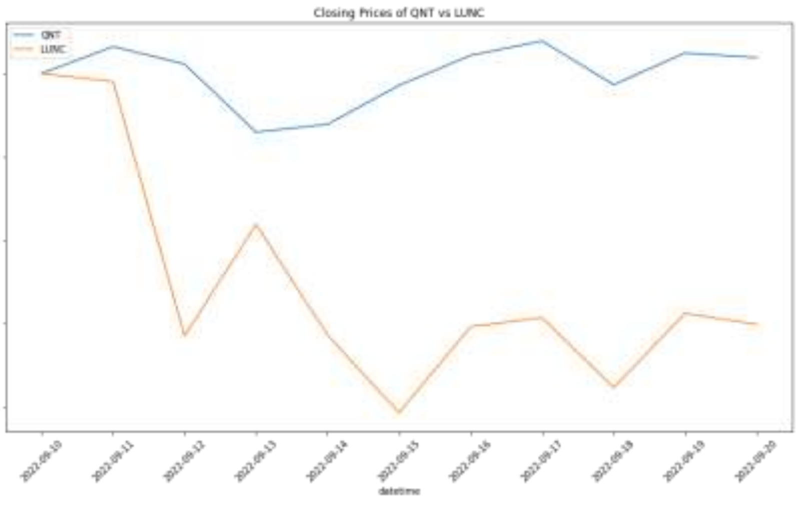

Closing Prices of LUNA vs XEM
LUNA
XEM
1.0
0.9
0.8
0.7
0.6
0.5
0.4
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime

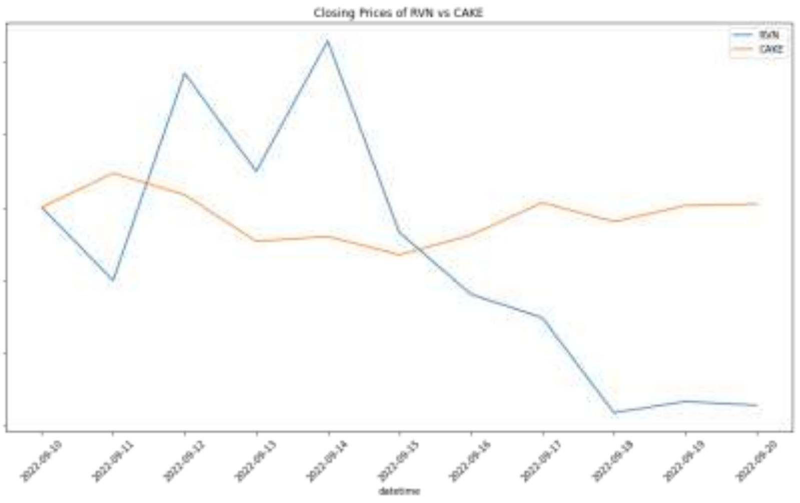

Closing Prices of COMP vs CHZ
COMP
CHZ
1.25
1.20
1.15
1.10
1.05
1.00
0.95
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime

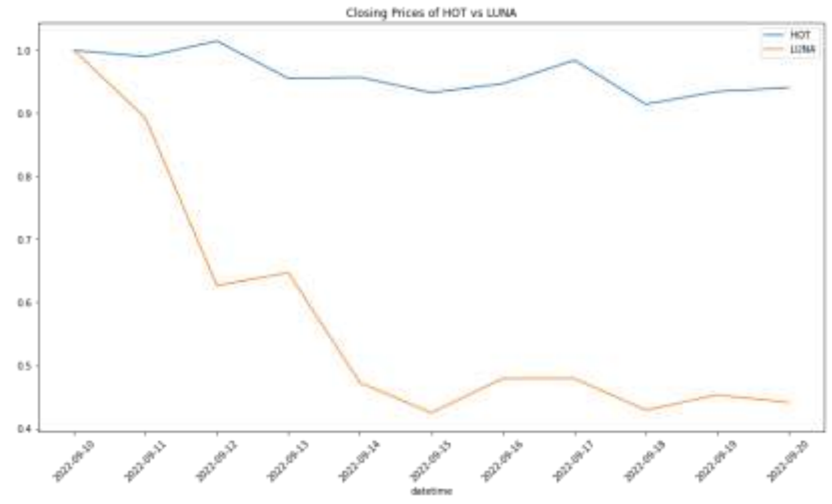

Closing Prices of ENJ vs LUNA
ENJ
LUNA
1.0
0.9
0.8
0.7
0.6
0.5
0.4
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime

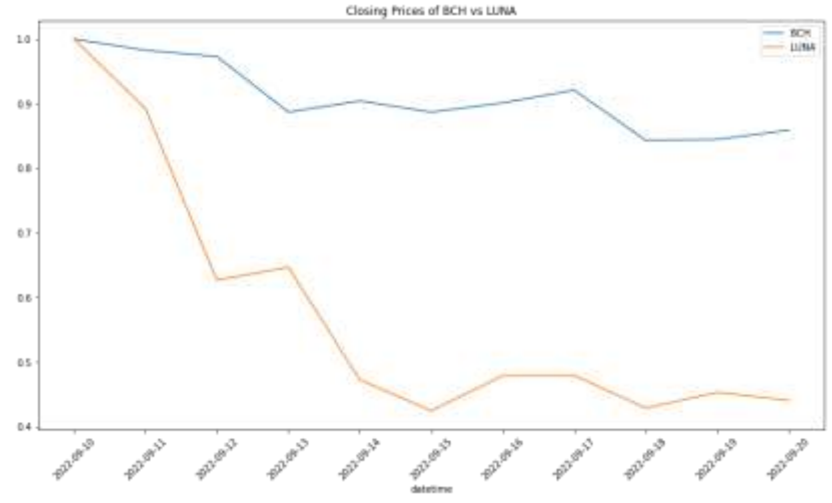

Closing Prices of ADA vs LUNC
ADA
LUNC
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime


Closing Prices of STX vs LUNA
STX
LUNA
1.0
0.9
0.8
0.7
0.6
0.5
0.4
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime


Closing Prices of QNT vs LUNC
QNT
LUNC
1.0
0.9
0.8
0.7
0.6
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime


Closing Prices of RVN vs CAKE
RVN
CAKE
1.2
1.1
1.0
0.9
0.8
0.7
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime


Closing Prices of HOT vs LUNA
HOT
LUNA
1.0
0.9
0.8
0.7
0.6
0.5
0.4
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime


Closing Prices of BCH vs LUNA
BCH
LUNA
1.0
0.9
0.8
0.7
0.6
0.5
0.4
2022-09-10
2022-09-11
2022-09-12
2022-09-13
2022-09-14
2022-09-15
2022-09-16
2022-09-17
2022-09-18
2022-09-19
2022-09-20
datetime