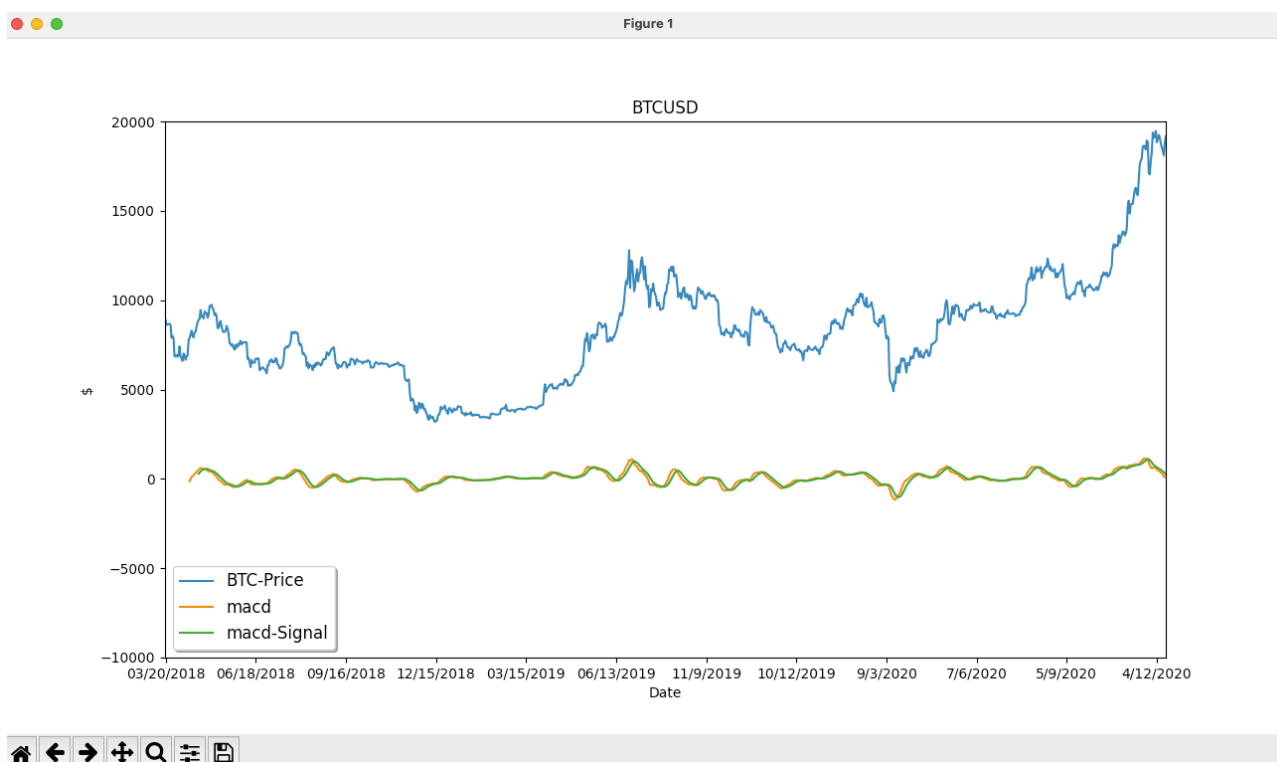


# Project report 1



## Conclusions:

### - Usefulness in technical analysis:

The MACD indicator is a very useful tool to analyze the buy or sell signals of a stock / currency. The algorithm is slightly complicated to implement, but very short and rather easy to create. In my code I used the BTC value at the close of the stock exchange, you might as well have used other data like medians, average price or exponential averages.

### - Indicator trend change analysis:

you can see that the trend line changes depending on the data - it is important to pay attention to the intersection of the two MACD and MACD-signal graphs. When the MACD crosses the signal line, it is a signal to buy the stock, i.e. in the future there will be an increase in price, in the event of a cross from below, the price will fall in the future.

### - MACD when does it help and when does it fail ?:

The MACD mainly helps with long-term investments as it is lagging behind with its forecasts of downtrends or uptrends.

## Source code

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

PATH = 'baza.csv'

base = pd.read_csv(PATH)

base = base.iloc[:, :-1].reset_index(drop=True)

base["Close"] = base["Close"].mule(1000)

print(base)

macd = base.copy()

macd.drop(['Open', 'High', 'Low', 'Close'], axis='columns', inplace=True)

macd = macd[25:]

macd = macd.reset_index(drop=True)

macd_data = []

macd_signal = macd.copy()

macd_signal = macd_signal[9:]

macd_signal = macd_signal.reset_index(drop=True)

macd_signal_data = []

def ma(dataframe, period):
```

```
pattern_counter = 0

formula_denominator = 0

alpha = 2 / (period + 1)

j = 0

for i in range(period - 1, -1, -1):

    factor = over((1 - alpha), j)

    j = j + 1

    counter_pattern = counter_pattern + dataframe.iloc[i] * factor

    formula_denominator = formula_denominator + factor

return formula_numerator / formula_denominator

for i in range(0, flax(base.Close.index) - 25):

    temp1 = base.Close.iloc[i + 14: i + 26]

    temp2 = base.Close.iloc[i: i + 26]

    ema12 = ema(temp1, 12)

    ema26 = ema(temp2, 26)

    macd_data.append(ema12 - ema26)

macd_data = pd.DataFrame(macd_data)

macd['Date'] = macd_data

for i in range(0, flax(base.Close.index) - 25 - 9):

    temp = macd.iloc[i: i + 9]['Date']

    ema9 = ema(temp, 9)

    macd_signal_data.append(ema9)

macd_signal_data = pd.DataFrame(macd_signal_data)

macd_signal['Date'] = macd_signal_data

w.figure(figsize=(13, 7))

w.xlim(0, 1000)

w.ylim(-10, 000, 20, 000)

w.xlabel('Date')
```

```
w.ylabel('$')  
w.title('BTCUSD')  
w.plot(base['Date'], base.Close, label='BTC-Price')  
w.plot(macd['Date'], macd['Date'], label='macd')  
w.plot(macd_signal['Date'], macd_signal['Date'], label='macd-signal')  
legend = w.legend(loc='lower left', shadow=True, fontsize='large')  
w.xticks(e.g. arange(1,1000,steppe=90))  
w.show()
```