

Projekt 1 - Sprawozdanie

Mikołaj Bąk 188968

Plik .csv zawierający ceny akcji firmy State Street Corp pobrałem ze strony stooq.pl. Dane obejmują 1000 dni poprzedzających dzień 11.03.2023 (nie licząc weekendów, a zatem od 22.03.2019). Do wczytania pliku użyłem biblioteki pandas:

```
def main():  
  
    import pandas as pd  
    import numpy as np  
    import matplotlib.pyplot as plt  
  
    dataframe = pd.read_csv(r'C:\Users\mikalaj\PycharmProjects\StockMarketMACD\stt_us_d.csv')  
    dates = dataframe.iloc[:, 0]  
    prices = dataframe.iloc[:, 1]
```

Następnie stworzyłem funkcję do obliczania wykładniczej średniej kroczącej na podstawie wzoru:

$$EMA_N = \frac{p_0 + (1 - \alpha)p_1 + (1 - \alpha)^2 p_2 + \dots + (1 - \alpha)^N p_N}{1 + (1 - \alpha) + (1 - \alpha)^2 + \dots + (1 - \alpha)^N}$$

```
def ema(prices, day, N):  
    alpha = 2/(N+1)  
    upper_sum = 0  
    lower_sum = 0  
    for i in range(0, N):  
        p = float(prices.iloc[day-i])  
        upper_sum += p * math.pow(1 - alpha, i)  
        lower_sum += math.pow(1 - alpha, i)  
    return upper_sum/lower_sum
```

Wykorzystałem to do obliczenia wskaźnika MACD dla każdego dnia począwszy od 26. według wzoru:

$$MACD = EMA_{12} - EMA_{26}$$

```
macd = pd.DataFrame(index=np.arange(dates.size), columns=np.arange(1))  
for i in range(0, 26):  
    macd.iloc[i, 0] = 0  
for i in range(26, dates.size):  
    ema_26 = ema(prices, i, 26)  
    ema_12 = ema(prices, i, 12)  
    macd.iloc[i, 0] = ema_12 - ema_26  
print('MACD:')  
print(macd)
```

Dla dni 0 – 26 przyjąłem MACD równy 0 – nie da się obliczyć wskaźnika dla tych dni przez brak danych z dni poprzedzających.

Następnie obliczyłem SIGNAL (od 35. dnia, gdyż robię wykładniczą średnią kroczącą z 9 ostatnich MACD, a MACD zaczyna się 26. dnia; $26 + 9 = 35$).

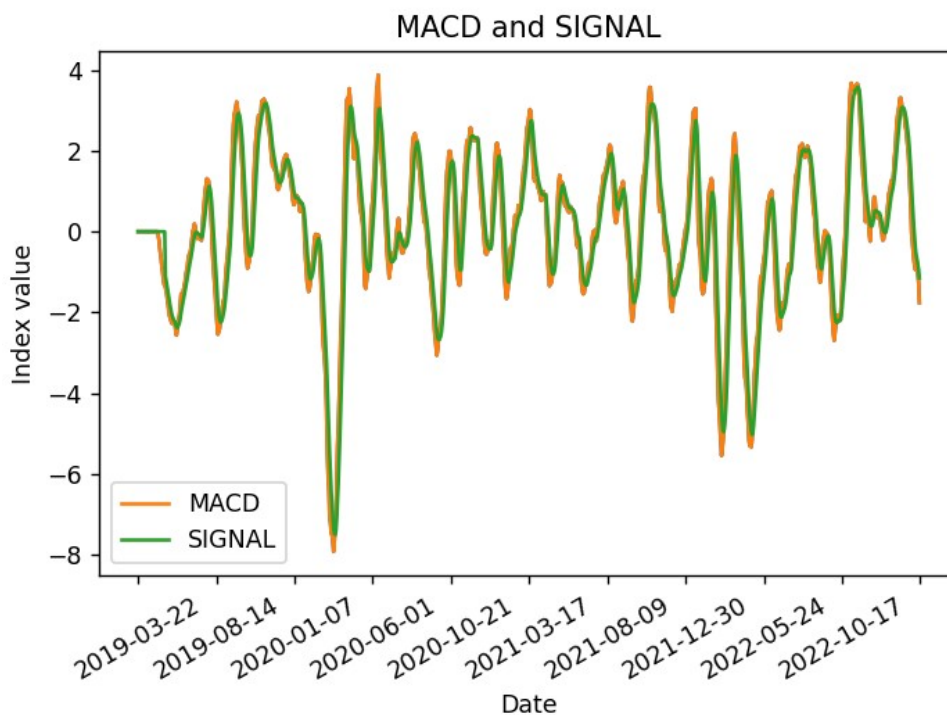
```
signal = pd.DataFrame(index=np.arange(dates.size), columns=np.arange(1))
for i in range(0, 35):
    signal.iloc[i, 0] = 0
for i in range(35, dates.size):
    signal.iloc[i, 0] = ema(macd, i, 9)
print('SIGNAL:')
print(signal)
```

Kolejnym krokiem było stworzenie wykresu MACD i SIGNAL oraz wykresu cen akcji. W tym celu posłużyłem się biblioteką matplotlib.pyplot.

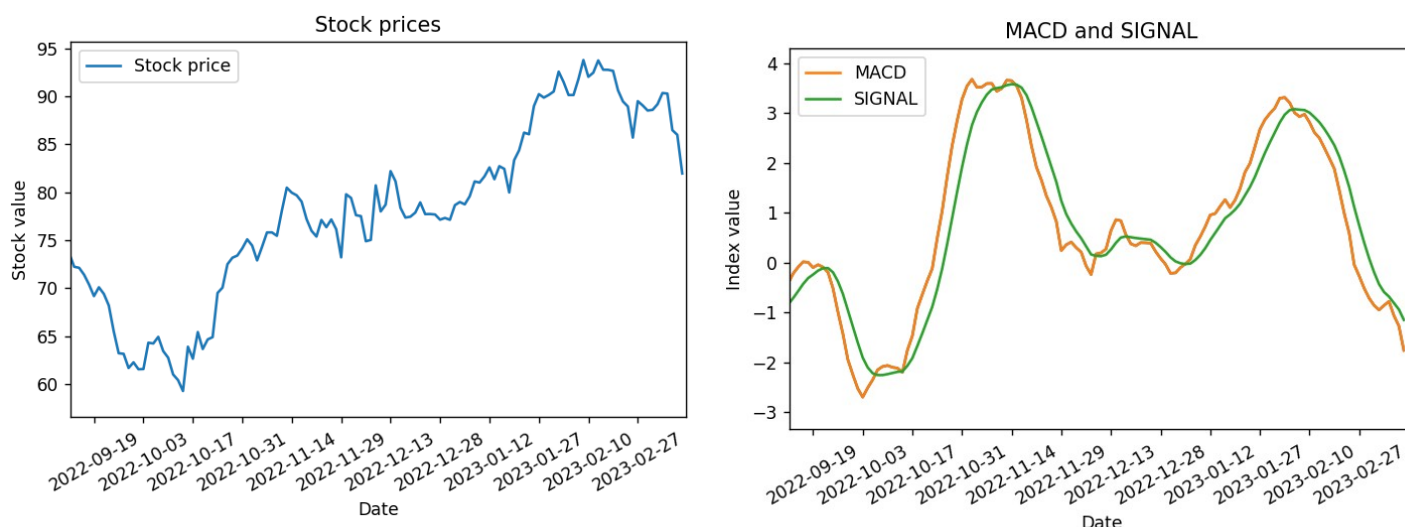
```
plt.subplots_adjust(bottom=0.15)
plt.xticks(np.arange(0, len(macd) + 1, 10))
plt.xticks(rotation=30)
plt.plot(dates, macd)
plt.plot(dates, macd, label="MACD")
plt.plot(dates, signal, label="SIGNAL")
plt.show()

plt.subplots_adjust(bottom=0.15)
plt.xticks(np.arange(0, len(prices) + 1, 10))
plt.xticks(rotation=30)
plt.plot(dates, prices)
plt.show()
```

Wykres dla całości:



Na wykresie dla 1000 dni trudno dostrzec szczegóły. Przyjrzałem się zatem części przedstawiającej ostatnie pół roku; po lewej wykres cen akcji w tym samym okresie.



Wskaźnik reaguje na najbardziej wyraźne zmiany trendów na początku października oraz lutego. W listopadzie wskaźnik gwałtownie rośnie i opada, choć cena akcji pozostaje mniej więcej taka sama – false positive, wskaźnik wykrył zmiany, których nie było. Jest to jeden z największych problemów wskaźnika MACD.

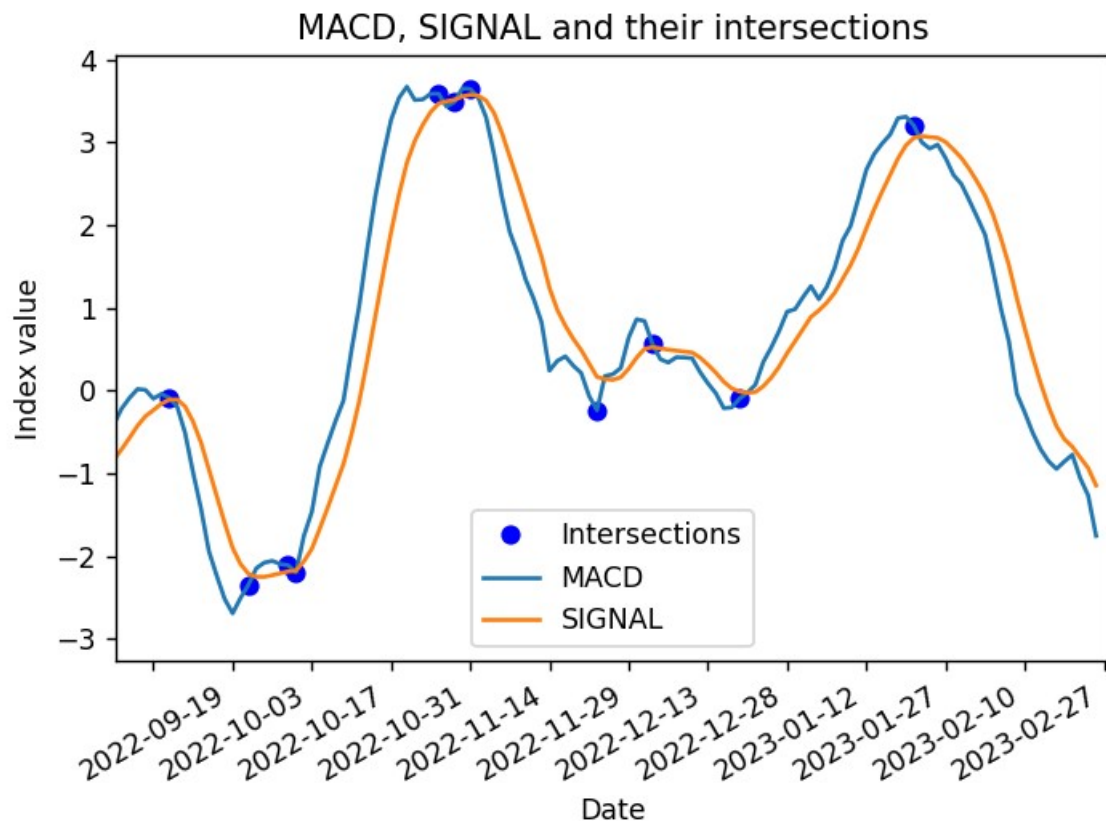
Pozostało mi stworzyć algorytm, który pozwoli mi zarobić sprzedając i kupując akcje w odpowiednich momentach. Jako startowy majątek przyjmuję 1000 dolarów.

```
money = 1000
stocks = 0
# I should buy or sell when SIGNAL intersects MACD.
# Days when I trade stocks will be marked as points
trade_points = pd.DataFrame(index=np.arange(dates.size), columns=np.arange(1))
sell_dates = pd.DataFrame(index=np.arange(dates.size), columns=np.arange(1))
buy_dates = pd.DataFrame(index=np.arange(dates.size), columns=np.arange(1))
# detect intersection by change in sign of difference
d = macd - signal
for i in range(35, dates.size-1):
    if float(d.iloc[i]) == 0. or float(d.iloc[i]) * float(d.iloc[i+1]) < 0.:
        trade_points.iloc[i] = macd.iloc[i]
        # crossover at i
        if float(d.iloc[i]) < 0:
            stocks = math.floor(money / prices.iloc[i])
            money -= stocks * prices.iloc[i]
            buy_dates.iloc[i] = i
        else:
            money += stocks * prices.iloc[i]
            stocks = 0
            sell_dates.iloc[i] = i
print('Money: ' + str(round(money, 2)))
```

Przez 1000 dni powiększyłem mój majątek 9,5-krotnie, z 1000 do 9504.34 dolarów.

Gdybym inwestował jedynie przez ostatnie pół roku, zamiast 1000 dolarów miałbym około 1400 – wciąż dobry wynik.

A oto wykres, w którym oprócz MACD/SIGNAL oznaczyłem dni, w których kupowałem/sprzedawałem akcje:

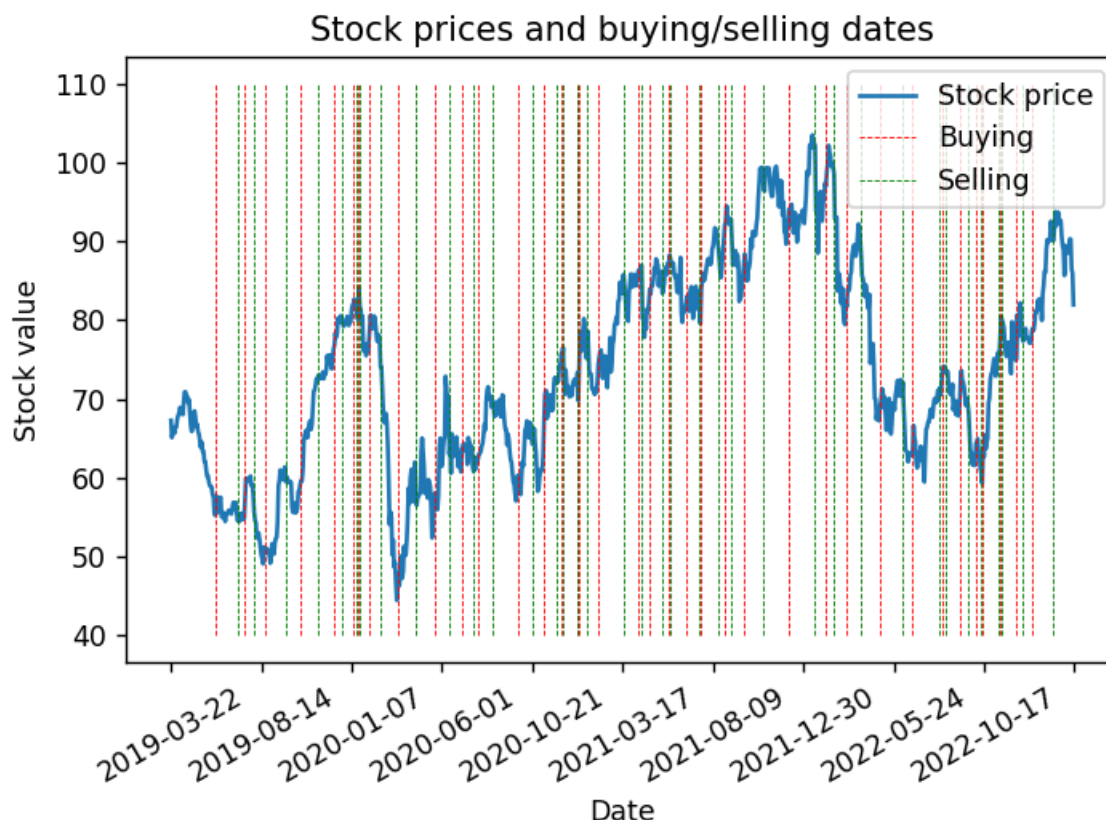


Oraz kod, którym go wygenerowałem:

```
plt.subplots_adjust(bottom=0.15)
plt.xticks(np.arange(0, len(macd) + 1, 10))
plt.xticks(rotation=30)
plt.plot(dates, trade_points, 'bo', label="Intersections")
plt.plot(dates, macd, label="MACD")
plt.plot(dates, signal, label="SIGNAL")
plt.legend(loc='best')
plt.show()
```

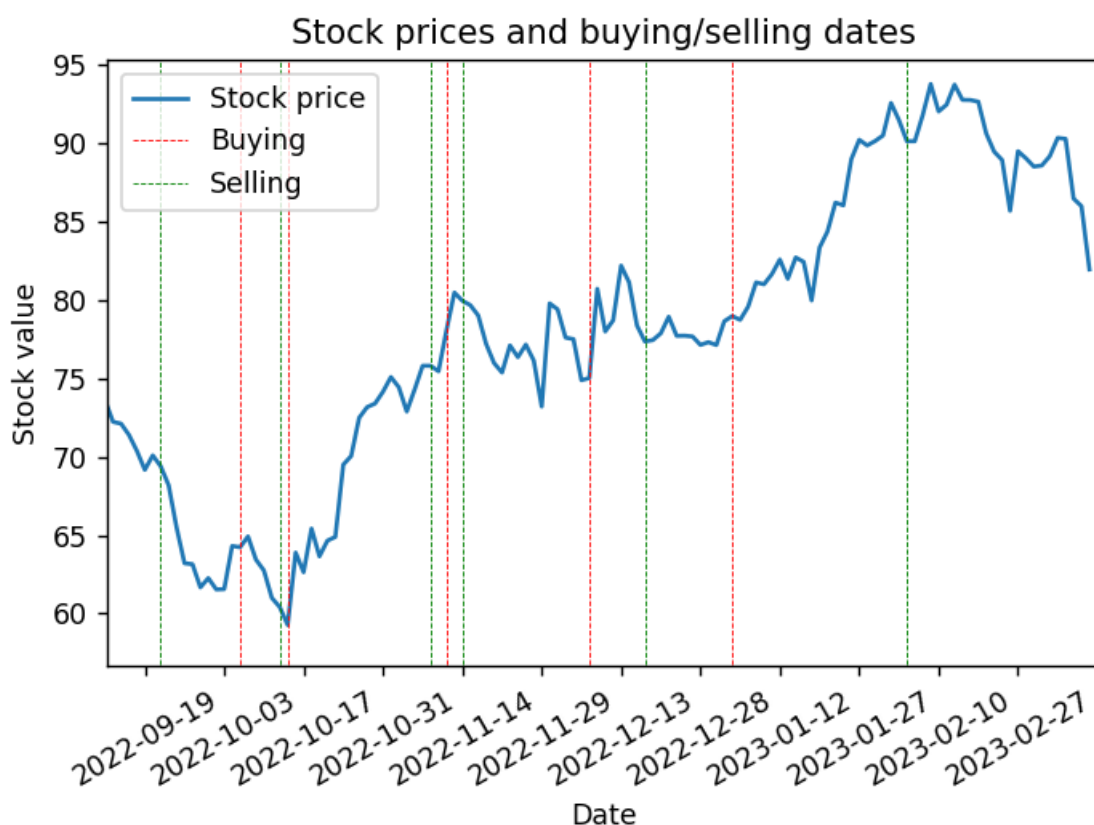
Ze względu na kwantyzację czasu punkty przecięcia są czasem wykrywane z opóźnieniem, jednak każde przecięcie jest wykryte.

Na koniec naniosłem daty kupna/sprzedaży na wykres cen akcji:

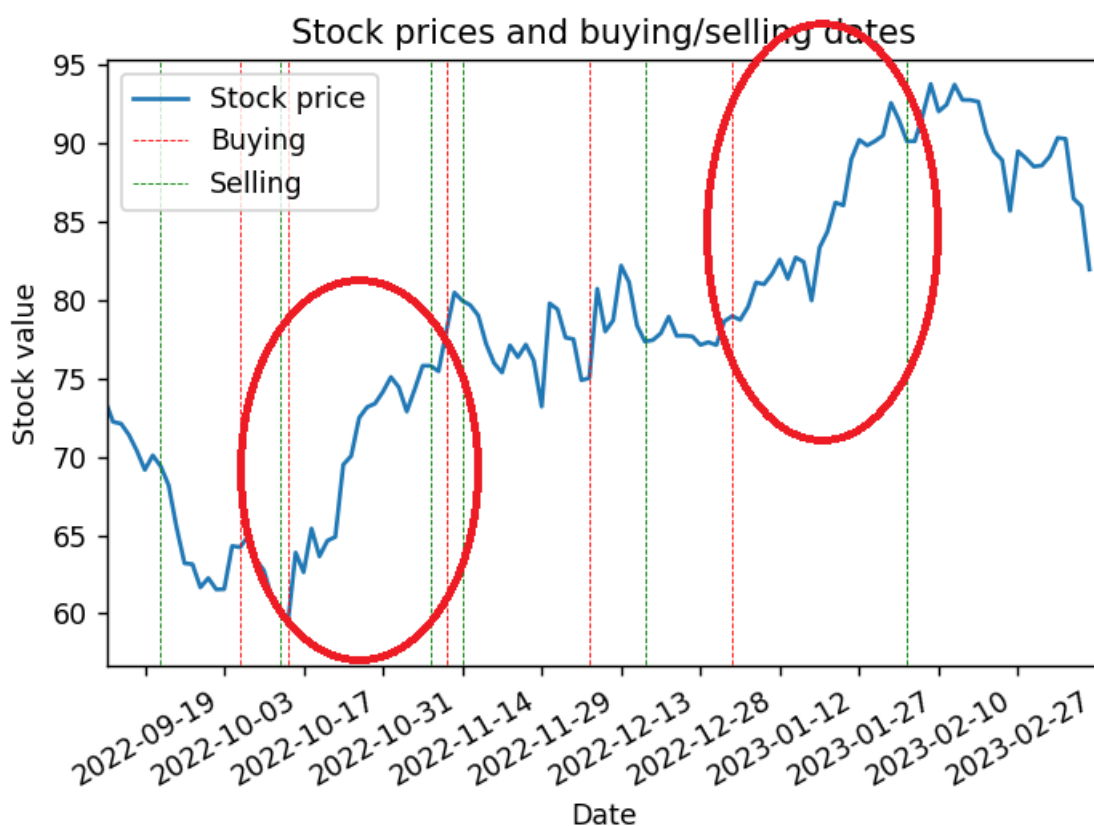


Transakcje są wykonywane bardzo często, co ma sens przy dużych fluktuacjach cen akcji.

W okresie ostatniego pół roku widać, że algorytm poradził sobie lepiej niż w symulacji, którą zacząłem 12.09.2022 – MACD jest lepszy w inwestowaniu długoterminowym przez to, że potrzebuje informacji z aż 35 poprzednich dni.

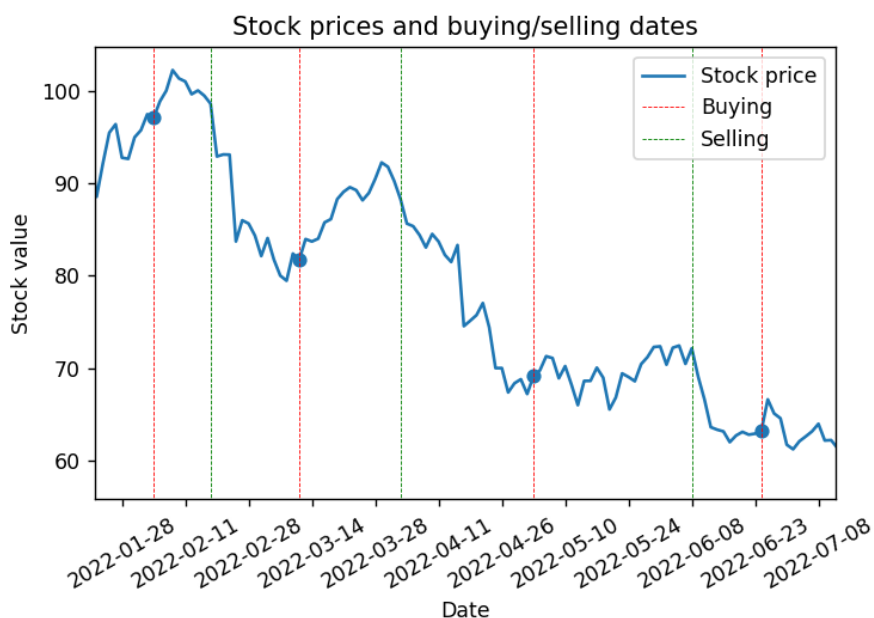


Na tym wykresie widać również, że algorytm radzi sobie świetnie z długimi okresami wzrostu cen akcji – na początku kupuje akcje, a następnie czeka z ich sprzedażą aż do odwrócenia się trendu, generując duży zysk. Okresy te zaznaczyłem poniżej:

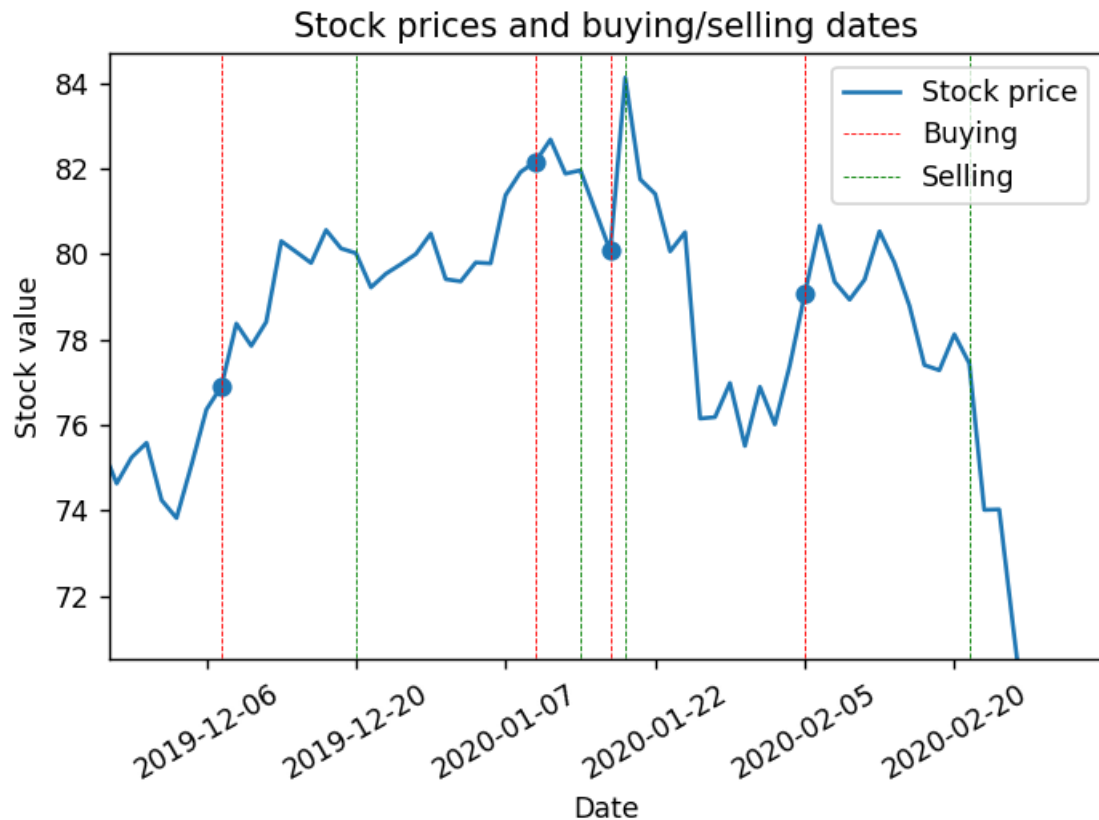


Zaraz po pierwszym zaznaczonym okresie widać również fałszywe wykrycie, o którym pisałem wcześniej – w okolicy dnia 2022-10-31 akcje są sprzedane, po chwili kupione drożej i sprzedane po tej samej cenie – wychodzi z tego niewielka strata.

Sprawdziłem w innych okresach, czy wskaźnik radzi sobie również z długimi okresami spadku cen akcji. Na wykresie poniżej widać, że w każdym okresie spadku radzi sobie znakomicie.



Na koniec postanowiłem przyjrzeć się widocznym na pełnym wykresie "Stock prices and buying/selling dates" zagęszczeniom dat kupna i sprzedaży. Czy generują zysk, czy też są transakcje występują niepotrzebnie często?



Na przybliżeniu widać, że w okresie największego zagęszczenia transakcji cena akcji zmienia się gwałtownie. Najpierw akcje sprzedane są z zyskiem, następnie kupione drożej – błąd wskaźnika. Potem jest tylko lepiej – sprzedaż, kupno, sprzedaż i duży zysk. Oznacza to, że MACD radzi sobie z gwałtownymi zmianami cen akcji i transakcje z dnia na dzień potrafią przynosić zyski.

Błędy wskaźnika nie są krytyczne, najczęściej polegają na sprzedaniu akcji nieco drożej niż zakupiono lub po tej samej cenie – strata jest niewielka.



I na koniec największa trudność/przeszkoda w projekcie:



Stack Overflow is currently offline for maintenance

Routine maintenance usually takes less than an hour. If this turns into an extended outage, we will **tweet updates from @StackStatus** or **post details on the [status blog](#)**.

[*]