

Projekt 1 Metody Numeryczne

Jakub Sachajko 179976

March 2021

1 Wskaźnik giełdowy MACD

Wskaźnik MACD czyli moving average convergence/divergence ze względu na swoją prostotę jest jednym z najpopularniejszych wskaźników analizy technicznej na świecie. MACD sprawdza się w inwestycjach długoterminowych, jednak nie nadaje się do inwestycji krótkoterminowych, ponieważ wskaźnik MACD jest często spóźniony, przez co wskaźnik ten ma problemy przy gwałtownych zmianach wartości.

MACD opiera się na EMA czyli wykładniczej średniej kroczącej. EMA to pewnego rodzaju średnia ważona która uwzględnia datę dodania danej wartości w czasie. Przymiotnik kroczący zawdzięcza dynamicznej wartości ostatnich N próbek względem aktualnych danych. EMA czyli exponential moving average wyrażona jest wzorem:

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

(EMA) 1

Zmienne:

N - Liczba okresów

$\alpha = 2/(N + 1)$

p_n - próbka z danego dnia Wskaźnik MACD składa się z dwóch linii na wykresie.

Jedną z nich jest linia MACD, a druga SIGNAL. Kiedy MACD przecina SIGNAL od dołu wtedy powinno się kupić akcje. Kiedy MACD przecina SIGNAL od góry wtedy powinno się sprzedać akcje.

2 Analiza Danych

Do wykonania projektu wczytuje 1000 wartości z pliku cdr.csv. Posłużyłem się danymi polskiej spółki CDProjekt Red. 1000 danych wejściowych pobieram z otwarcia giełdy kolejnych dni aż do 5 marca 2021 roku. Program został napisany w pythonie więc do pobierania danych wykorzystałem bibliotekę pandas. Do wizualizacji użyłem biblioteki matplotlib.

Implementacja EMA, MACD oraz SIGNAL w pythonie:

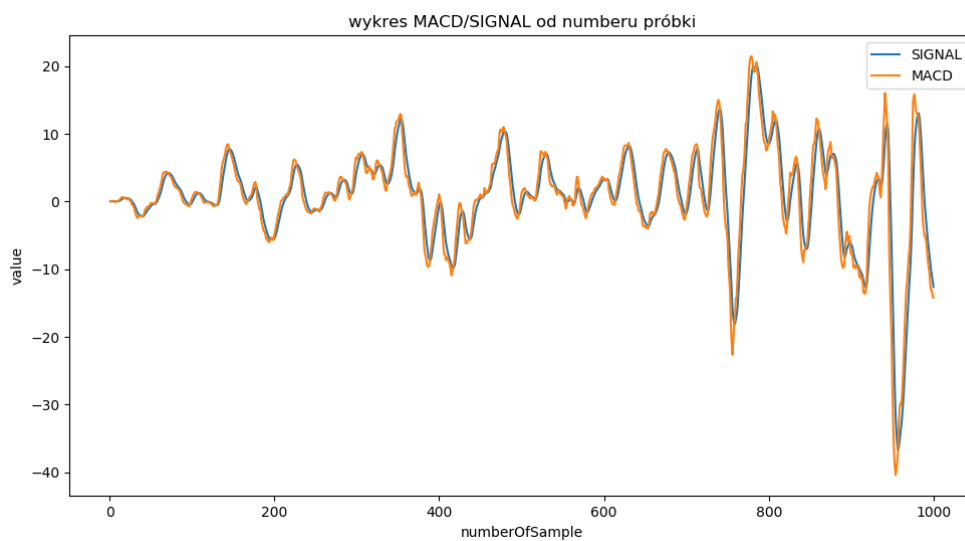
```

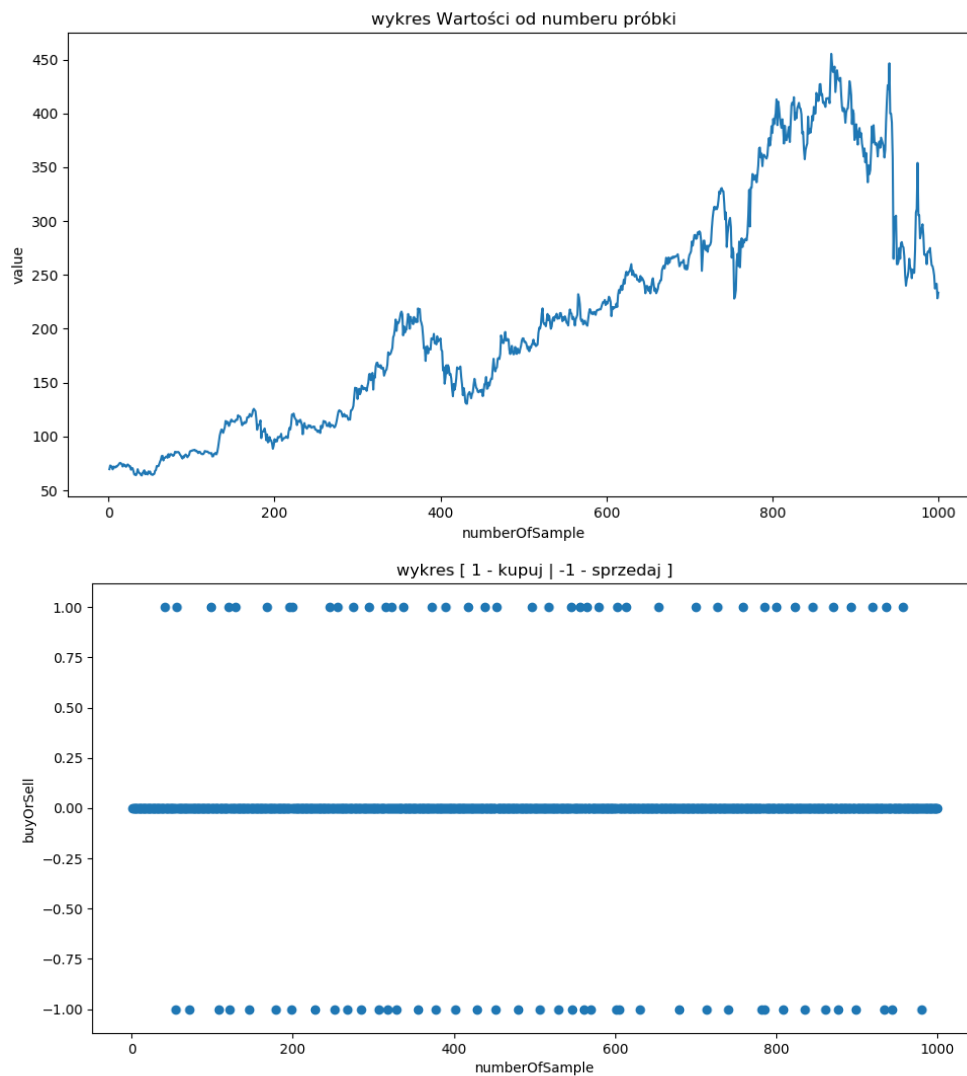
1 def EMA(numberOfEMA, numberOfSample, listOfValues):
2     EMAValue=0
3     EMABase=0
4     counterMathHelper=0
5     for i in range(numberOfSample, numberOfSample - numberOfEMA, -1):
6         if i<0:
7             break
8         EMAValue = pow(1-(2/(numberOfEMA+1)), counterMathHelper)*listOfValues[i] + EMAValue
9         EMABase = pow(1-(2/(numberOfEMA+1)), counterMathHelper) + EMABase
10        counterMathHelper += 1
11    EMAValue = EMAValue / EMABase
12    return EMAValue
13
14 def MACD(numberOfSampleMACD):
15     return EMA(12, numberOfSampleMACD, df.Otwarcie) - EMA(26, numberOfSampleMACD, df.Otwarcie)
16
17 def SIGNAL(numberOfSampleSIGNAL):
18     return EMA(9, numberOfSampleSIGNAL, MACDValues)

```

Następnie funkcje te zostały wykorzystane do policzenia MACD oraz SIGNAL dla 1000 próbek i zapisane w oddzielnych listach. Po czym zostały one umieszczone na wykresie.

Na wykresie pierwszym przedstawiona została zależność MACD od numeru próbek. Na wykresie drugim przedstawiona została zależność wartości akcji od numeru próbek. Na wykresie trzecim przedstawiona zostało kiedy według zasady przecinania się SIGNAL i MACD warto kupić/sprzedać akcje.





Pierwsze 34 pomiary mogą być niedokładne ponieważ przy liczeniu MACD potrzeba 25 próbek wstecz, a przy liczeniu SIGNAL trzeba 8 próbek MACD wstecz. Gdybyśmy chcieli przyjrzeć się działaniu programu z bliska zamieszczone zostaną dane dla ostatnich 120 próbek. Do utworzenia wykresów użyłem danej części w programie:

```

1 plt.subplot(2,2,1)
2 plt.plot(listOfThousand,SIGNALValues)
3 plt.plot(listOfThousand,MACDValues)
4 plt.title("wykres MACD/SIGNAL od numberu próbki")
5 plt.xlabel('numberOfSample')
6 plt.ylabel('value')
7 plt.legend(['SIGNAL', 'MACD'])

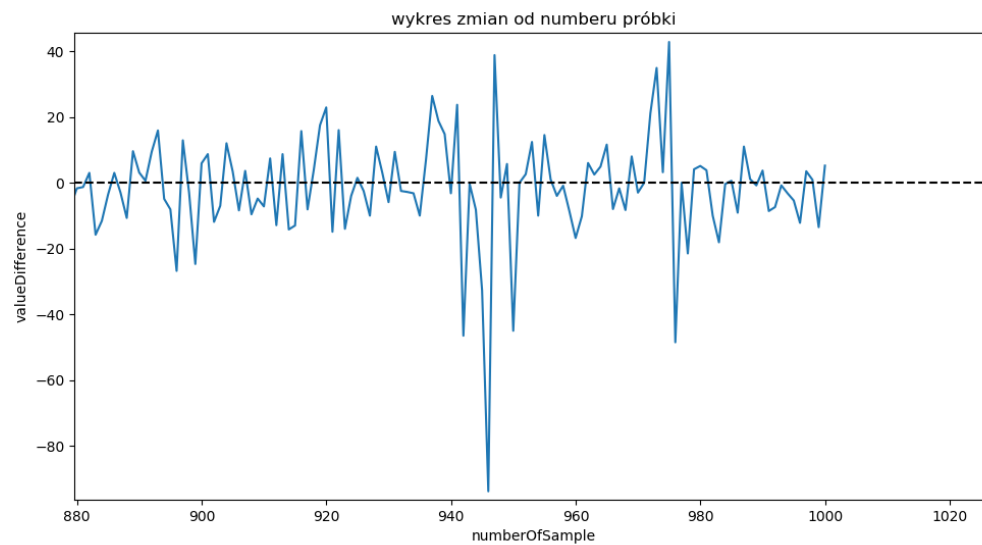
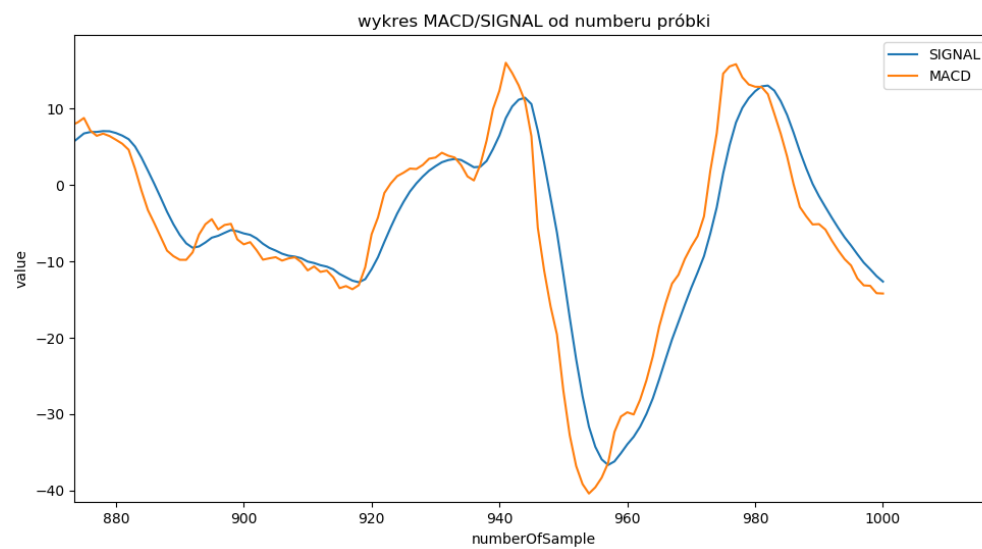
```

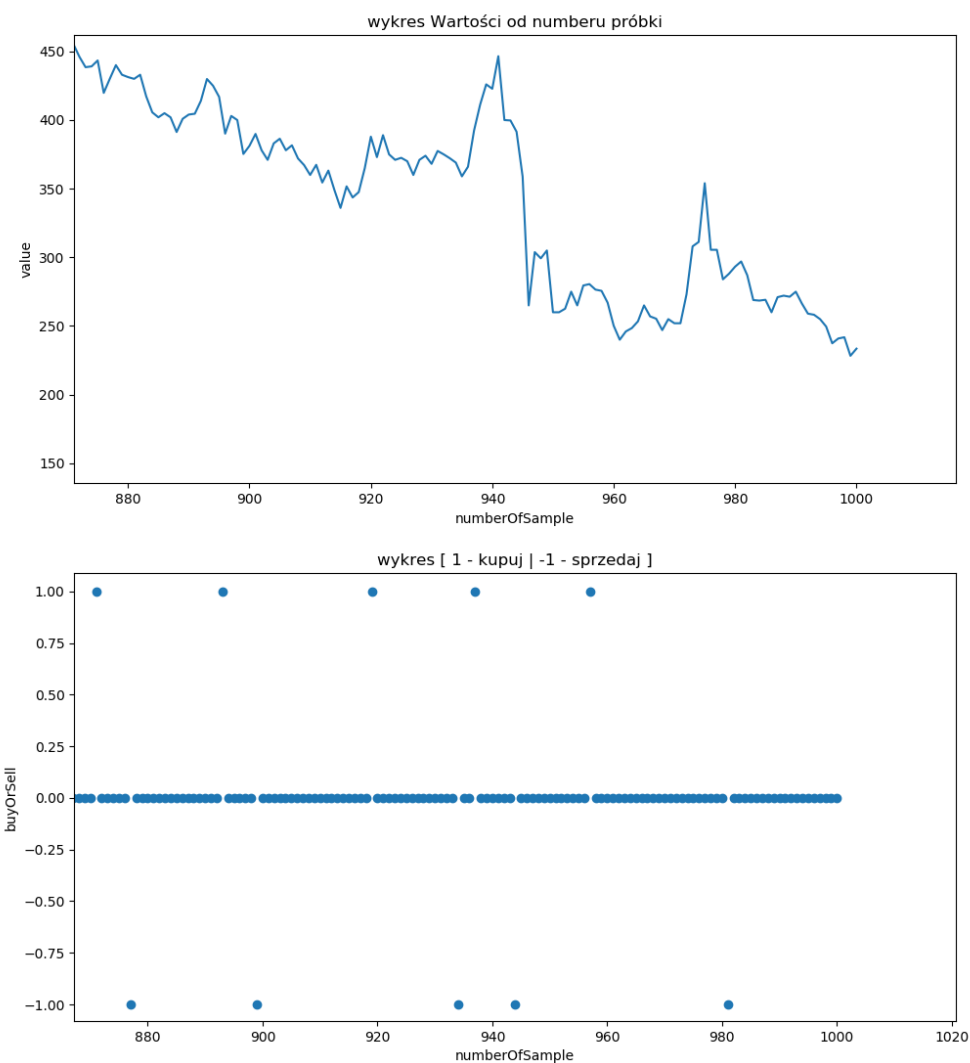
```

8 plt.subplot(2,2,2)
9 plt.title("wykres Warto ci od numeru pr bki")
10 plt.xlabel('numberOfSample')
11 plt.ylabel('value')
12 plt.plot(listOfThousand,realValues)
13 plt.subplot(2,2,3)
14 plt.title("wykres zmian od numeru pr bki")
15 plt.xlabel('numberOfSample')
16 plt.ylabel('valueDifference')
17 plt.plot(listOfThousand,checkIfGoesUpOrDown)
18 plt.axhline(y=0, color='k',linestyle='--')
19 plt.subplot(2,2,4)
20 plt.title("wykres [ 1 - kupuj | -1 - sprzedaj ]")
21 plt.xlabel('numberOfSample')
22 plt.ylabel('buyOrSell')
23 plt.axhline().remove()
24 plt.scatter(listOfThousand,shouldIBuy)
25 plt.show()

```

Na wykresie pierwszym przedstawiona została zależność MACD od numeru próbki dla ostatnich 120 pomiarów. Na wykresie drugim przedstawiona została zależność zmian wartości akcji od numeru próbki dla ostatnich 120 pomiarów. Na wykresie trzecim przedstawiona została zależność wartości akcji od numeru próbki dla ostatnich 120 pomiarów. Na wykresie czwartym przedstawiona zostało kiedy według zasady przecinania się SIGNAL i MACD warto kupić/sprzedać akcje dla ostatnich 120 pomiarów.





Z wykresów jesteśmy w stanie odczytać, że w skali 1000 pomiarów wskaźnik MACD jest lekko spóźniony w porównaniu do rzeczywistej wartości, mimo to możemy zauważyć, że wykres ten na pierwszy rzut oka odwzorowuje zmiany. Jednak zmiany te odnotowywane są z opóźnieniem i niepełną dokładnością. Gdy przyjrzymy się wykresom dla ostatnich 120 pomiarów gwałtowny spadek odnotowany jest z opóźnieniem co sprawia, że długoterminowe akcje prawdopodobnie przyniosą zysk, a krótkotrwałe inwestycje nie są zalecane.

3 Algorytm

Po przeanalizowaniu wskaźnika MACD jestem w stanie zaproponować algorytm, który na podstawie przecięć linii SIGNAL i MACD podejmie decyzje o kupnie i sprzedaży. Wcześniej użyty wykres [1 - kupuj | -1 - sprzedaj] ukazuje, w których momentach algorytm ten będzie kupował i sprzedawał akcje. Program inwestujący zacznie z 1000 jednostek waluty początkowej. Kod do podejmowania decyzji o transakcjach:

```
1 kapital = 1000
2 liczbaAkcji = 0
3
4 print("kapita początkowy wynosi :",kapital)
5 print("liczba akcji wynosi :",liczbaAkcji)
6 print("-----")
7
8 listOfThousand = list(range(1,1001))
9 for i in range(1000):
10     MACDValues.append(MACD(i))
11     SIGNALValues.append(SIGNAL(i))
12     realValues.append(df.Otwarcie[i])
13     if i==0:
14         checkIfGoesUpOrDown.append(0)
15         shouldIBuy.append(0)
16     else:
17         checkIfGoesUpOrDown.append(realValues[i]-realValues[i-1])
18         if (i>=33):
19             if MACDValues[i] < SIGNALValues[i] and MACDValues[i-1] > SIGNALValues[i-1]:
20                 shouldIBuy.append(-1)
21                 kapital = kapital + (liczbaAkcji * realValues[i])
22                 liczbaAkcji = 0
23             elif MACDValues[i] > SIGNALValues[i] and MACDValues[i-1] < SIGNALValues[i-1]:
24                 shouldIBuy.append(1)
25                 liczbaAkcji = liczbaAkcji + (kapital//realValues[i])
26                 kapital = kapital - (liczbaAkcji * realValues[i])
27             else:
28                 shouldIBuy.append(0)
29         else:
30             shouldIBuy.append(0)
31
32     print("Nr:",i+1,"posiada akcji:",liczbaAkcji,"posiada kapita :",kapital,"warto akcji:",
33           realValues[i])
34
35 if liczbaAkcji != 0:
36     kapital += liczbaAkcji*realValues[999]
37     liczbaAkcji = 0
38
39 print("-----")
40 print("kapita końcowy wynosi :",kapital)
41 print("liczba akcji wynosi :",liczbaAkcji)
```

Po uruchomieniu programu program zakończył z zyskiem i wynosił 3143.58 więc zysk wyniósł 2143.58 co ukazuje, że długoterminowe inwestycje mogą być opłacalne. Jednak akcje firmy CD Projekt Red wzrosły w najlepszym wypadku około 350 jednostek więc nie jest to program, który w 100 procentach wykorzystuje potencjał zysku. Gdyby akcje traciły na wartości program i tak traciłby pieniądze.

4 Wnioski

Przydatność Wskaźnika MACD do pewnego stopnia jest widoczna gdy posiada się dużo danych, i długoterminowe inwestycje. W przypadku Gdy akcje traca na wartości strata pieniężna jest w sposób minimalizowana. Jednak gdy liczy się na krótkoterminowy zysk, wtedy nie jest to dobra opcja ponieważ możliwe gwałtowne wzrosty i spadki cen akcji są w stanie wyrządzić szkody kapitałowi inwestora. Jednak jestem przekonany, że programy wykorzystujące sztuczną inteligencję są powszechne, więc z tego sprawozdania, wynika że wskaźnik MACD nie jest przydatny w obligacja giełdowych.