

Projekt I Metody Numeryczne

Wskaźnik giełdowy MACD

Wykonał: Piotr Kopecki nr indeksu 184746

Projekt wykonałem w języku Python, do obsługi plików csv użyłem biblioteki csv, do obsługi wykresów wykorzystałem matplotlib. Jako dane wejściowe wykorzystałem plik csv zawierający historię kursu akcji firmy Disney w przedziale czasowym: 2016-04-12 -- 2020-03-31 (1000 dni).

Pracę nad projektem rozpocząłem od obsługi pliku csv, z którego potrzebowałem pobrać wartości kursu w danych dniach. Fragment kodu wykonujący te czynności:

```
class CsvReader:
    data = []
    date = []

    def get_data(self):
        with open('DIS.csv', 'r') as file:
            reader = csv.reader(file, delimiter=',')
            for row in reader:
                self.data.append(float(row[3]))
        return self.data

    def get_date(self):
        with open('DIS.csv', 'r') as file:
            reader = csv.reader(file, delimiter=',')
            i = 0
            for row in reader:
                self.date.append(i)
                i += 1
            return self.date
```

Następnie postanowiłem zająć się obsługą wykresów. Moim celem było wyświetlenie w jednym oknie dwóch wykresów, jeden przedstawiający jak zmieniał się kurs na przestrzeni 1000 dni, drugi natomiast przedstawiający wartości MACD/SIGNAL. Finalnie do drugiego wykresu dodałem również wyświetlanie liter B od „Buy” i S od „Sell” w momentach, w których mój algorytm dokonuje zakupu/sprzedaży akcji. Zanim zrozumiałem i zaimplementowałem działanie MACD do wyświetlenia drugiego wykresu posługiwałem się sztucznie utworzonymi danymi.

Funkcje te znajdują się w klasie Graph

Funkcja obsługująca rysowanie wykresu kursu:

```
def draw_exchange(self, x, y):  
    plt.plot(x, y, label="Kurs DIS")  
    plt.xlabel("Data")  
    plt.ylabel("Kurs w danym dniu")  
    plt.legend()  
    plt.grid(True)
```

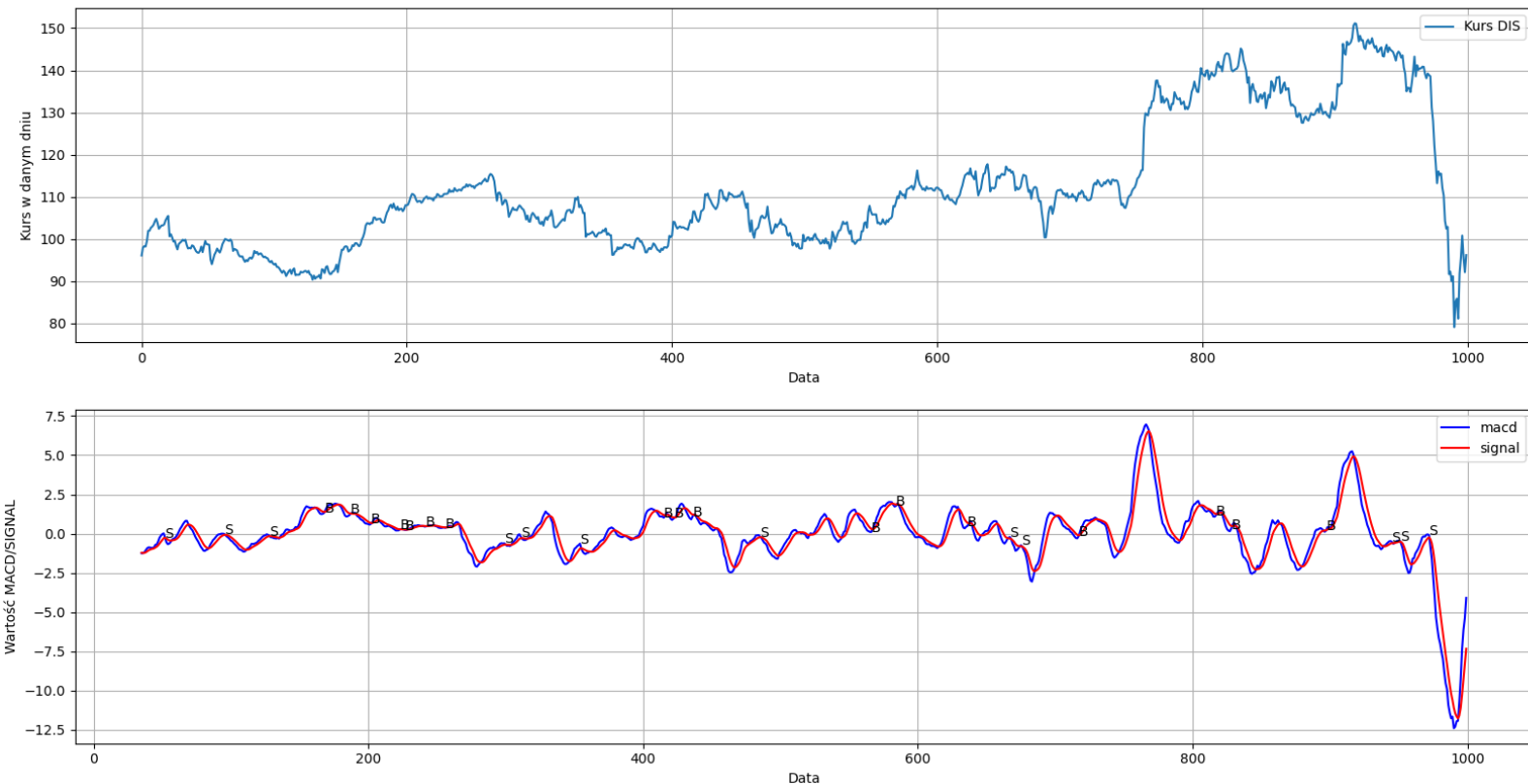
Funkcja obsługująca rysowanie wykresu MACD/SIGNAL:

```
def draw_macd_signal(self, x, macd, signal, buy_or_sell, with_or_without_letter):  
    plt.plot(x, macd, label="macd", color='blue')  
    plt.plot(x, signal, label="signal", color='red')  
  
    if(with_or_without_letter):  
        for i in range(1, len(macd) - 1):  
            if buy_or_sell[i] != "-":  
                plt.annotate(buy_or_sell[i], xy=(x[i], macd[i]))  
  
    plt.legend()  
    plt.grid(True)  
    plt.ylabel('Wartość MACD/SIGNAL')  
    plt.xlabel('Data')
```

Funkcja rysująca oba te wykresy w jednym:

```
def draw_exchange_macd_signal(self, x, y, macd, signal, buy_or_sell, with_or_without_letter):  
    plt.figure('Piotr Kopecki 184746')  
    plt.subplot(2, 1, 1)  
    self.draw_exchange(x, y)  
    plt.subplot(2, 1, 2)  
    self.draw_macd_signal(x[35:], macd, signal, buy_or_sell, with_or_without_letter)  
    plt.savefig("Caly_wykres.png")  
    plt.show()
```

Finalny wygląd wykresu:



Następnie rozpocząłem pracę nad implementacją MACD. Do jego zrozumienia przysłużyła mi się strona www.fxmag.pl oraz wzór podany w instrukcji do projektu.

Funkcje przedstawione poniżej znajdują się w klasie Macd

Funkcja obliczająca ema, jako parametry przyjmuje **n** (26 bądź 12), **dane** data oraz **dni** date

```
def calc_ema(self, n, data, date):
    alfa = 2 / (n + 1)
    counter = 0.0
    denominator = 0.0
    d = self.get_reversed_data_from_to_from(data, n, date)
    for i in range(n + 1):
        if i == 0:
            counter += d[i]
            denominator += 1
        else:
            counter += ((1 - alfa) ** i) * d[i]
            denominator += (1 - alfa) ** i
    return counter / denominator
```

Mając sprawnie działającą funkcję wyliczającą ema mogłem wyliczyć MACD oraz SIGNAL.

```
def calc_macd(self, data):
    self.macd = []
    for i in range(len(data)):
        if i >= 26:
            ema_26 = self.calc_ema(26, data, i)
            ema_12 = self.calc_ema(12, data, i)
            self.macd.append(ema_12 - ema_26)
        else:
            self.macd.append(0.0)

    def calc_signal(self):
        self.signal = []
        for i in range(35, len(self.macd)):
            self.signal.append(self.calc_ema(9, self.macd, i))
```

Następnie zająłem się implementacją algorytmu który dla początkowego kapitału równego w moim przypadku 1000 * cena startowa jednostki akcji oraz dla ustalonej wartości procentu aktualnego kapitału, który ma być inwestowany w przypadku zakupu (przyjąłem 5 %) ma wykonać symulacje inwestycji w danym okresie i odpowiedzieć mi na pytanie, czy warto byłoby w ten sposób inwestować.

Wyliczenie momentów dobrych do zakupu/sprzedaży akcji

```
def calc_buy_or_sell(self):
    self.buy_or_sell = []
    tempM = self.get_macd()
    tempS = self.get_signal()
    for i in range(1, len(tempM)):
        if tempM[i] > tempS[i] and tempM[i-1] < tempS[i-1] and tempM[i] > 0:
            self.buy_or_sell.append("B")
        elif tempM[i] < tempS[i] and tempM[i-1] > tempS[i-1] and tempM[i] < 0:
            self.buy_or_sell.append("S")
        else:
            self.buy_or_sell.append("-")
```

Kupuję akcję za 5 % aktualnego kapitału w przypadku, gdy wykres MACD przecina SIGNAL od dołu oraz jest większy od 0, sprzedaje, gdy przecina od góry oraz jest mniejszy od 0.

Symulacja inwestycji jest zaimplementowana w klasie InvestingSimulator

```
def buy(self, prize):
    money_in = self.current_capital * self.percent_of_capital_to_invest
    stock_for_money_in = money_in / prize
    self.current_capital -= money_in
    self.how_much_stock_owned += stock_for_money_in

def sell(self, prize):
    self.current_capital += prize * self.how_much_stock_owned
    self.how_much_stock_owned = 0.0

def invest(self, buy_or_sell, data):
    for i in range(1, len(buy_or_sell)):
        if(buy_or_sell[i] == "B"):
            self.buy(data[i])
        elif(buy_or_sell[i] == "S"):
            self.sell(data[i])
    if(self.how_much_stock_owned > 0.0):
        self.sell(data[len(data) - 1])
```

Ostatecznie otrzymuje:

```
C:\Users\piotr\Documents\ProjektNomer\yozne\1\venv\Scripts\python.exe 0.70
Starting money equals 1000 * starting action prize
Starting money: 96070.0
Money after investing: 105238.77
Capital after investing is worth 109.54 % of starting capital
```

Jak widać inwestycja przebiegła pomyślnie, zysk wyniósł prawie 10 % zainwestowanego kapitału.

Wnioski:

Podsumowując uważam, iż wskaźnik MACD może być bardzo użyteczny w analizie technicznej. Nie jest to nieomylna wyrocznia, co udowodnił kiedy podczas badania różnych algorytmów otrzymywałem wielokrotnie mniejszy kapitał od startowego, jednak warto go znać i stosować razem z innymi wskaźnikami np. EPS, Williams %R.