

Srednja škola Zvane Črnje

Carduccijska 20

52210 Rovinj

Tema:

AI bot za trgovanje kriptovalutama

Mentor:

Lovro Šverko, prof. mentor

Učenik:

Gabriel Matošević, 4,d

Smjer: tehničar za računalstvo

Rovinj, ljetni rok

šk. god. 2021./2022

Sadržaj	
PREDGOVOR	2
1 UVOD	3
1.1 Blockchain tehnologija	3
1.2 Kriptovalute	4
1.3 Trgovanje	4
2 TRGOVANJE KRIPTOVALUTAMA	5
2.1 Razlike trgovanja kriptovalutama i drugim imovinama	5
2.2 Predviđanje porasti ili pada vrijednosti kriptovaluta	5
2.3 Korištenje programskih jezika u svrhu analize	6
2.4 Najkorišteniji programski jezik za analizu kriptovaluta	6
2.5 Baze podataka i njihova uporaba u analizi podataka	7
2.6 Programi za analizu cijena kriptovaluta	8
3 PRINCIP RADA	10
3.1 Korišteni materijali i alati	10
3.1.1 Visual studio 2022	10
3.1.2 Biblioteke programa	10
3.2 Opis funkcionalnosti	12
3.3 Opis koda	13
3.3.1 Opis funkcije BBsMomentumStrategy, funkcija za analizu podataka te kupnju i prodaju kriptovaluta.	13
3.3.2 Opis koda baze podataka	20
4 ZAKLJUČAK	25
5 LITERATURA	26
6 PRILOZI	27
6.1 Popis slika	27
6.2 Popis tablica	27

PREDGOVOR

Izjavljujem pod punom odgovornošću da sam završni rad izradio samostalno, isključivo znanjem stečenim tijekom srednjoškolskog obrazovanja, služeći se navedenim izvorima podataka i uz stručno vodstvo mentora Lovra Šverka, prof., kojemu se još jednom srdačno zahvaljujem. Također se zahvaljujem svojim kolegama i ostalim profesorima na pomoći.

1 UVOD

Program za trgovanje kriptovalutama obuhvaća analizu cijene kriptovalute, izračunate rizike, kupnju i prodaju te mogućnost oblikovanja programa po uputama korisnika.

Takav program sastoji se od strategije za analizu cijene, baze podataka koja sadržava cijene koje će biti analizirane te opcionalnog grafičkog dijela programa za graf cijena.

1.1 Blockchain tehnologija

Blockchain tehnologija je decentralizirani sustav gdje se izmjenjuju informacije. Blockchain se sastoji od blokova koji sadrže informaciju i digitalni potpis koji potvrđuje jedinstvenost te informacije. Svaki digitalni potpis sastoji se od hash vrijednosti koja se dobiva tako da se koristi računalna snaga u smislu brute force dešifriranja određenog broja koji određuje specifični algoritam.

Putem blockchain-ova mogu se napraviti sustavi prijenosa informacija pri kojima osoba koja koristi računalnu snagu bude nagrađena za odrađen rad. Putem tog broja potvrđuje se prenešena informacija.

Za potvrdu informacija koriste se funkcije koje nemaju inverznu funkciju poput SHA-256 te tako osiguravamo blockchain to jest nitko ga ne može promijeniti.

1.2 Kriptovalute

Postoje blockchain-ovi poput Bitcoina koji omogućuju korištenje računalne snage kako bi mogli dešifrirati bitcoin-ov algoritam te nas na taj način blockchain nagradi s kriptovalutom poput Bitcoin-a koja je slična zlatu ili novcu. Taj proces se također može zvati majnanje (eng. Mining).

Kriptovalutama se može trgovati kao dionicama na tržištu kupaca i prodavača. Jedna od najpoznatijih kripto mjenjačnica jest Binance.

Kao i dionice, kriptovalute mogu rasti i padati, ali za razliku od dionica kriptovalute su jako nestabilne te mogu imati promjene u cijenama mnogo veće od promjena cijena dionica te se smatraju jako rizičnim.

1.3 Trgovanje

Trgovanje je postupak projektiranja i predviđanja promjena cijena imovina te stvaranje long ili short pozicija nad njima kako bi došli do profita. Long pozicija je pozicija u trgovanju gdje osoba kupuje imovinu te ju prodaje u daljnjem roku očekujući da će vrijednost te imovine narasti. Short pozicija je pozicija gdje osoba prodaje tuđu imovinu očekujući da će vrijednost te imovine pasti.

2 TRGOVANJE KRIPTOVALUTAMA

2.1 Razlike trgovanja kriptovalutama i drugim imovinama

Kriptovalute su napravljene kako bi ljudi mogli stvarati transakcije online bez potrebe fizičke pomoći dok dionice, forex i slično zahtijevaju osoblje koje pregledava skupne podatke transakcija pomoću alata, algoritma i slično. Kriptovalute se mogu trgovati 24 sata svaki dan dok su npr. forex i dionice dostupne za trgovanje samo kada su tržišta otvorena.

2.2 Predviđanje porasti ili pada vrijednosti kriptovaluta

Ako ste ikada gledali vijesti mogli ste vidjeti kako reporteri pričaju o budućem padu ili porastu cijene popularne i prve kriptovalute Bitcoin. Ponekad te vijesti mogu biti istinite, ali često nisu pošto kriptovalute jako nepredvidljive te je svaka pretpostavka o porastu ili padu cijene samo to, pretpostavka.

Cijena kriptovaluta ne može se sa sto postotnom sigurnošću predvidjeti te zato je trgovanje kriptovalutama jako rizično.

Postoje dvije vrste analize cijena kriptovaluta:

- Prva vrsta analize je tehnička analiza pomoću alata poput pokretnih prosjeka (pokretni prosjek uzima cijenu valute u prošlih n dana (n je broj koji mi biramo) te nam vraća prosječnu vrijednost te valute) ili pomoću standardne devijacije cijene.
- Druga vrsta analize je putem događaja / promjena u svijetu, jedan primjer koji bi mogao pomoći porastu cijene kriptovaluta je da jedna država u svijetu prihvati kriptovalute kao način izmjene plaćanja a suprotni primjer bi bio da jedna firma odustane od mogućnosti plaćanja s kriptovalutama.

2.3 Korištenje programskih jezika u svrhu analize

Mnogo mjenjačnica kriptovaluta imaju API¹, on služi kao veza između naših zahtjeva i izvora to jest mjenjačnica kriptovaluta koje sadrže trenutne cijene kriptovaluta.

Putem API-eva možemo:

- Zahtijevati trenutnu i prošlu cijenu neke kriptovalute
- Stvarati narudžbe
- Otvarati long i short pozicije
- Itd.

Putem programa koji stvorimo možemo raditi sve što inače možemo raditi na tim mjenjačnicama, ali brže.

2.4 Najkorišteniji programski jezik za analizu kriptovaluta

U svijetu postoji mnogo programskih jezika koji su napravljeni za specifične svrhe, neki su brži no teži za koristiti, a neki su sporiji no lakši za koristiti itd.

Za analizu cijene kriptovaluta, stvaranje narudžba i otvaranje long i short pozicija kod promjena cijena u periodu do dvije minute nije potrebno mnogo računalne snage te nije potrebno imati brz programski jezik osim ako ne stvaramo program koji analizira podatke te otvara long ili short pozicije za skoro svaku promjenu cijene.

Većina programa koji analiziraju cijenu kriptovaluta su napravljeni u programskom jeziku python.

Python je programski jezik koji je brz, lagan za korištenje i učinkovit te je zato najviše preporučen za uporabu kod programa koji analiziraju cijene kriptovaluta.

¹ Eng. Application Programming Interface

2.5 Baze podataka i njihova uporaba u analizi podataka

Baza podataka je skup podataka spremljen u tablicu. Tablica može imati retke i stupce pri čemu nam stupci govore vrstu podataka to jest njihov simbol, a redak je sami podatak.

Putem baze podataka možemo spremati podatke poput cijene kriptovalute u određenom vremenu. Također možemo mijenjati te podatke, čitati ih, brisati i dodavati nove podatke u tablicu.

Baze podataka su ključne za analizu cijene kriptovalute zbog njihove jednostavnosti i efikasnosti.

Na sljedećoj slici „Slika 1“ je primjer podataka u bazi podataka sa 3 stupca, symbol (Označava kriptovalutu), DateTime (Označava datum u epoch vremenskom formatu), Price (Označava cijenu određene kriptovalute u američkim dolarima)

	symbol	DateTime	Price
	Filter	Filter	Filter
1	BTCUSDT	1650299915020	39292.06
2	BTCUSDT	1650299915048	39292.05
3	BTCUSDT	1650299915659	39292.06
4	BTCUSDT	1650299916081	39292.05
5	BTCUSDT	1650299916105	39292.06
6	BTCUSDT	1650299916244	39292.06
7	BTCUSDT	1650299916974	39292.06
8	BTCUSDT	1650299917044	39292.06
9	BTCUSDT	1650299917267	39292.06
10	BTCUSDT	1650299917595	39292.05
11	BTCUSDT	1650299917659	39292.06
12	BTCUSDT	1650299917709	39292.05
13	BTCUSDT	1650299918119	39292.06
14	BTCUSDT	1650299918233	39292.05
15	BTCUSDT	1650299918237	39292.05
16	BTCUSDT	1650299918241	39291.97
17	BTCUSDT	1650299918244	39289.01
18	BTCUSDT	1650299918246	39286.84
19	BTCUSDT	1650299918254	39286.8
20	BTCUSDT	1650299918258	39286.72

Slika 1 Primjer baze podataka

2.6 Programi za analizu cijena kriptovaluta

Putem programskog jezika poput pythona može se uzeti trenutna i prošla cijena jedne kriptovalute te se ta vrijednost može spremiti u bazu podataka.

Kada bi spremili cijenu kriptovalute u bazu podataka te skupili dovoljno cijena (ovisno o periodu analize što može biti jedna sekunda, jedna minuta, tjedan dana ili duže/kraće) možemo započeti sa analizom.

Putem tehničke analize sa nakupljenim podacima možemo stvoriti long ili short poziciju ovisno o projekciji analize.

Rizik ovakvih programa je sama nepredvidljivost kriptovaluta. U jednom programu je skoro pa nemoguće stvoriti projekciju cijene koja je sto posto istinita.

Pozitivna strana ovakvih programa je mogućnost stvaranja programa koji bi analizirali cijene kriptovaluta kada sama osoba koja trguje kriptovalutama nema vremena.

Loša strana ovakvih programa je mogućnost loše analize cijene kriptovaluta i stvaranje pozicija koji gube novac.

3 PRINCIP RADA

3.1 Korišteni materijali i alati

3.1.1 Visual studio 2022

Visual Studio Community je besplatan, potpuno opremljen IDE² za studente, suradnike otvorenog koda i individualne programere.

3.1.2 Biblioteke programa

Slijedeća tablica opisuje sve korištene biblioteke u programu za analizu, bazu podataka i stvaranje grafa.

Tablica 1 Biblioteke programa

Biblioteka	Opis
subprocess	subprocess omogućuje vam da pokrenete nove procese, povezati se na njihove ulazne/izlazne/error cijevi i dobijete njihove povratne kodove.
pandas	pandas je popularni alat za analizu podataka baziran na python-u. Predstavlja raznolik raspon uslužnih programa, u rasponu od raščlanjivanja više formata datoteka do pretvaranja cijele tablice podataka u NumPy matrični niz. To čini pandas pouzdanim saveznikom u znanosti o podacima i strojnom učenju.
sqlalchemy	sqlalchemy je sveobuhvatan skup alata za rad s bazama podataka.
matplotlib	matplotlib je zbirka funkcija te svaka funkcija pyplot ili animation čini neke promjene slici: npr. stvara figuru, stvara područje za crtanje

² Eng. Integrated Development Environment

	na slici, iscrtava neke linije u području crtanja, ukrašava graf s oznakama, stvara graf koji konstantno mijenja svoje vrijednosti, itd.
time	time je alat koji nam omogućuje rad sa vremenom, npr. mjerač vremena, funkcija <code>time.sleep()</code> koja nam omogućuje pauziranje programa određeni broj vremena.
datetime	datetime je alat koji nam omogućuje rad sa datumima i vremenom, razlikuje se sa time modulom tako da datetime se koristi samo za datume dokle time module se koristi samo za vrijeme. Pomoću ovog modula možemo mijenjati datume, stvarati nove itd.
numpy	numpy se koristi za znanstveno računanje i rad s nizovima. Osim svog višedimenzionalnog niza objekata, također nudi funkcionalne alate visoke razine za rad s nizovima.
asyncio	asyncio se koristi kao temelje za mnogo Python asinkronih okvira koje pruže mreže i web poslužitelje visoke kvalitete, biblioteke za spajanje baze podataka, itd.
os	os se koristi kako bi mogli stvoriti komande na operativnom sustavu u Pythonu, poput brisanja podataka, stvaranja podataka, itd.
sqlite3	sqlite3 je alat za modificiranje, brisanje i stvaranje baze podataka.
binance	binance je API od kripto mjenjačnice binance, sa binance API možemo stvarati narudžbe, hvatati cijene kriptovaluta, itd.

3.2 Opis funkcionalnosti

Moj program koristi programski jezik python za analizu podataka, stvaranje pozicija i izvršavanje naredbi.

Prvo stvara bazu podataka cijene kriptovalute Bitcoin koja ima 3 stupca (simbol, cijena, datum/vrijeme) i 3600 retka. Zatim briše stare podatke kako bi ostalo 3600 retka i kako ne bi zauzimalo previše memorije.

Uzeo sam 3600 redaka kako bi moj program analizirao podatke koji su stari otprilike 12 minuta. Binance API koji koristim šalje podatak cijene Bitcoin-a otprilike svakih 0,2 milisekunde, a pomnožimo li 0,2 milisekunde sa 3600 retka, dobijemo 12 minuta.

Te podatke analiziram na dva načina: kratkotrajno i dugotrajno.

Kratkotrajno: Tražim razliku trenutne cijene sa cijenom prije tri minute, ako je cijena narasla za 0.15% onda je kratkotrajni uvjet zadovoljen.

Dugotrajno: Uzimam sve cijene u posljednjih 12 minuta te uzimam standardnu devijaciju tih cijena. Ako je trenutna cijena ispod prosječne cijene umanjene za standardnu devijaciju, onda je dugotrajni uvjet zadovoljen.

Sljedeća slika „Slika 2“ prikazuje kako program ostvaruje dobit od oko 0.2% (lijevo) i grafički prikaz kako se ta dobit ostvarila (desno)



Slika 2 Primjer narudžbe i zadovoljenih uvjeta

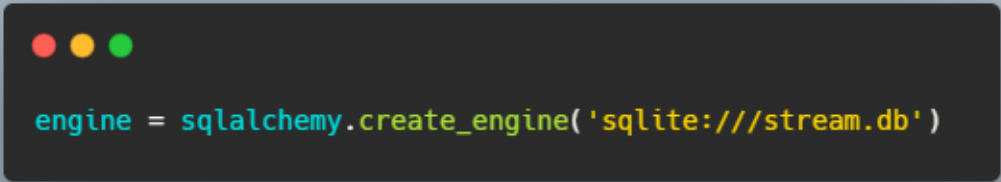
3.3 Opis koda

3.3.1 Opis funkcije BBSMomentumStrategy, funkcija za analizu podataka te kupnju i prodaju kriptovaluta.

Sljedeća slika, „Slika 3“ prikazuje funkciju BBSMomentumStrategy

```
def BBSMomentumStrategy(lookback, win, loss, qty, open_position=False):
    engine = sqlalchemy.create_engine('sqlite:///stream.db')
    cutloss = qty*0.98
    while True:
        if qty < cutloss:
            print("\nTrading stopped, loss is greater than 2 percent")
            break
        while True:
            df = pd.read_sql('BTCUSD1', engine)
            lookbackperiod = df.iloc[-lookback:]
            cumret = (lookbackperiod.Price.pct_change() + 1).cumprod() - 1
            BBlookbackperiod = df.iloc[-(lookback*20):]
            MA = BBlookbackperiod['Price'].mean()
            STDEV = 1.5 * BBlookbackperiod['Price'].std()
            BOLD = MA - STDEV
            entry = 0.3 * (MA / BOLD - 1)
            if not open_position:
                if lookbackperiod.iloc[-1].Price < BOLD and cumret[cumret.last_valid_index()] > entry:
                    order = df.iloc[-1]
                    print("\nTrade started at " + time.strftime("%d/%m/%Y %H:%M:%S"))
                    open_position = True
                    break
            if open_position:
                while True:
                    df = pd.read_sql('BTCUSD1', engine)
                    sincebuy = df.loc[df.DateTime > order.DateTime]
                    BBlookbackperiod = df.iloc[-(lookback*20):]
                    MA = BBlookbackperiod['Price'].mean()
                    STDEV = 2 * BBlookbackperiod['Price'].std()
                    BOLU = MA + STDEV
                    if len(sincebuy) > 1:
                        sincebuyret = (sincebuy.Price.pct_change() + 1).cumprod() - 1
                        last_entry = sincebuyret[sincebuyret.last_valid_index()]
                        if sincebuy.iloc[-1].Price > BOLU and last_entry > 0:
                            qty = qty * (1 + last_entry)
                            win += 1
                            print("Trade ended at " + time.strftime("%d/%m/%Y %H:%M:%S") +
                                  "\nStarting price: " + str(order.Price) + " Ending price: " + str(sincebuy.iloc[-1].Price) +
                                  "\nPrice change: " + str(last_entry) + " \nWins: " + str(win) + " \nLosses: " + str(loss) +
                                  "\nBTC amount: " + str(qty))
                            open_position = False
                            break
                        if -0.0015 < 1 - BOLU / order.Price:
                            stoploss = -0.0015
                        else:
                            stoploss = 1 - BOLU / order.Price
                        if last_entry < stoploss:
                            qty = qty * (1 + last_entry)
                            loss += 1
                            print("Trade ended at " + time.strftime("%d/%m/%Y %H:%M:%S") +
                                  "\nStarting price: " + str(order.Price) + " Ending price: " + str(sincebuy.iloc[-1].Price) +
                                  "\nPrice change: " + str(last_entry) + " \nWins: " + str(win) + " \nLosses: " + str(loss) +
                                  "\nBTC amount: " + str(qty))
                            open_position = False
                            break
```

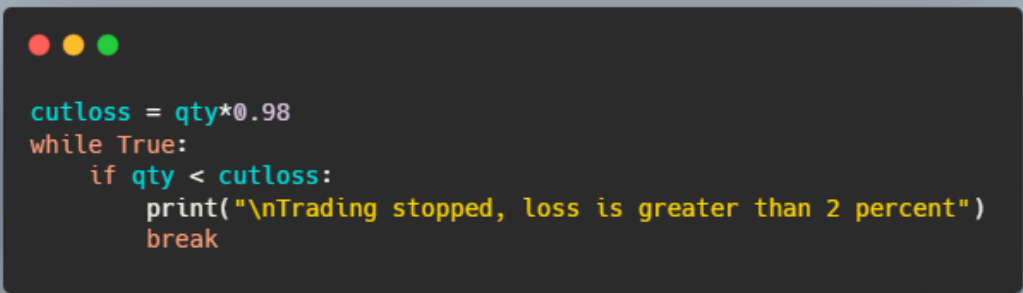
Slika 3 Funkcija BBSMomentumStrategy



```
engine = sqlalchemy.create_engine('sqlite:///stream.db')
```

Slika 4 Kod za engine

Engine je varijabla koja stvara SQL vezu sa bazom podataka gdje su spremljene cijene kriptovaluta te nam omogućava čitanje i pisanje u tu bazu podataka.



```
cutloss = qty*0.98
while True:
    if qty < cutloss:
        print("\nTrading stopped, loss is greater than 2 percent")
        break
```

Slika 5 Kod za cutloss

Zaustavlja funkciju BBSMomentumStrategy ako se izgubi više od 2% početne vrijednosti kapitala.

```

while True:
    df = pd.read_sql('BTCUSD', engine)
    lookbackperiod = df.iloc[-lookback:]
    cumret = (lookbackperiod.Price.pct_change() + 1).cumprod() - 1
    BBlookbackperiod = df.iloc[-(lookback*20):]
    MA = BBlookbackperiod['Price'].mean()
    STDEV = 1.5 * BBlookbackperiod['Price'].std()
    BOLD = MA - STDEV
    entry = 0.3 * (MA / BOLD - 1)
    if not open_position:
        if lookbackperiod.iloc[-1].Price < BOLD and cumret[cumret.last_valid_index()] > entry:
            order = df.iloc[-1]
            print("\nTrade started at " + time.strftime("%d/%m/%Y %H:%M:%S"))
            open_position = True
            break

```

Slika 6 Kod za uvjete kupnje

Glavni dio funkcije, petlja u kojoj izračunavamo kada će funkcija kupiti.

Df je varijabla koja pomoću engine čita trenutne podatke baze podataka stream.db.

Lookbackperiod je varijabla koja pomoću df siječe bazu podataka te uzima samo zadnje lookback podatke gdje je lookback broj koji nam govori koliki je period analize podataka.

Cumret uzima lookbackperiod-ove cijene te stvara kumulativni proizvod cijena. Ukratko, cumret nam govori koliko se cijena promijenila od najstarije do najnovije cijene (u postocima) u cijenama od lookbackperiod.

BBlookbackperiod je sličan lookbackperiod-u ali se koristi u svrhu veće analize podataka kako bi mogli stvoriti Bollinger Bands.

Bollinger Bands ili BB se računaju tako da uzmemo prosječnu vrijednost ili MA nekog skupa brojeva te njihovu standardnu devijaciju, standardnu devijaciju ili STDEV pomnožimo sa 1.5 (Može biti više ali je za naše svrhe 1.5 „zlatni broj“) te raspon od MA-STDEV i MA+STDEV se zove raspon od Bollinger Bands i govori nam otprilike gdje se promjena cijena dešava te za koliko bi se cijena mogla promijeniti unutar jednog trenutka.

BOLD nam govori najmanju cijenu raspona Bollinger Bands i izračunava se formulom MA-STDEV.

MA nam govori prosječnu cijenu od cijena BBlookbackperiod-a.

Entry, $0.3 * (MA / BOLD - 1)$ nam govori koliko je MA veći od BOLD u postotcima, množimo sa 0.3 kako bi zadovoljili uvjet (više o tome dalje u dokumentu).

Nakon „if not open_position“ uvjeta koji nam govori ako smo već kupili ili nismo možemo nastaviti na provjeru uvjeta za kupnju.

Prvi uvjet za kupnju je da je zadnja cijena veća od BOLD:

„lookbackperiod.iloc[-1].Price < BOLD“

Poanta prvog uvjeta je pronaći trenutak gdje su cijene ispod BB raspona jer cijene inače žele ostati u tom rasponu BB.

Drugi uvjet za kupnju je da je cumret veći od varijable entry, a entry je pomnožen sa 0.3 kako bi mogli koristiti ostatak razlike $MA / BOLD - 1$ za profit:

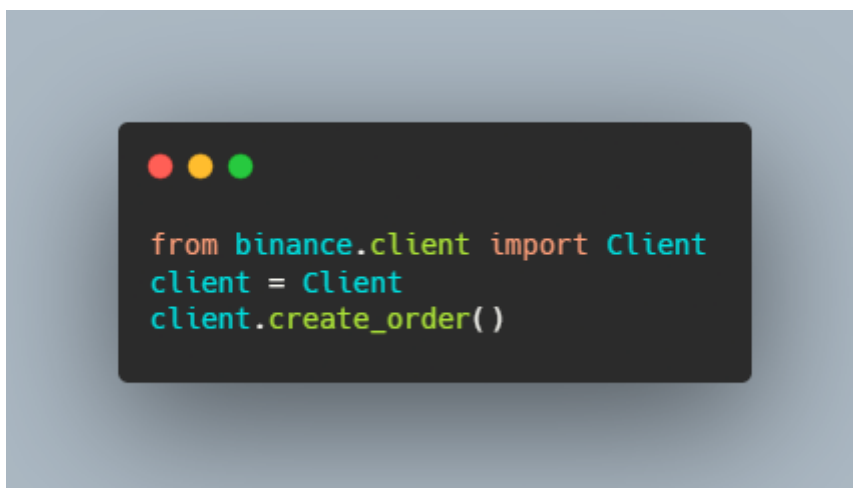
„cumret[cumret.last_valid_index()] > entry“

Poanta drugog uvjeta je pronaći trenutak gdje cijena dobiva „zamah“, trenutak gdje cijena počinje rasti kako bi mogli kupiti prije nego što nastavi rasti.

Vizualizacija ovog postupka se nalazi u slici broj 2.

Order je varijabla koja zapisuje zadnju cijenu, to jest cijenu na kojoj smo kupili.

Napomena: Ovaj program može stvoriti prave narudžbe ali se koristi za edukacijske potrebe. Kod za prave narudžbe je slijedeći:



Slika 7 Kod za mogućnost prave narudžbe

Client je programska zamjena za pravu osobu u mjenjačnici binance putem koje možemo stvarati narudžbe poput „client.create_order()“.

Nakon što smo napravili narudžbu trebamo odrediti trenutak kada ćemo ju zaustaviti to jest kada ćemo prodati:

Napomena: „print()“ funkcije su maknute iz koda u slici kako bi se bolje razumjela logika koda.

```
if open_position:
    while True:
        df = pd.read_sql('BTCUSDT', engine)
        sincebuy = df.loc[df.DateTime > order.DateTime]
        BBlookbackperiod = df.iloc[-(lookback*20):]
        MA = BBlookbackperiod['Price'].mean()
        STDEV = 2 * BBlookbackperiod['Price'].std()
        BOLD = MA + STDEV
        if len(sincebuy) > 1:
            sincebuyret = (sincebuy.Price.pct_change() + 1).cumprod() - 1
            last_entry = sincebuyret[sincebuyret.last_valid_index()]
            if sincebuy.iloc[-1].Price > BOLD and last_entry > 0:
                qty = qty * (1 + last_entry)
                win += 1
                open_position = False
                break
            if -0.0015 < 1 - BOLD / order.Price:
                stoploss = -0.0015
            else:
                stoploss = 1 - BOLD / order.Price
            if last_entry < stoploss:
                qty = qty * (1 + last_entry)
                loss += 1
                open_position = False
                break
```

Slika 8 Kod za uvjete prodaje

„If open_position“ uvjet nam govori jesmo li kupili ili ne, a pošto jesmo možemo nastaviti.

Df varijabla opet učitava bazu podataka pomoću engine-a.

Sincebuy varijabla nam govori sve podatke iz baze podataka df koje su upisane nakon narudžbe.

Ovaj put STDEV množimo sa 2 kako bi dobili veći prostor za profit.

Drugi dio BB raspona nakon BOLD je BOLD, računa se putem formule MA + STDEV

„If len(sincebuy) > 1“ uvjet je namijenjen za slučaj ako još nismo upisali cijenu u bazu podataka varijable nakon narudžbe kako ne bi došlo do greške u programu zbog nedostatka podataka u varijabli sincebuy.

Sincebuyret je ista varijabla kao i cumret samo što uzima kumulativni proizvod cijena od varijable sincebuy.

Last_entry uzima zadnju promjenu cijene od varijable sincebuyret.

„If sincebuy.iloc[-1].Price > BOLU and last_entry > 0“ je uvjet koji treba biti zadovoljen kako bi prodali kriptovalutu.

Prvi je uvjet da je zadnja uzeta cijena veća od BOLU, gornji dio raspona BB.

Drugi je uvjet da je profit pozitivan.

Komanda

```
„if -0.0015 < 1 - BOLU / order.Price:
```

```
stoploss = -0.0015
```

```
else:
```

```
stoploss = 1 - BOLU / order.Price“
```

određuje stoploss, to jest kada je najbolje prodati kako bi izgubili što manje novca.

„If -0.0015 < 1 - BOLU / order.Price“ je uvjet koji nam govori ako je promjena cijene od 1 - BOLU/(cijena na kojoj smo kupili) veća od -0.15% stoploss će biti -0.15%, a ako nije onda će stoploss biti 1 - BOLU/(cijena na kojoj smo kupili).

Ovaj dio koda pomaže sa nesigurnošću promjene cijene, ako je BB raspon malen onda će i stoploss biti malen. Ako je velik onda će stoploss biti maksimalno -0.15% kako ne bi došlo do masivnog gubitka novca.

Pri prodaji određujemo da je „open_position false“ kako bi mogli opet napraviti narudžbu.

3.3.2 Opis koda baze podataka

Sljedeća slika, „Slika 9“ prikazuje kod za stvaranje baze podataka:

```
#Delete old database
if os.path.exists('stream.db'):
    os.remove('stream.db')

#Global variable
StrategyTimePeriod = 3600

def DatabaseShortening(engine):
    sqliteConnection = sqlite3.connect('stream.db')
    cursor = sqliteConnection.cursor()
    df = pd.read_sql('BTCUSDT', engine)
    DateTime = df.iloc[0].DateTime
    cursor.execute('DELETE FROM BTCUSDT WHERE DateTime == ?', (str(DateTime),))
    sqliteConnection.commit()
    sqliteConnection.close()

def createframe(msg):
    df = pd.DataFrame([msg])
    df = df.loc[:,['s','E','p']]
    df.columns = ['symbol','DateTime','Price']
    df.Price = df.Price.astype(float)
    return df

async def main():
    engine = sqlalchemy.create_engine('sqlite:///stream.db')
    client = await AsyncClient.create()
    socket = BinanceSocketManager(client).trade_socket('BTCUSDT')
    async with socket:
        msg = await socket.recv()
        frame = createframe(msg)
        dupecheck = frame
        for i in range(StrategyTimePeriod):
            while frame.DateTime.equals(dupecheck.DateTime):
                msg = await socket.recv()
                frame = createframe(msg)
            frame.to_sql('BTCUSDT', engine, if_exists='append', index=False)
            dupecheck = frame
```

Slika 9 Kod baze podataka

```

if os.path.exists('stream.db'):
    os.remove('stream.db')

```

Slika 10 Kod za obrisati staru bazu podataka

Ovaj dio koda učitava (ako postoji) staru bazu podataka „stream.db“ te ju briše.

```

def DatabaseShortening(engine):
    sqliteConnection = sqlite3.connect('stream.db')
    cursor = sqliteConnection.cursor()
    df = pd.read_sql('BTCUSDT', engine)
    DateTime = df.iloc[0].DateTime
    cursor.execute('DELETE FROM BTCUSDT WHERE DateTime == ?', (str(DateTime),))
    sqliteConnection.commit()
    sqliteConnection.close()

```

Slika 11 Kod za skratiti trenutnu bazu podataka

SqliteConnection je varijabla putem koje se spajamo na bazu podataka „stream.db“ putem SQL³-a, ova varijabla nam omogućava čitanje i brisanje te baze podataka.

Cursor je varijabla poput varijable engine koja je spomenuta kod koda za analizu cijene, koristi sqliteConnection kako bi mogla izvršavati čitanje i brisanje baze podataka.

Df je varijabla koja učitava podatke baze podataka cijena.

Varijable cursor i sqliteConnection koriste različitu biblioteku od varijable df, dok su cursor i sqliteConnection komande slične programskom jeziku SQL, df koristi python-ovu biblioteku za brisanje i čitanje baze podataka.

Koristimo dvije biblioteke kako bi lakše mogli doći do rezultata, svaka ima svoju korist.

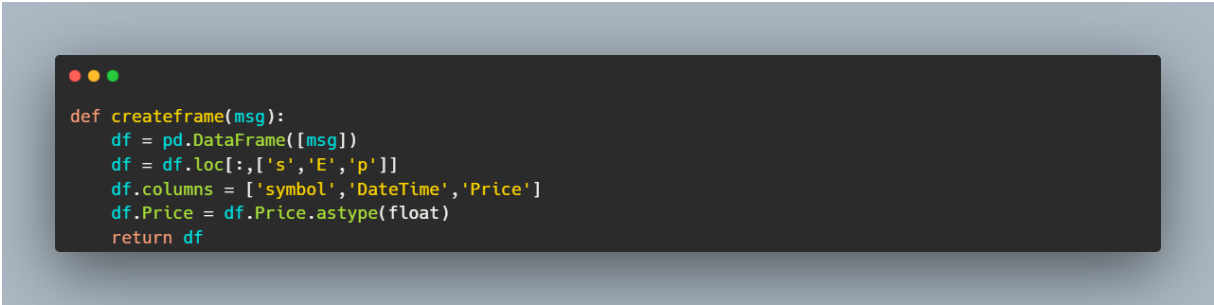
DateTime je varijabla koja ima vrijednost zadnjeg vremena u podacima varijable df.

Komanda „cursor.execute('DELETE FROM BTCUSDT WHERE DateTime == ?', (str(DateTime),))“ briše svaki redak baze podataka koji ima datum manji od varijable DateTime.

Komanda „sqliteConnection.commit()“ sprema promjene u bazu podataka „stream.db“.

³ Eng. Structured Query Language

Komanda „`sqliteConnection.close()`“ zatvara vezu sa bazom podataka „`stream.db`“ i varijablom `sqliteConnection`.



```
def createframe(msg):  
    df = pd.DataFrame([msg])  
    df = df.loc[:,['s','E','p']]  
    df.columns = ['symbol','DateTime','Price']  
    df.Price = df.Price.astype(float)  
    return df
```

Slika 12 Kod za preoblikovanje API poruke

Df je varijabla koja učitava podatke poruke varijable `msg` u format baze podataka, `msg` je varijabla koja je dobivena pozivom API-a za dobivanje cijena kriptovaluta.

Komanda „`df = df.loc[:,['s','E','p']]`“ pronalazi stupce „`s`“, „`E`“ i „`p`“ to jest stupce „`symbol`“, „`datum`“ i „`cijena`“ te mijenja varijablu `df` da pokazuje samo te stupce.

Komanda „`df.columns = ['symbol','DateTime','Price']`“ mijenja imena prošlih stupca „`s`“, „`E`“ i „`p`“ u „`symbol`“, „`DateTime`“ i „`Price`“.

Komanda „`df.Price = df.Price.astype(float)`“ formatira sve retke stupca „`Price`“ da budu float kako bi mogli imati decimalne točke.

Komanda „`return df`“ vraća varijablu `df`.

```

async def main():
    engine = sqlalchemy.create_engine('sqlite:///stream.db')
    client = await AsyncClient.create()
    socket = BinanceSocketManager(client).trade_socket('BTCUSDT')
    async with socket:
        msg = await socket.recv()
        frame = createframe(msg)
        dupecheck = frame
        for i in range(StrategyTimePeriod):
            while frame.DateTime.equals(dupecheck.DateTime):
                msg = await socket.recv()
                frame = createframe(msg)
                frame.to_sql('BTCUSDT', engine, if_exists='append', index=False)
                dupecheck = frame
        while True:
            while frame.DateTime.equals(dupecheck.DateTime):
                msg = await socket.recv()
                frame = createframe(msg)
                frame.to_sql('BTCUSDT', engine, if_exists='append', index=False)
                dupecheck = frame
            DatabaseShortening(engine)

if __name__ == "__main__":
    loop = asyncio.get_event_loop()
    loop.run_until_complete(main())

```

Slika 13 Glavni kod baze podataka

Engine je varijabla koja učitava novo stvorenu bazu podataka „stream.db“.

Client je varijabla koja se koristi za stvaranje veze između Binance API-a i našeg programa kako bi mogli stvarati API zahtjeve.

Socket je varijabla koja otvara „socket“ to jest vezu između varijable client i Binance API-a, kako bi mogli stvarati zahtjeve na kriptovalutu „Bitcoin“.

Komanda „async with socket:“ koristi se kako bi mogli stvoriti mjesto gdje naš program čeka da nam se zahtjev ispuni, zato je asinkrono ili „async“, mi čekamo.

Varijabla msg je poruka koju smo dobili zbog zahtjeva za trenutnu cijenu kriptovalute „Bitcoin“.

Frame je varijabla msg koja je prošla kroz funkciju „createframe()“ kako bi poruka bila povoljna za upis u bazu podataka.

Dupecheck je varijabla koju ćemo koristiti kako bi mogli identificirati identične upise u bazu podataka.

Koristimo komandu „for i in range(StrategyTimePeriod):“ kako bi prvo stvorili broj upisa u bazu podataka jednak varijabli StrategyTimePeriod, a vrijednost te varijable je 3600.

Petlja „while frame.DateTime.equals(dupecheck.DateTime):“ stvara varijablu frame više puta sve dok ta varijabla nije jednaka varijabli dupecheck kako ne bi bilo duplikatnih redaka u bazi podataka te nakon što je taj uvjet zadovoljen, možemo stvoriti upis u bazu podataka.

Komanda „frame.to_sql('BTCUSDT', engine, if_exists='append', index=False)“ upisuje to jest nadodaje varijablu kao redak u bazu podataka „stream.db“.

Nakon što smo upisali podatak opet pokrećemo komandu „dupecheck = frame“ kako bi mogli koristiti varijablu dupecheck za micanje identičnih vrijednosti.

Kada „for“ petlja završi stvaramo „while True:“ petlju koja se ponavlja beskonačno mnogo puta. Sve su komande u toj petlji iste kao i u prošloj „for“ petlji osim funkcije „DatabaseShortening(engine)“ koja skraćuje bazu podataka za 1 redak nakon svakog novog retka kako ne bi bilo nepotrebnih upisa.

4 ZAKLJUČAK

Moje je osobno mišljenje o trgovanju sa kriptovalutama da je taj proces veoma riskantan. Moglo bi se reći da su bot-ovi za trgovanje profitabilni, no to su spekulacije koje ne mogu biti potvrđene zbog same prirode bilo kojeg tržišta. Skoro pa nijedno tržište se ne može predvidjeti samo sa promjenom cijene, pošto postoje vanjski utjecaji, a tehnička analiza nije dovoljna da bi se predvidjele promjene u tržištu sa potpunom sigurnošću.

Putem blockchain tehnologije koja je veoma korisna za decentralizirane sustave može se stvoriti mnogo zanimljivih i inovativnih programa koji još nemaju dovoljno utemeljenu svrhu kako bi se mogli primijeniti u pravom svijetu. Zbog tog su razloga blockchain tehnologija i kriptovalute, po mom mišljenju, još uvijek nedovoljno razvijene te im je teško pridodati realnu vrijednost.

5 LITERATURA

1. Carbon.now.sh. 2022. *Carbon | Create and share beautiful images of your source code*. [online] Dostupno na: <<https://carbon.now.sh/>> [Pristupljeno 09.05.2022.].
2. Docs.python.org. 2022. asyncio — Asynchronous I/O — Python 3.10.4 documentation. [online] Dostupno na: <<https://docs.python.org/3/library/asyncio.html>> [Pristupljeno 24.04.2022.].
3. Docs.python.org. 2022. subprocess — Subprocess management — Python 3.10.4 documentation. [online] Dostupno na: <<https://docs.python.org/3/library/subprocess.html>> [Pristupljeno 23.04.2022.].
4. Docs.sqlalchemy.org. 2022. Overview — SQLAlchemy 1.4 Documentation. [online] Dostupno na: <<https://docs.sqlalchemy.org/en/14/intro.html>> [Pristupljeno 24.04.2022.].
5. Educative: Interactive Courses for Software Developers. 2022. What is pandas in Python?. [online] Dostupno na: <<https://www.educative.io/edpresso/what-is-pandas-in-python>> [Pristupljeno 24.04.2022.].
6. Home, P., support, c. and </form>, 2022. How to Install NumPy {Windows, Linux and macOS}. [online] Knowledge Base by phoenixNAP. Dostupno na: <<https://phoenixnap.com/kb/install-numpy#>> [Pristupljeno 24.04.2022.].
7. Matplotlib.org. 2022. Pyplot tutorial — Matplotlib 2.0.2 documentation. [online] Dostupno na: <https://matplotlib.org/2.0.2/users/pyplot_tutorial.html> [Pristupljeno 24.04.2022.].
8. Pastebin. 2022. [online] Dostupno na: <<https://pastebin.com/iFSLxbLB>> [Pristupljeno 05.03.2022.].
9. Visual Studio. 2022. Visual Studio 2022 | Download for free. [online] Dostupno na: <<https://visualstudio.microsoft.com/vs/>> [Pristupljeno 23.04.2022.].
10. Youtube.com. 2022. *How To Build a Simple Cryptocurrency Live Trading Bot with Python using the Binance API*. [online] Dostupno na: <https://www.youtube.com/watch?v=rc_Y6rdBqXM&t=284s> [Pristupljeno 05.03.2022.].

6 PRILOZI

6.1 Popis slika

Slika 1 Primjer baze podataka	8
Slika 2 Primjer narudžbe i zadovoljenih uvjeta	12
Slika 3 Funkcija BBsMomentumStrategy	13
Slika 4 Kod za engine	14
Slika 5 Kod za cutloss	14
Slika 6 Kod za uvjete kupnje	15
Slika 7 Kod za mogućnost prave narudžbe	17
Slika 8 Kod za uvjete prodaje	18
Slika 9 Kod baze podataka	20
Slika 10 Kod za obrisati staru bazu podataka	21
Slika 11 Kod za skratiti trenutnu bazu podataka	21
Slika 12 Kod za preoblikovanje API poruke	22
Slika 13 Glavni kod baze podataka	23

6.2 Popis tablica

Tablica 1 Biblioteke programa	10
-------------------------------	----

Rb.	Datum konzultacija	Bilješke o napredovanju	Potpis mentora

Prilog br. 3: Bilješke mentora – posljednja stranica rada

Mentor je prihvatio izradbu: _____
(potpis)

Datum predaje rada: _____

Ocjena pisanog rada: _____

Datum obrane rada: _____

Ocjena obrane rada: _____

Konačna ocjena: _____

Povjerenstvo:

1. mentor: _____

2. profesor struke: _____

3. profesor struke: _____

Prostor za izdvojeno mišljenje ili eventualni komentar: