

# ДИПЛОМНА РАБОТА

на тема

.....

Дипломант: XXX XXX XXX

ученик от 12А клас в

ЧПГДН "СофтУни Светлина"

Научен ръководител: XXX XXX XXX

Рецензент: XXX XXX XXX

Дата: 12 май 2023

Сесия: май-юни 2023 г.

София, 2023 г.

## Съдържание

ДИПЛОМНА РАБОТА	1
Съдържание	2
Увод	3
Проблем	4
Цели на дипломния проект	5
Задачи, произтичащи от целите	6
Глава 1. Важни Базови термини и терминология	7
Глава 2. Индикатори	15
Глава 3. Проектиране и имплементация	20
Свързване на програмата миналото на пазара	20
Свързване на програмата с real time пазара	29
Внедряване	32
Глава 4. Ръководство за потребителя (смени заглавието)	32
Заклучение	33
Информационни източници	33
Рецензия на дипломен проект	34

## Увод

### Идея за проекта

Реших да избира темата за бот, с алгоритми, който търгува вместо вас, защото имам голям интерес към това, да създам нещо с което да си изкарвам парите, без да се налага да участвам пряко в процеса. Още от малък представата ми за това, как да си изкарвам парите, не е била свързана с типичната за повечето хора работа от девет до пет. Аз съм си бунтар още от малък и мразя някой да ме командва. Харесвам да правя това, което аз искам и сам на себе си да съм си шеф. Основната идея зад този бот е именно това, аз да създам нещо, което да работи за мен. В началото тръгнах да го правя с изкуствен интелект, но в следствие установих, че лаптопа ми не е достатъчно добър, за да поддържа нещо, толкова голямо и сложно. Затова реших да го направя по малко по – математически начин, като ползвам алгоритми и индикатори.

Решението ми да направя крипто бот, също беше много повлияно от най – близкият ми приятел. Той е много умно момче, макар на моменти да не му личи, винаги е готов да помогне на приятел в нужда и е изключително благонамерен. Когато на мен ми трябваха пари, той ми предложи да ме научи да тъгувам крипто, тъй като това е нещо, с което той се занимава и е изключително добър. Прекарахме цяло лято заедно, като той ми обясняваше всичко. Даде ми книги да чета, дори ми прати пари, с които аз да тъгувам. След известно време аз се отказах, тъй като не беше моето нещо, но той не се ядоса, нито си поиска парите обратно. Той ме подкрепи в решението ми и каза, че ако имам нужда от някакъв съвет в крипто начинанието, винаги ще е там да ми помогне. До ден днешен аз пазя парите, които ми беше пратил. Това, че съм ги запазил, се оказа много позитивна случка, тъй като сега те ще бъдат инвестирани именно в това, да разработя бота, който ще търгува вместо вас.

### Какво представлява крипто търговията

Крипто търговията е процес на купуване, продаване и обмен на цифрови активи, познати още като криптовалути. Процесът споменат в предното изречение се изпълнява чрез онлайн платформи или борси, познати още като ексчейнджи. Търговията на криптовалути е нещо сравнително ново. Много хора вярват, че там се намира бъдещето на финансите и лека полека всички започват да се насочват

натам. Кripto търговията е много по – различна от традиционните фондови пазари, въпреки че хората, не разбиращи от двете неща, не биха намерили разлика. В традиционната търговия на акции, която се извършва на централизирана борса, има трета страна, която извършва комуникацията между купувача и продавача, докато при търговията на крипто, това действие се извършва на децентрализирани борси - там не е замесено трето лице и комуникацията между „продавач“ и купувач се извършва директно. Още един от плюсовете на крипто търговията е, че пазара е отворен постоянно, за разлика от този на акциите. Както знаете акции не могат да се купуват по време на празнични дни и имат определени часови периоди, в които могат да се продават и купуват. Ключът към успешната крипто търговия е разбирането на различните видове налични активи и научаването как да разчитате пазара и да идентифицирате печеливши трендове. Кripto активите се предлагат в различни форми, включително монети, токени и деривати. Монетите са най-често срещаният тип актив, но токениите също набират популярност, тъй като предлагат повече възможности за инвеститорите. Дериватите са форма на актив, който извлича стойността си от базов актив. Трейдърите също трябва да разбират различните стратегии за търговия и как да идентифицират добри входни и изходни точки. Техническият анализ помага на трейдърите да вземат информирани решения кога да купуват, продават или държат активите си. Фундаменталният анализ помага на трейдърите да разберат какво движи пазара и как може да се държи в бъдеще. Кripto търговията е рисковано, но потенциално печелившо начинание за тези, които разбират пазарите и имат правилните инструменти и стратегии.

## Проблем

### Ботове за крипто търговия

Ботовете за крипто трейдване са автоматизирани програми, предназначени да анализират и търгуват криптовалути въз основа на предварително определени условия. Кripto ботовете могат да се използват за изпълнение на много стратегии за трейдване, от прости като купуване и държането на валутата, до сложни алгоритмични програми, базирани на технически индикатори. На пазара има различни видове ботове за крипто трейдване, всеки от които отговаря на различни тактики в трейдването. Най-често използваният бот е ботът за маркет мейкър, който е предназначен да осигури

ликвидност (колко лесно даден актив може да превърнат в пари в брой, без да бъде засегната неговата пазарна цена.) на пазара. Този тип бот помага за преодоляване на пропастта между купувачи и продавачи, като прави поръчки между тях. Той играе ролята на мост между купувача и продавача. Той е програмиран да поставя лимитирани поръчки на малко по-ниска от текущата пазарна цена и малко по-висока от текущата пазарна цена. Това помага за увеличаване на ликвидността на пазара и улеснява трейдването за купувачи и продавачи. Друг тип бот за крипто трейдване е арбитражният бот, който е предназначен да се възползва от несъответствията в цените между борсите. Този тип бот търгува на различни пазари и борси, за да улови разликата в цените. По този начин той може да се възползва от разликите в цените и да се възползва от тях, за да изкара печалби. Третият тип бот за крипто трейдване е ботът за следване на трендовете. Той е програмиран да следва трендовете на пазара. Този тип бот е предназначен да следи текущата пазарна инерция и да се възползва от нея. Той е програмиран да купува, когато цените се покачват, и да продава, когато цените падат, като по този начин бързо прави печалби. И накрая, има ботове за алгоритмична трейдване, които са програмирани с алгоритъм за изпълнение на определен стратегии за трейдване. Тези ботове са програмирани да анализират данни, да идентифицират модели и да извършват сделки въз основа на модела, които разпознават. Алгоритмичните ботове за трейдване стават все по-популярни, тъй като осигуряват по-голяма точност от ръчното трейдване.

## Цели на дипломния проект

### Цели на бота за търговия на криптовалути:

**Автоматизиране на търговията** - Ботът може да бъде програмиран да изпълнява определени търговски стратегии и да открива възможности за печалба, а вие дори не трябва да гледате пазара.

**Увеличаване на ефективността** - Ботът може да обработва големи обеми от данни по-бързо, отколкото човек, като резултатът е увеличаване на ефективността и постигане на по-големи печалби.

**Минимизиране на човешките грешки** - Човешките грешки могат да бъдат изключени, когато търговията е автоматизирана, тъй като ботът ще следи пазарните условия и ще изпълнява търговските стратегии, без да се излага на човешки емоционални състояния.

**Създаване на допълнителен доход** - Ботът може да бъде настроен да работи непрекъснато, като резултатът е възможността за генериране на допълнителен доход.

#### **Плюсове на бота за търговия на криптовалути:**

**Скорост** - Ботът може да извършва търговията с криптовалути в реално време, като реагира на промените на пазара много по-бързо отколкото човек.

**Автоматизация** - Ботът може да бъде програмиран да изпълнява определени търговски стратегии и да автоматизира процеса на търговията, като резултатът е минимизиране на човешките грешки.

**Постоянство** - Ботът може да работи 24 часа в денонощието, 7 дни в седмицата, без да се нуждае от почивка, като гарантира постоянство в търговията.

## **Целта на настоящата дипломна работа е да бъде ...**

Във връзка с поставената цел за разработване на бот, търгуващ криптовалути :

- Извършване на проучване и подготовка на обзор на проблемната област
- Проучване на инструментите и технологиите за работа с пазари и технически анализи в Python
- Проектиране на система, работеща на миналото на пазара, анализираща поведението на програмата
- Реализация на системата в истинския пазар, анализ на програмата в истинския пазар, с хипотетични пари
- Извеждане на изводи и заключения на базата на разработения проект

## Глава 1. Важни, базови термини и терминологии

### 1.1 – High, Low, Closing and Open prices

**High price:** High price-а на дадена криптовалута се отнася до най-високата цена, която валутата е достигнал през определен период от време, като например, един ден, седмица или месец. Например, ако high price - а на биткойн за един ден е \$50 000, това означава, че в даден момент през този период за търговия цената на биткойн е достигнала или надхвърлила \$50 000. High price - а е значителен, защото представлява пиковата стойност на валутата през даден период от време и трейдърите често я използват като ориентир за идентифициране на потенциални нива на съпротива.

**Low price:** Low price-а, от друга страна, представлява най-ниската цена, която валутата е достигнал през определен период от време. Използвайки същия пример като преди, ако low price-а на биткойн за един ден е \$45 000, това означава, че в даден момент през този ден цената на биткойн е паднала до или под \$45 000. Low price - а е значителен, защото представлява най-ниската стойност на актива през даден период от време и трейдърите често я използват като референция за идентифициране на потенциални нива на support.

**Closing price:** Closing price-а е крайната цена на актива в края на определен период от време, като един ден, седмица или месец. Това е цената, на която се установява, когато пазарът затвори за този период. Например, ако closing price - а на биткойн за един ден е \$48 000, това означава, че в края на този ден цената на биткойн е била \$48 000. Closing price - а е важен, защото често се използва за изчисляване на технически индикатори като moving average - а и за определяне дали валутата е спечелила или загубил стойност през дадения период от време.

**Open price:** Open price-а е ценовото ниво на валутата в началото на определен период от време. Това е цената, на която валутата е започнала да се търгува при отварянето на пазара за този период. Например, ако Open price-а на биткойн за един ден е \$47 000, това означава, че в началото на този

ден цената на биткойн е била \$47 000. *Open price* - а е важен, защото предоставя на трейдърите начална референтна точка за търговската активност през деня и може да се използва за идентифициране на потенциални *price gap* - ове или възможности за трейдване.

Разбирането на тези четири концепции е от решаващо значение за търговците, за да вземат информирани решения кога да купуват или продават валутите. Чрез анализиране на *high*, *low*, *open* и *close* цените на даден актив, търговците могат да придобият представа за пазарните трендове, да идентифицират потенциални нива на *support* и *resistance* и да разработят съответните стратегии за търговия.

## 1.2 - *Candles, Bullish and Bearish markets, Zero line*

**Candles:** В техническия анализ кендъла е визуално представяне на движението на цената за определен период от време. Всеки кендъл показва *opening*, *closing*, *high* и *low* цените на даден актив през този период от време. Тялото на кендъла представлява *opening* и *closing* цените, докато "*wicks*" или "*shadows*" над и под тялото представляват съответно *high* и *low* цените. Кендълите могат да се използват за идентифициране на трендове и *pattern* - ите в движенията на цените, като *bullish* или *bearish* трендове.

**Bullish market:** *Bullish market* - а се отнася за период от време, когато цената и трендовете са положителни, като цените обикновено се повишават доста. По време на *bullish market*-а, инвеститорите и трейдърите са оптимисти и уверени в бъдещия потенциал за растеж на пазара. *Bullish market* - ите често се характеризират с големи обеми на търговия и положителни новини и събития, които повишават цените.

**Bearish market:** *Bearish market* - а се отнася до период от време, когато цените и трендовете са отрицателни, като цените намаляват значително. По време на *bearish market* инвеститорите и търговците са песимисти и несигурни за бъдещето на пазара. *Bearish market* често се характеризират с ниски нива на търговия, негативни новини и събития, които понижават цените.



**Zero line:** Zero line-a е хоризонтална линия на технически индикатори, като MACD (Moving Average Convergence Divergence), която представлява неутрална позиция на пазара. Това е точката, в която сигналната линия и линията MACD се пресичат, което показва, че няма трендове в нито една посока. Когато линията на MACD пресича над нулевата линия, това е bullish сигнал, показващ, че тренда се измества нагоре. Обратно, когато линията на MACD пресича под нулевата линия, това е bearish сигнал, показващ, че тренда се измества към низходяща посока.

Bullish и bearish пазарите и zero line-a са важни за търговците да вземат информирани решения относно покупката и продажбата на криптовалuti. Чрез анализиране на движенията на цените чрез кендъли, идентифициране на bullish и bearish трендове и използване на технически индикатори като zero line - а, търговците могат да получат представа за пазарните настроения и да разработят ефективни стратегии за търговия, които да максимизират техните печалби.

### 1.3 - Wick and Shadow

В техническия анализ термините "wick" и "shadow" се използват, за да опишат тънките линии, които се простират от горната и долната част на кендълите на графика.

**Wick или shadow**, които се простират от върха на кендъла, са известни като горен wick (опаищица), докато wick или shadow, които се намират на дъното на кендъла, са известни като долна опаищица или „сянка“. Дължината на опаищицата показва колко цената на актива се е преместила над или под opening или closing цената през периода на трейдването.

Например, ако началната цена на Bitcoin е \$50 000, а най-високата цена, която е достигнал през определен период от време, е \$52 000, тогава горната опаищица ще представляват разликата между

\$50 000 и \$52 000. Същото важи и за долната опашика, която представлява разликата между *low price* - а и *closing price* - а на актива през определения период.

***Wicks and shadows*** могат да предоставят ценна информация на търговците относно силата на тренда, потенциалните нива на *support* и *resistance* и общия вид на пазара. Например дългите горни опашици могат да показват, че има силен *resistance* на определено ниво на цената, докато дългите долни опашици могат да показват, че има силен *support* на определено ниво на цената.

В допълнение, *wicks and shadows* могат да се използва за идентифициране на потенциални точки на обръщане на цената. Ако кендъла има дълъга горна опашица, последван от дълъг долга дола „сянка“, това може да означава, че е имало много активност за покупка и продажба по време на периода на търговия, но активът в крайна сметка е *closing price* - а е бил на ниво, подобно на *opening price* - а. Това може да е знак за нерешителност на пазара, което може да доведе до обръщане на цената.

Като цяло разбирането на *wick*-овете и *shadow* - ите в техническия анализ може да предостави на търговците ценна информация за пазарните трендове, потенциални нива на *resistance* и *support* и точки на обръщане на цените.

#### ***1.4 – Histogram, MACD and Signal line***

Хистограмата е графично представяне на разликата между два *moving average* - а. Хистограмата показва кендъли, които са над или под нулева линия, показвайки дали *short-term moving average* - а е над или под *below the long - term moving average* - а.

Хистограмите са често използвани с *MACD (Moving Average Convergence Divergence)* показателите, които са апопулярни *momentum indicator*-и, използвани от трейдъри, за да определят потенциални *buy* и *sell* сигнали.

*MACD хистограмата бива създадена, като извадим 26 - period exponential moving average (EMA) от 12 - period EMA. Това изчисление създава линия, която се генерира над и под zero line-a, която представлява неутралната позиция на пазара.*

*След това хистограмата се генерира спрямо разликата между MACD линията и signal line-a, която обикновено е 9 - period EMA на MACD линията. Хистограмата показва кендъли, които са над или под zero line - а.*

*Когато MACD линията пресече над сигналната линия, кендълите на хистограмата стават положителни, което показва, че краткосрочния тренд е възходяща. Обратно, когато MACD линията пресече под сигналната линия, кендълите на хистограмата стават отрицателни, което показва, че краткосрочния тренд е низходяща.*

*Търговците използват хистограми в индикатора MACD, за да идентифицират потенциални сигнали за покупка или продажба. Например, когато кендълите на хистограмата станат положителни, това може да означава, че започва възходяща тренд (bullish market) и търговците може да обмислят закупуването на активите. От друга страна, когато кендълите на хистограмата станат отрицателни, това може да означава, че започва низходящ тренд (bearish market) и търговците може да обмислят продажбата на активите.*

*Като цяло хистограмите в техническия анализ предоставят на търговците визуално представяне на разликата между два moving average - а и могат да се използват за идентифициране на потенциални сигнали за покупка или продажба на валутата.*

### **1.5 – EMA**

ЕМА означава *Exponential Moving Average*, което е вид *moving average*, използван в техническите анализи, да се разпознават потенциални *buy* и *sell* сигнали.

*Moving average*-а е линия, която представлява средната *closing* цена на актив за определен период от време. Примерно ако в последните пет дни *closing* цената на биткойн е била: \$ 52,000, \$ 48,000, \$ 50,000, \$ 49,000 и \$ 54,000, *moving average*-а им се равнява на \$ 50,600  $((A1+A2+....)/n)$ . ЕМА се различава от *simple moving average* - а (*SMA*) по това, че придава по-голяма тежест на последните данни за цените, докато *SMA* дава еднаква тежест на всички данни за цените за периода.

ЕМА обикновено се използват в анализа на графиките за идентифициране на трендовете и потенциални нива на *support* или *resistance*. Линията ЕМА се изчислява, като се вземе средната стойност на *closing price* - а на валутата за определен период от време, като се дава по-голяма тежест на най-новите цени.

Например, 50-дневен ЕМА ще бъде изчислен, като се вземе средната стойност на *closing* цените на актива през последните 50 дни, като по - голяма тежест се дава на най-новите цени. Тъй като се добавят нови данни за цената, линията ЕМА ще се регулира, за да отрази новата информация.

Търговците използват ЕМА, за да идентифицират трендове и потенциални сигнали за покупка или продажба. Например, когато цената на даден актив е над ЕМА линията, това може да показва, че активът е във възходящ тренд и търговците може да обмислят закупуването на актива. От друга страна, когато цената е под ЕМА линията, това може да показва, че активът е в низходящ тренд и търговците може да обмислят продажбата на актива.

Трендовете се отнасят до общата посока, в която се движи цената на дадена криптовалута за определен период от време. Търговците използват анализ на трендовете, за да идентифицират *pattern*-и и потенциални сигнали за покупка или продажба на пазара.

Има три основни типа трендове: възходящ тренд, низходящ тренд и страничен (известен също като *range-bound*). Възходяща тенденция възниква, когато цената на дадена криптовалута непрекъснато нараства с течение на времето. Низходящ тренд възниква, когато цената последователно намалява с течение на времето. Страничен тренд възниква, когато цената се движи в сравнително тесен диапазон, без ясна посока.

Търговците използват различни инструменти за технически анализ, за да идентифицират трендовете, като трендови линии, *moving average* – и *pattern* - и на диаграми. Например тренд линията е права линия, която свързва две или повече ценови точки и може да помогне за идентифициране на общата посока на тренда.

Идентифицирането на трендовете е важно за търговците, защото може да предостави информация за потенциални сигнали за покупка или продажба. Например, ако дадена криптовалута е във възходящ тренд, търговците може да обмислят закупуването ѝ, тъй като това показва, че има голямо търсене. Обратно, ако дадена криптовалута е в низходящ тренд, търговците може да обмислят продажбата на актива, тъй като това показва, че има високо количество от него.

Важно е да се отбележи, че трендовете могат да се променят с времето въз основа на пазарните условия и търговската дейност. Търговците трябва непрекъснато да ги наблюдават и да променят стратегиите си в съответствие с тях.

### 1.7 - API Keys

*API (Application Programming Interface), е набор от правила и протоколи, които позволяват на различни софтуерни приложения да комуникират помежду си. В контекста на крипто търговията API се използват от ексчейнджите (борсите), за да позволят на разработчиците и търговците да имат достъп до своите данни за търговия и да извършват сделки чрез програма.*

***API ключът** е уникален идентификатор, който се използва за удостоверяване на потребителя и предоставяне на достъп до API - а. Това е като парола, която ви позволява да получите достъп до данните на борсата и да извършвате действия, като например извършване на трейдове или проверка на баланса на вашия акаунт. API ключът обикновено се предоставя от борсата и трябва да се пази поверителен, тъй като всеки с достъп до ключа може потенциално да извършва действия от ваше име.*

***API secret key** е по-дълъго и по-сложно словосъчетание от букви и цифри , който се използва за подписване на заявки, направени към API. Това е като цифров подпис, който гарантира автентичността на данните, които се предават. API secret key – а, също се предоставя от борсата и трябва да се пази поверителен, тъй като всеки с достъп до ключа може потенциално да променя данните, които се предават.*

*Имплементирането на API ключа във кода може да осигури няколко предимства. Първо, той позволява да се автоматизират стратегии за търговия и да се извършват сделки чрез програма, което може да бъде по-бързо и по-ефективно от ръчната търговия. Второ, позволява достъп до пазарни данни в реално време и вземане на информирани решения въз основа на най-новата информация. Трето, може да помогне за управление портфолиото по-ефективно чрез автоматично балансиране на финансите въз основа на предварително дефинирани правила (каквото е написано в кода, да бъде изпълнено).*

*Въпреки това е важно използването на API key – а да бъде внимателно и сигурно. API key - овете и API secret key - овете трябва да се палят поверителни и да не се споделят с никого. Освен това, винаги трябва кода да бъде тестван в безопасна среда, преди да бъде приложен в активен трейдинг портфейл, тъй като грешки и бъгове в кода могат да доведат до неочаквани загуби.*

## Глава 2. Индикатори

### 2.1 - MACD Indicator

Стратегията MACD (Moving Average Convergence Divergence) е популярен инструмент за технически анализ, използван от трейдърите за идентифициране на потенциални възможности за трейдване на финансовите пазари. MACD се изчислява с помощта на два moving average-a (ЕМА) за различни периоди от време и се използва за генериране на сигнали, които показват дали дадена валута е купувана твърде много или е продавана твърде много в даден период от време. Стратегията включва покупка и продажба на активи въз основа на тези сигнали.

MACD се състои от три основни компонента:

- 1. Линия MACD:** Това е разликата между 12-дневната ЕМА и 26-дневната ЕМА. Линията MACD се движи над и под нулевата линия, която е хоризонтална линия, която представлява точката на равновесие между купувачи и продавачи. Когато линията на MACD пресече над нулевата линия, това се счита за *bullish* сигнал, което показва, че инерцията се измества в полза на купувачите. Когато линията на MACD пресече под нулевата линия, това се счита за *bear* сигнал, което показва, че инерцията се измества в полза на продавачите.
- 2. Сигнална линия:** Това е 9-дневна ЕМА на линията MACD. Сигналната линия помага да се идентифицират потенциалните обръщания на трендовете. Когато линията на MACD пресече над сигналната линия, това се счита за *bullish* сигнал, което показва, че купувачите набират сила. Когато линията на MACD пресече под сигналната линия, това се счита за изходящ сигнал, което показва, че продавачите набират сила.

3. **Хистограма:** Хистограмата е разликата между MACD линията и сигналната линия. Хистограмата помага да се идентифицира силата на momentum-а. Когато хистограмата е положителна, това показва, че купувачите контролират, а когато е отрицателна, това показва, че контролират продавачите.

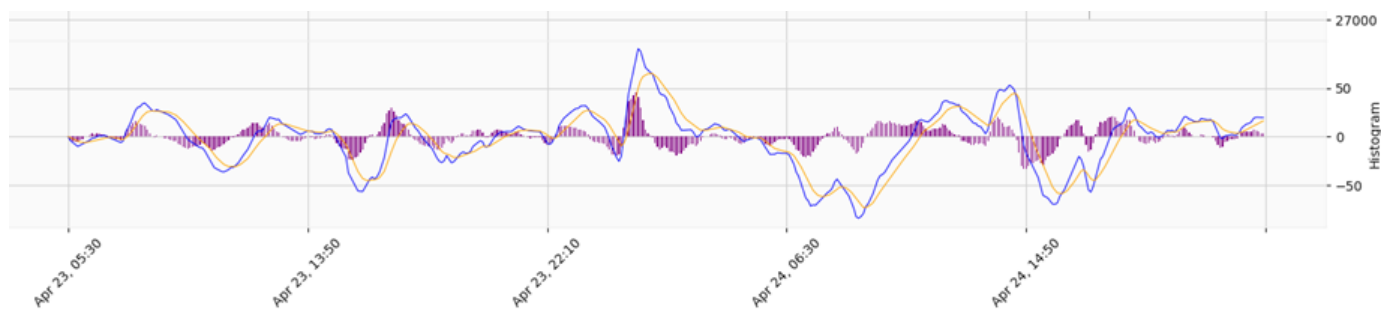
Стратегията MACD включва използването на тези три компонента за генериране на сигнали за покупка и продажба. Сигналите се генерират, когато MACD линията пресече над или под сигналната линия, или когато хистограмата пресече над или под нулевата линия. Сигнал за покупка се генерира, когато MACD линията пресече над сигналната линия или когато хистограмата пресече над нулевата линия. Сигнал за продажба се генерира, когато MACD линията пресече под сигналната линия или когато хистограмата пресече под нулевата линия.

Търговците могат да използват стратегията MACD, за да идентифицират потенциални възможности за трейдване на различни финансови пазари, включително акции, форекс и криптовалути. Въпреки това, като всяка стратегия за трейдване, стратегията MACD не е безпогрешна и трейдърите винаги трябва да бъдат внимателни и да използват подходящи техники за намаляване на риска.

MACD стратегията

На снимката по – долу може да се види, вече имплементираната MACD стратегия:





## 2.2 - Supertrend Indicator

Индикаторът *Supertrend* е популярен инструмент за технически анализ, който обикновено се използва от търговците за идентифициране на посоката на тренда в цената на даден актив.

В основата си индикаторът *Supertrend* се основава на два ключови параметъра: *Average True Range (ATR)* и *multiple factor* (често наричан „factor“ или „multiplier“). *ATR* измерва волатилността (колко често се променя цената) на даден актив, докато множителят определя силата на тренда.

За да изчислим *Supertrend*, започваме с изчисляване на *Average True Range (ATR)* за определен период от време. Това се прави, като се вземе средната стойност на *true range values*, които са най-големите от следните три стойности: разликата между *current high* и *current low*, разликата между *current high* и *previous low* или разликата между *current low* и *previous close*.

Ето пример за изчисляването на *ATR* за последните 3 дни:

Ден 1: *High* = \$50, *Low* = \$40, *Close* = \$45

***True Range*** =  $\max(50 - 40, \text{abs}(50 - 45), \text{abs}(40 - 45)) = \max(10, 5, 5) = 10$

***ATR*** = 10 (понеже това е първи ден, *ATR* се равнява на *True Range* - 10)

Ден 2: *High* = \$55, *Low* = \$43, *Close* = \$53

***True Range*** =  $\max(55 - 43, \text{abs}(55 - 45), \text{abs}(43 - 45)) = \max(12, 10, 2) = 12$

***ATR*** =  $[(\text{ATR} * 2) + \text{True Range}] / 3 = [(10 * 2) + 12] / 3 = 10.67$

Ден 3: *High* = \$58, *Low* = \$50, *Close* = \$55

***True Range*** =  $\max(58 - 50, \text{abs}(58 - 53), \text{abs}(50 - 53)) = \max(8, 5, 5) = 8$

$$ATR = [(ATR * 2) + True Range] / 3 = [(10.67 * 2) + 8] / 3 = 9.11$$

Това означава, че в последните 3 дни цената се е движела в диапазон от 9.11\$. Колкото по – голям е резултата от ATR, толкова по – голям шанс има, дадената валута да смени рязко посоката си.

След като имаме ATR, можем да го използваме за изчисляване на горната и долната линия на Supertrend. Горната линия се изчислява чрез умножаване на ATR по множителя и добавянето му към подвижната средна стойност на цената на актива. Долната линия се изчислява чрез изваждане на ATR, умножен по множителя, от подвижната средна стойност на цената на актива.

$$\begin{aligned} Upper\ band &= (moving\ average\ of\ price) + (ATR \times multiplier) \\ &= \$50 + (\$9.11 \times 2) \\ &= \$68.22 \end{aligned}$$

$$\begin{aligned} Lower\ band &= (moving\ average\ of\ price) - (ATR \times multiplier) \\ &= \$50 - (\$9.11 \times 2) \\ &= \$31.77 \end{aligned}$$

След това самият Supertrend се изчислява, като се вземе средната стойност на горната и долната линия. Когато цената на актива е над Supertrend, се счита, че е във възходящ тренд, докато цена под Supertrend се счита за низходящ.

$$\begin{aligned} Supertrend &= (upper\ band + lower\ band) / 2 \\ &= (\$68.22 + \$31.77) / 2 \\ &= \$49.99 \end{aligned}$$

Тоест, ако цената е \$51, тя ще е във възходящ тренд. И обратното, ако цената е \$47, то тя е в низходящ тренд

*Едно от основните предимства на индикатора Supertrend е, че той се адаптира към променящите се пазарни условия. Например, когато волатилността е висока, ATR ще бъде по-голям и линиите на Supertrend ще се разширят съответно. Обратно, когато волатилността е ниска, ATR ще бъде по-малък и линиите на Supertrend ще се стегнат.*

*В обобщение, индикаторът Supertrend е инструмент за технически анализ, който използва average true range(ATR) и multiplier, за да определи горната и долната линия на индикатора, следващ тренда. Това позволява на търговците да идентифицират посоката на трендовете и да се адаптират към променящите се пазарни условия.*

*Ето и снимка на индикатора, имплементиран в кода:*



*На снимката може да видите червените линии - това сигнализира, че цената е била под Supertrend-a (downtrend), а зелените линии, че цената е била над Supertrend-a, което означава, че е била в uptrend.*

## Глава 3. Проектиране и имплементация

Програмата е базирана на индикаторите описани по - нагоре в документацията. Основната идея е, че когато MACD линията пресече сигналната линия, програмата гледа дали Supertrend indicator - а е позитивен. Ако и двете условия са изпълнени, програмата купува крипто валутата, за която е предназначена, и гледа да изкара профит. Има два случая, в които програмата изкарва профити.

Първият е, когато Supertrend - а смени цвета си (от зелен на червен, тоест стане downtrend), вторият е, ако цената се вдигне с 2.5%. В случая на загубите, понякога сменянето на Supertrend - а води до загуба, но тя е минимална, тъй като няма как в крипто търговията да имате само победи. Вторият начин по който може да загубите е, ако цената падне с 0.05% под последния записан high на кендълите (за тази цел е създадена променлива, която се казва `highest_candle_price`, която има за цел, след покупката на даден актив да запазва в себе си най - високата стойност, която цената е стигнала от началото на покупката )

### Свързване на програмата с миналото на пазара

За създаването на един такъв бот, който да търгува на истинския пазар, винаги трябва да има тестване в изкуствена среда и спрямо миналото на пазара, защото не можете да пуснете нещо, което не сте сигурни дали ще работи, на пазара да търгува с истински пари. Тестван в миналото на пазара, този бот има успеваемост от около 80%, като това на снимката по - долу е статистиката на бота за 4 месеца:

```
38
39 ##### Wallet
40 wallet = 10000
41
42
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

THE HIGHEST CANDLE WAS: 28629.42

total profit: 71420.83  
total loss: -2168.70  
money in the wallet: 79252.13

W:926  
L:197

		date	side	price	amount	usdt	wallet
0	2023-01-01 00:25:00	buy	16535.54	0.181427	3000.000000	7000.000000	
1	2023-01-01 00:25:00	buy	16535.54	0.181427	3000.000000	4000.000000	
2	2023-01-01 02:40:00	sell	16550.37	0.362855	6005.381137	10005.381137	
3	2023-01-01 09:20:00	buy	16515.92	0.181741	3001.614341	7003.766796	
4	2023-01-01 09:20:00	buy	16515.92	0.181741	3001.614341	4002.152455	
...	...	...	...	...	...	...	
3364	2023-05-01 10:50:00	buy	28522.72	0.831225	23708.800603	31611.734137	
3365	2023-05-01 11:10:00	sell	28564.25	1.662450	47486.642762	79098.376898	
3366	2023-05-01 12:25:00	buy	28536.97	0.831536	23729.513070	55368.863829	
3367	2023-05-01 12:25:00	buy	28536.97	0.831536	23729.513070	31639.350759	
3368	2023-05-01 13:30:00	sell	28629.42	1.663072	47612.777114	79252.127874	

Както сами може да видите на снимката по - горе, това е статистиката, която програмата връща за последните 4 месеца (от 1 януари 2023, до 1 май 2023). Бота е High Frequency, което означава, че много често купува и продава. На снимката, началният капитал е \$10 000, като програмата търгува “BTC USDТ”, а диапазона в който тъгува е 5 минути.

```
import pandas as pd
import matplotlib.pyplot as plt
from binance.client import Client
import numpy as np
symbol="BTCUSDТ"
timeframe="5m"
starting_date="1 january 2023"

#get the data
info = Client().get_historical_klines(symbol,timeframe,starting_date)
dataframe_df = pd.DataFrame(info, columns=['timestamp', 'open', 'high', 'low',
'close', 'volume', 'close_time', 'quote_av', 'trades', 'tb_base_av',
'tb_quote_av', 'ignore'])

data=dataframe_df.copy() #make a copy of the data

#leave only ohlcv columns in the data
data.drop(columns=data.columns.difference(['timestamp', 'open', 'high', 'low',
'close', 'volume']), inplace=True)

#make time in milliseconds
data.set_index(data['timestamp'], inplace=True)
data.index=pd.to_datetime(data.index, unit='ms')
del data['timestamp']

data['open'] = pd.to_numeric(data['open'])
data['high'] = pd.to_numeric(data['high'])
data['low'] = pd.to_numeric(data['low'])
data['close'] = pd.to_numeric(data['close'])

ema12 = data['close'].ewm(span=12, adjust=False).mean()
ema26 = data['close'].ewm(span=26, adjust=False).mean()
macd = ema12 - ema26
macdsignal = macd.ewm(span=9, adjust=False).mean()
macdhist = macd - macdsignal
```

Програмата започва с имплементацията на нужните за добиване на информацията библиотеки, като Pandas, която служи за манипулиране на данни, Matplotlib за чертане на диаграми, Numpy за математически операции и Binance API клиент за извличане на исторически данни. След това кодът използва Client на Binance API - а (API е начина по който програмата и отсрещната страна си комуникират, в случай как програмата и ексчейнджа си комуникират). Това се случва, за да се извлекат историческите данни на съответната валута, от съответната дата. След това, тези данни са преобразувани в pandas DataFrame, за да може да се работи с тях. Правим копие на датата, за да не едитваме оригиналната дата, като след това запазваме само стойностите, които ще ни трябват за изчисленията и работата на програмата, като ги превръщаме в числа, тий като те ни биват подадени като string. След това изчисляваме MACD индикатора, както и Signal Line - а. В следващата част от кода записвам променливи, които ще следят случващото се на пазара, като ги използваме и за статистика, накрая на тестовия период:

```
##### Wallet

wallet = 10000
trade=[] #keep track of the trade

##### Profit/Loss
profit = 0
loss = 0

profit_ratio = 2.5
percentage_of_stop_loss= 0.05

total_p=0 #total_profit
total_l=0 #total_loss

##### Keeping track of the asset
in_position = False # checks if a purchase is being made

btc_bought = 0
purchase_price = 0
highest_candle_price=0
num_buys = 0

count_wins=0 #total_wins
count_loss=0 #total_loss
```

В следващата част от кода се случва калкулацията на Supertrend индикатора:

Този код е взет от [Part Time Larry](#)

```
def Supertrend(data, atr_period, multiplier):

    high = data['high']
    low = data['low']
    close = data['close']

    # calculate ATR
    price_diffs = [high - low,
                   high - close.shift(),
                   close.shift() - low]
    true_range = pd.concat(price_diffs, axis=1)
    true_range = true_range.abs().max(axis=1)
    # default ATR calculation in supertrend indicator
    atr = true_range.ewm(alpha=1/atr_period, min_periods=atr_period).mean()
    # data['atr'] = data['tr'].rolling(atr_period).mean()

    # HL2 is simply the average of high and low prices
    hl2 = (high + low) / 2
    # upperband and lowerband calculation
    # notice that final bands are set to be equal to the respective bands
    final_upperband = upperband = hl2 + (multiplier * atr)
    final_lowerband = lowerband = hl2 - (multiplier * atr)

    # initialize Supertrend column to True
    supertrend = [True] * len(data)

    for i in range(1, len(data.index)):
        curr, prev = i, i-1

        # if current close price crosses above upperband
        if close[curr] > final_upperband[prev]:
            supertrend[curr] = True

        # if current close price crosses below lowerband
        elif close[curr] < final_lowerband[prev]:
            supertrend[curr] = False

        # else, the trend continues
```

```
    else:
        supertrend[curr] = supertrend[prev]

        # adjustment to the final bands
        if supertrend[curr] == True and final_lowerband[curr] <
final_lowerband[prev]:
            final_lowerband[curr] = final_lowerband[prev]
        if supertrend[curr] == False and final_upperband[curr] >
final_upperband[prev]:
            final_upperband[curr] = final_upperband[prev]

        # to remove bands according to the trend direction
        if supertrend[curr] == True:
            final_upperband[curr] = np.nan
        else:
            final_lowerband[curr] = np.nan

    return pd.DataFrame({
        'Supertrend': supertrend,
        'Final Lowerband': final_lowerband,
        'Final Upperband': final_upperband
    }, index=data.index),

atr_period = 10
atr_multiplier = 1

supertrend, = Supertrend(data, atr_period, atr_multiplier,)
data = data.join(supertrend)
```

Във функцията Supertrend задаваме базата данни data, в която по - рано сложихме данните, нужни за изчисляването на стратегията, както и average true range (ATR) периода, заедно със multiplier-a.

Вътре във функцията извличаме high, low и close цените от DataFrame. След това във функцията изчисляваме true range - а (TR), като използваме трите възможни ценови разлики, изчисляваме ATR от TR, използвайки exponential moving average (EMA) и изчисляваме цената HL2 ((high - low) / 2) (средната стойност на високите и ниските цени).

След това задаваме началните горна и долна линия съответно, като final\_upperband и final\_lowerband и изчислени от цената HL2 и ATR, умножени по множител. В началото задаваме колоната на Supertrend - а, във всички редове от DataFrame - а да е True. След това въртим for цикъл, в който проверяваме close



цената на валутата в съответния ден. Ако цената е под `final_lowrband` - а , задаваме стойността на Supertrend - а, да е False, а ако е над `final_upperband`, стойността остава True. След това проверяваме, ако Supertrend - а е True и `final_lowerband` - а за текущия ред е по-малък от предишния `final_lowerband`, `final_lowerband` - а се задава да бъде равен на предишния `final_lowerband`. По същия начин, ако Supertrend - а е False и `final_upperband` за текущия ред е по-голям от предишния `final_upperband`, `final_upperband` - а се задава равен на предишния `final_upperband`. Това се прави, за да може ако има несъответствие в дирекцията на тренда и съответно горните и долни линии, тя да бъде поправена.

```
for i in range(len(data)):
    buy_percent_of_trade = 0.3 * wallet
    trend_up=False
    trend_down=False

    if data['Supertrend'][i] == True:
        trend_up=True
    else:
        trend_down=True

    if data['high'][i] > highest_candle_price:
        highest_candle_price = data['high'][i]

    if macd[i] > macdsignal[i] and macd[i-1] < macdsignal[i-1] :

        if trend_up and not in_position:
            purchase_price = data['close'][i]

            target_price = (1 + (profit_ratio/100)) * purchase_price

            purchase_amount = buy_percent_of_trade/purchase_price

            btc_bought += purchase_amount

            wallet -= (purchase_amount * purchase_price)

            print(f"Purchased {purchase_amount} BTC at
{purchase_price:.2f} USDT each")
            trade.append({'date':data.index[i], 'side':'buy', 'price':
purchase_price, 'amount': purchase_amount,
'usdt':purchase_amount*purchase_price, 'wallet':wallet})
```

```
num_buys+=1
in_position = True
highest_candle_price=0

if wallet != 0 and trend_up and num_buys<2:

    purchase_amount = buy_percent_of_trade/purchase_price

    btc_bought += purchase_amount

    wallet -= (purchase_amount * purchase_price)

    print(f"Purchased {purchase_amount} BTC at
{purchase_price:.2f} USDT each")
    trade.append({'date':data.index[i], 'side':'buy',
'price': purchase_price, 'amount': purchase_amount,
'usdt':purchase_amount*purchase_price, 'wallet':wallet})

    num_buys+=1
    in_position = True
    highest_candle_price=0

    stop_loss_price = highest_candle_price - (percentage_of_stop_loss *
highest_candle_price)
```

Кодът по - горе започва със фор цикъл, който обхожда всички данни, които програмата е събрала по - рано. По време на всяка итерация кодът първо задава процента от портфейла, който да се използва за закупуване на актив, и инициализира две променливи (boolean), зададени на False, като те са отговорни за следенето на това, дали тренда е uptrend или downtrend.

След това кодът проверява дали индикаторът Supertrend е True или False. Ако е True, променливата trend\_up е зададена на True, в противен случай trend\_down е True. Променливата high\_candle\_price се актуализира, ако високата цена на текущата свещ е по-висока от предишната най-висока цена, като накрая на кода, след като сме продали закупената валута, тази променлива става 0, съответно тя се променя със всяка покупка на актива.

След това кодът проверява дали MACD линията е пресякла Signal линията. MACD индикатора е предназначен да се ползва по този начин, но линиите трябва да се пресичат под нулевата линия, което подsigурява набирането на momentum от тренда, но аз го направих, да няма значение дали са под нулевата линия, тъй като така бота изкарва повече профит. Ако това условие е изпълнено и trend\_up също е True, бота купува един актив на съответния процент от портфейла. Кодът също така изчислява target price, който е 2.5% над цената на съответно закупеният актив и проследява цената на която е закупен, количеството на покупката и оставащата сума в портфейла.

Кодът има и допълнителна проверка за закупуване на актив, ако портфейлът не е празен и текущия trend е възходяща и преди това са направени по-малко от две покупки. Тоест максималният брой на закупения актив е не повече от 2.

И накрая, кодът задава stop\_loss\_price за актива, която е 0.05 процента под най-високата цена по време на текущия тренд.

```
elif in_position and ((df['high'][i] >= target_price) or (df['low'][i] <=
stop_loss_price) or df['Supertrend'][i]==False):
    sell_price = df['high'][i]
    btc_sold = btc_bought
    btc_bought = 0
    if df['Supertrend'][i]==False and in_position:
        if df['high'][i]>purchase_price:

            profit = btc_sold * (sell_price - purchase_price)
            wallet += ((btc_sold*sell_price) )
            total_p += profit
            sell_signal.append(i)

            print(f"Sold {btc_sold:.8f} BTC at {sell_price:.2f} USDT for
a profit of: {profit:.2f}. - Supertrend Succeeded ")
            trade.append({'date':df.index[i], 'side':'sell', 'price':
sell_price, 'amount': num_buys*purchase_amount, 'usdt':btc_sold*sell_price,
'wallet':wallet})
        else:

            loss = btc_sold * (purchase_price - sell_price)
            wallet += ((btc_sold*sell_price) )
            total_l -= loss
            sell_signal.append(i)

            print(f"Sold {btc_sold:.8f} BTC at {sell_price:.2f} USDT for a
```

```
loss of {loss:.2f}. - Supertrend failed")
        trade.append({'date':df.index[i], 'side':'sell', 'price':
sell_price, 'amount': num_buys*purchase_amount, 'usdt':btc_sold*sell_price,
'wallet':wallet})
```

```
if data['high'][i] >= target_price:
    profit = btc_sold * (sell_price - purchase_price)
    wallet += (btc_sold*sell_price )
    total_p += profit

    count_wins+=1

    print(f"Sold {btc_sold:.8f} BTC at {sell_price:.2f} USDT for a
profit of: {profit:.2f}. - Reached 1.5 profit")
    trade.append({'date':data.index[i], 'side':'sell', 'price':
sell_price, 'amount': num_buys*purchase_amount, 'usdt':btc_sold*sell_price,
'wallet':wallet})

    elif data['high'][i]<stop_loss_price:
        loss = btc_sold * (purchase_price - sell_price)
        wallet += (btc_sold*sell_price )

        total_l -= loss

        count_loss+=1

        print(f"Sold {btc_sold:.8f} BTC at {sell_price:.2f} USDT for a
loss of {loss:.2f}. - Activated Stop Loss")
        trade.append({'date':data.index[i], 'side':'sell', 'price':
sell_price, 'amount': num_buys*purchase_amount, 'usdt':btc_sold*sell_price,
'wallet':wallet})

    in_position = False

    num_buys=0

    print(f'THE HIGHEST CANDLE WAS: {highest_candle_price}')
    print(' ')
    last_purchase_price = 0
    #highest_candle_price=0
```

В кода по - горе е процеса, извършващ се при продажба на вече закупеният актив. Програмата проверява, дали `in_position = True` (дали е закупен актив), ако това условие е вярно, програмата проверява, дали цената е стигнала `target_price` - а, дали е ударила `stop_loss_price` - а, или дали Supertrend индикатора е станал False. Съответно в кода по - горе е описано какво се случва, ако някое от тези условия се случи. Да вземем за пример, ако Supertrend индикатора върне стойността False. В такъв случай проверяваме, дали цената на която е върнал тази стойност е по - голяма от тази, на закупената цена. Ако това се окаже истина, програмата е изкарал профит, в противен случай, програмата е загубила пари. После цените, печалбите и загубите се записват в `trade` масив, който да държи стойностите, както и се принтира хипотетичната покупка. Ако цената на закупеният актив удари `target_price` - а, или `stop_loss` - а, се случва същото нещо, като в края на кода, след продаването на актива, броят на закупените активи, както и `highest_candle_price` стават 0, за да могат да се преизчисляват след това, както и `in_position` става False, разбира се.

## Свързване на програмата с real time пазара

Работата на бота на истинския пазар е малко по - сложна. В момента, все още бота се тества на истинския пазар и не мога да кажа с ръка на сърцето, че ще успее толкова добре, колкото се справи на миналото на пазара. В кода по - долу ще видите единствената разлика между двата кода, и тя е, че във кода, свързващ програмата с бота ползвам `websocket`.

WebSocket е комуникационен протокол, който позволява обмен на данни в реално време между уеб браузър и сървър чрез една дълготрайна връзка. Клиентът (уеб браузър, в нашия случай, програмата) изпраща HTTP заявка до сървъра, искайки да установи WebSocket връзка. Сървърът отговаря с HTTP отговор, показващ дали WebSocket връзката е установена или не.

Ако връзката е установена, както клиентът, така и сървърът могат да изпращат данни един на друг по всяко време, без да е необходимо да изпращат нова HTTP заявка или отговор. Това се нарича двупосочна комуникация (за пример, връзката между два телефона, когато говорите с някого, работи по същия начин, с двупосочна комуникация). Данните, изпратени между клиента и сървъра, могат да бъдат във всякакъв формат (текст, JSON, байнъри и т.н.) и могат да бъдат изпратени веднага щом са достъпни, без да е необходимо да чакате първо да бъдат изпратени други данни. Когато клиентът или

сървърът искат да затворят връзката, те могат да изпратят съобщение, което показва, че връзката трябва да бъде затворена. WebSockets обикновено се използват за приложения в реално време като онлайн игри, приложения за чат и ботове за търгуване, където бързото и надеждно предаване на данни е важно.

```
last_row = data.iloc[-1]
SOCKET =
f"wss://stream.binance.com:9443/ws/{symbol_for_websocket}@kline_{interval}"

def on_open(ws):
    print('opened connection')

def on_close(ws):
    print('closed connection')

def on_message(ws, message):
    global last_row, data, supertrend, atr_multiplier, atr_period, macd_data
    global highest_candle_price, buy_percentage_of_trade, balances
    global trend_up, trend_down, stop_loss
    global profit_ratio, num_buys, btc_bought
    global profit, loss, count_loss, count_wins, target_price, purchase_price,
    quantity

    response = requests.get(url)
    data_for_price = response.json()
    price = float(data_for_price["price"])
    # print(f"Current BTCTUSD price: {price}")

    json_message = json.loads(message)
    kline = json_message['k']

    last_row.name = pd.Timestamp(int(kline['t']), unit='ms')
    last_row['open'] = float(kline['o'])
    last_row['high'] = float(kline['h'])
    last_row['low'] = float(kline['l'])
    last_row['close'] = float(kline['c'])

    # Update the last row of the dataframe with the new data
    data.iloc[-1] = last_row

    supertrend = Supertrend(data, atr_period, atr_multiplier)
```

```
macd_data = calculate_macd(data)
if kline['x'] :
    # Create a new row with the kline stats
    new_row = pd.Series({
        'open': float(kline['o']),
        'high': float(kline['h']),
        'low': float(kline['l']),
        'close': float(kline['c'])
    }, name = pd.Timestamp(int(kline['T']), unit='ms').round('S') )

    # Concatenate the new row with the existing dataframe
    data = pd.concat([data, new_row.to_frame().T])

    supertrend = Supertrend(data, atr_period, atr_multiplier)
    macd_data = calculate_macd(data)

if highest_candle_price < last_row['high']:
    highest_candle_price = last_row['high']

trend_up=False
trend_down=False

# print(data,supertrend,macd_data,highest_candle_price)

total_balance = 0.0

for balance in balances:
    asset = balance['asset']
    free = float(balance['free'])
    locked = float(balance['locked'])
    if free + locked > 0:
        if asset == 'USDT':
            total_balance += free + locked
        else:
            symbol = asset + 'USDT'
            ticker = client.get_symbol_ticker(symbol=symbol)
            price = float(ticker['price'])
            total_balance += (free + locked) * price

ws = websocket.WebSocketApp(SOCKET, on_open=on_open, on_close=on_close,
on_message=on_message)
ws.run_forever()
```

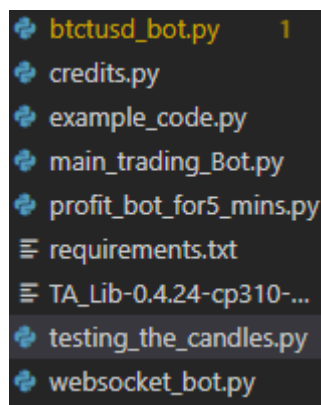
В кода по - горе има няколко основни неща, но другата част, която не съм качил, е като от предходния код(за тестването на бота в миналото на пазара). В началото, след като установя връзка с Binance, ъпдейтвам последният взет кендъл от пазара, тъй като в момента на свързването, ако кендъла не е затворен, информацията бива взета недовършена. Следователно в Websocket - а ъпдейтвам последния ред кендъл и от там нататък, когато `kline['x'] = True`, това означава, че кендъла е затворил, следователно го добавяме в DataFrame - а, като по - надолу взимам цената на актива. В края на кода пиша `ws.run_forever()`, за да може програмата да спре да работи, само когато я затворим

## Внедряване (deployment)

<https://github.com/GeorgiKirov04/Algorithmic-Crypto-Trading-Bot>

## Глава 4. Ръководство за потребителя (смени заглавието)

*Когато изтеглите файловете, отваряте папката, носеща името "Final\_Bot\_Binance". Там ще намерите няколко файла:*



Основният файл ще бъде `btctusd_bot.py`. Когато го отворите, нагласяте символа, както и интервала от време за търговия (съветвам да не променяте нищо, защото това е само прототип. Този бот ще бъде разработен много по - добре и с много повече функции). Отваряте файла, след което натискате **Ctrl + f5** и воала, имате бот, търгуващ крипто (не нося отговорност за загубите, които бота може да допусне)



## Заклучение

*В заключение мога само да кажа, че този проект беше много предизвикателен, а на много моменти си помислих, че няма да се получи и бях готов да се откажа. Този бот е само началото на нещо по-голямо, нещо по-добро, нещо революционно в моя живот.*

## Информационни източници

▶ [Supertrend Part 1 - Average True Range](#)

▶ [Supertrend Part 2 - Basic Bands in Python](#)

▶ [Supertrend 3 - Detecting Uptrends and Downtrends with Python](#)

▶ [Supertrend 4 - CCXT Crypto Bot - Order Execution with Python](#)

▶ [Building My First Python Crypto Trading Bot w/ Binance API](#)

<https://www.youtube.com/@moondevonyt>

<https://youtube.com/playlist?list=PLdtqDMvm2lvRwcfbzb2BXapv5ShX-RK-M> (първите 4 видеа)

▶ [Binance API Key - How to create & Security Guide](#)

▶ [BEST MACD Trading Strategy \[86% Win Rate\]](#)

▶ [How To Identify Trends in Markets \(Never Guess Again\)](#)

▶ [Introduction To Automated Trading With Python and Binance - REAL money Trades Of Cryptocurrencies](#)

▶ [Build a Real-Time Crypto Trading Bot in under 100 Lines of Code](#)

<https://github.com/hackingthemarkets/binance-tutorials/tree/master/rsibot>

<https://github.com/hackingthemarkets/supertrend-crypto-bot>

<https://coinmarketcap.com/alexandria/glossary>

<https://openai.com/blog/chatgpt>

<https://www.bing.com>

[https://binance-docs.github.io/apidocs/spot/en/#query-margin-asset-market\\_data](https://binance-docs.github.io/apidocs/spot/en/#query-margin-asset-market_data)

*И още много видеа, но това са основните*

## Рецензия на дипломен проект

Тема на дипломния проект		
Ученик		
Клас		
Професия		
Специалност		
Ръководител- консултант		
Рецензент		
Критерии за допускане до защита на дипломен проект	Да	Не
Съответствие на съдържанието и точките от заданието		
Съответствие между тема и съдържание		
Спазване на препоръчителния обем на дипломния проект		
Спазване на изискванията за оформление на дипломния проект		
Готовност за защита на дипломния проект		
Силни страни на дипломния проект		
Допуснати основни слабости		
Въпроси и препоръки към дипломния проект		

## ЗАКЛЮЧЕНИЕ:

Качествата на дипломния проект дават основание ученикът/ ученичката .....  
да бъде допуснат/а до защита пред членовете на комисията за подготовка, провеждане и оценяване на  
изпит чрез защита на дипломен проект- част по теория на професията.

.....05.2023г.

Рецензент:.....

град София