

변동성 돌파 전략을 활용한 가상화폐 거래 자동화 GUI 개발 과정

지도교수 : 한 상 훈

연구자 : 고 봉 성

< 목 차 >

1. 서론

2. 관련연구

2.1. API

2.1.1. API 개념과 작동방식

2.1.2. API 활용사례

2.2. 변동성 돌파 전략

2.2.1. 변동성 돌파 전략 개념

2.2.2. 변동성 돌파 전략 적용 사례

2.3. 가상화폐

2.3.1. 가상화폐 개념과 용어

2.3.2. 가상화폐와 주식의 차이

3. 성인콘텐츠

3.1. 가상화폐 거래소 선택

3.2. 가상화폐 거래소 가입

3.3. API 발급

3.4. 개발 환경 세팅 후 잔고 조회

3.5. 코드구현

4. 결론 및 향후과제

요 약

가상화폐의 등장에 따라 전통적인 자산을 추구하는 것이 안전하다는 견해도 있고, 가상화폐로 인해 큰 변화를 기대하기도 하는 입장의 전문가들이 있다. 본 논문은 주식시장과 다르게 24시간 운영을 하고, 변동 폭이 큰 가상화폐 거래를 하는 사람들이 하루 종일 핸드폰을 붙잡고 일상생활을 보내는 것을 줄여야 할 필요가 있다고 사료되어 스트레스를 덜 받고 시간을 절약하고자 본 논문을 쓰게 되었다.

주요어 : 가상화폐, API, 변동성 돌파 전략

1. 서론

내가 이 주제에 대해 논문을 쓰는 이유는 개인적인 이유도 있었다. 평소에 주식과 가상화폐를 하면서 핸드폰만 보고 있는 내 모습을 보고도 너무 지치면서 손해를 최대한 줄일 수 있고, 몸을 편하게 만들어주는 툴이 있었으면 좋겠다는 생각이 틈틈이 났다. 프로그래밍에 관심이 많은 학생으로써 한번 어떤 방식으로 제작을 할 수 있나 알아보던 도중 생각보다 복잡하고 여러 가지 과정이 필요한 것을 알게 되었다. API와 가상화폐, 파이썬에 대해서도 내가 알던 것보다 더 자세히 알아야 하고, 여러 가지 많은 지식이 있어야 한다는 것을 알았다. 생각보다 간단하지 않고 관심이 있는 분야인 만큼 이 프로그램을 만드는 과정을 논문으로 제작하면서 같이 해보면 어떨까 생각을 하게 되면서 이 주제에 대해 논문을 제작하게 되었다.

따라서, 본 논문에서는 변동성 돌파 전략을 바탕으로 가상화폐의 자동 거래를 위한 자동화 툴 과정을 제안하고자 한다. 이 제안으로 가상화폐 거래에서의 개인적인 손실이 크지 않기를 바라본다.

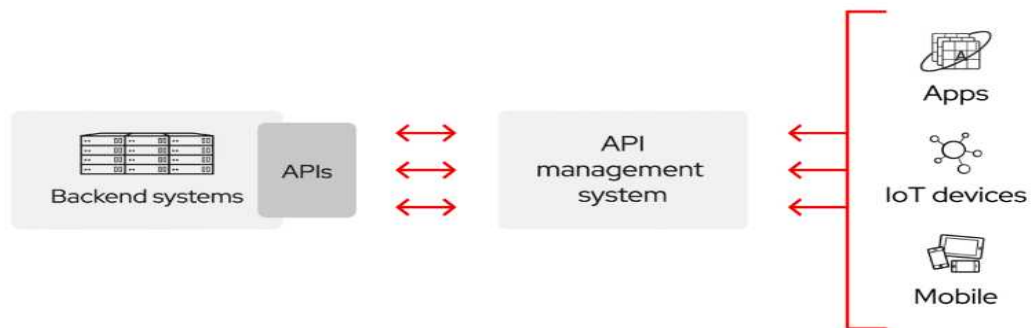
2. 관련연구

본 논문에서 제안하는 자동화 거래 툴을 위한 관련 연구로 오픈API, 실물 없이 디지털 상에서만 거래되는 가상화폐, 가상화폐의 가치 변화에 따른 거래 전략인 변동성 돌파 전략을 먼저 살펴보고자 한다.

2.1 API

2.1.1 API 개념과 작동방식

API는 키보드와 유사하다고 할 수 있다. 키보드를 통해 우린 컴퓨터와 대화한다. 키보드의 버튼을 누르면 키보드에서 컴퓨터로 전달되고 스크린에서 텍스트를 볼 수 있다. 키보드를 통해서 컴퓨터와 상호작용 하는 것이다. 이와 같은 맥락으로 API는 프로그램과 프로그램이 서로 상호작용하는 것을 도와주는 매개체이다. 코드들끼리 서로 소통하기 위해 만들어진 게 API이다.



[사진 1] API 구조

그림에 보이는 것처럼 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다. 주로 파일 제어, 창 제

어, 화상 처리, 문자 제어 등을 위한 인터페이스를 제공한다.

API를 통해 조직은 자체 소프트웨어 및 데이터 사용을 소유하지 않는 애플리케이션과 데이터로 확장할 수 있다. 이를 통해 기능 향상, 비용 절감, 기술 혁신, 그리고 운영 간소화 등의 다양한 이점을 누릴 수 있다.

2.1.2 API 활용사례

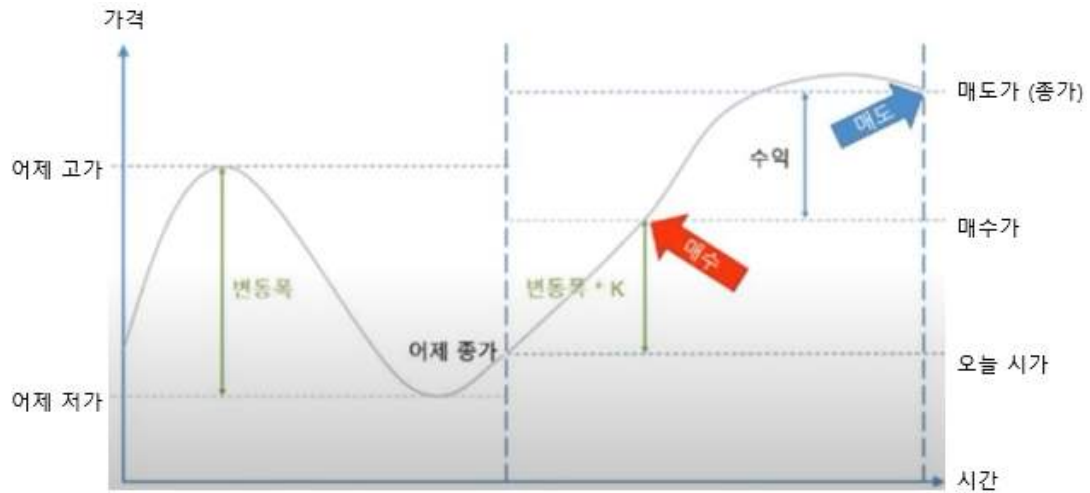
오픈API는 일반적으로 여러 사람들이 공동적으로 사용할 필요가 있는 데이터에 대한 사용을 개방하고 사용자가 해당 데이터에 대한 전문적인 지식이 없어도 쉽게 가공하여 사용할 수 있도록 데이터를 추상화하여 표준화한 인터페이스이다. 일반적인 웹 검색에 대한 결과는 그 결과에 대한 내용만을 인식할 수 있지만 오픈 API는 그 결과를 가공하여 자신만의 프로그램개발에 사용할 수 있다. 오픈 API를 통해 받은 XML이나 JSON 형태의 데이터를 피싱하여 자신이 구현하고자 하는 프로그램에 맞게 사용이 가능하다.

오픈API 제공 기업 및 공공기관 서비스 사례로는 Google에서 광고를 위한 AdSense API, 지도 서비스를 위한 GoogleMaps API를 제공하여 친구 찾기, 부동산정보 등 300여 개의 신규서비스를 창출했고, Amazon에서는 이미 10년 전부터 개발자 14만 명이 아마존의 오픈 API를 이용해 상품에 대한 정보를 핸들링하고 있으며 지금도 여러 가지 사업으로 오픈 API를 활용하고 있다. eBay는 3만 명이 넘는 eBay 개발자들이 월간 30억이 넘는 API를 호출하면서, 47% 이상의 상품이 API를 통해 등록되고 있다. 공공데이터포털(data.go.kr)은 국민의 선택으로 선정된 ‘국가중점개방데이터’ 36대 분야를 대용량 데이터로 개방했고, 브이월드(vworld.kr)에서는 공간정보에 대한 데이터를 2D, 3D 데이터로 오픈 API를 통해 제공함과 동시에 원시데이터도 함께 제공하고 있다.

2.2 변동성 돌파 전략

2.2.1 변동성 돌파 전략 개념

변동성 돌파 전략이란 “래리 윌리엄스”가 소개한 전략으로써, 변동성이 높은 시장에서 많이 유효하다. 그가 주로 사용한 전략인 변동성 돌파 전략의 원리는 아주 심플하다. 어제 의 기준으로 가장 낮은 가격인 저가와 가장 높은 가격인 고가의 변동 폭을 기준으로 오늘의 시작 가격의 그 변동 폭의 k배만큼 상승을 하게 되었을 때, 그때를 돌파 신호로 판단하고 이때 매수를 진행하게 된다. 그리고 그날 종가에 매도를 하는 전략이다. 진입 타이밍을 식으로 만들어보면 [현재가 > 당일 시가 + (전일 고가 - 전일 저가) * k(k값은 보통 0.4~0.6)] 이다. 하지만 이 전략을 믿고 한다면 MDD(최대 낙폭)가 너무 크기 때문에 반드시 자금관리 방식도 병행되어야 한다. 알려진 변동성 돌파 전략이 이런 방식이라 할 뿐이지 실제로 래리는 자금관리를 아주 강조했다.



[사진 2] 변동성 돌파 전략

정리하면, 전략이 굉장히 심플하지만 강력하다. 오늘 가격이 돌파 가격을 넘어가면 매수하고, 다음날 아침 장이 시작하면 모두 매도한다. 당일 매수해서 다음날 팔기 때문에 시장 심리에 휘둘리지 않고, 전략이 간단해서 자동매매 구현에 쉽다.

2.2.2 변동성 돌파 전략 적용 사례

“래리 윌리엄스”는 선물투자대회에서 1987년 11.376%의 수익률을 올리면서 1년 동안 1만 달러를 111만 달러로 만들었고, 그의 딸 “미셸 윌리엄스”도 1997년 16살의 나이에 래리로 부터 배운 매매기법으로 같은 대회에서 1000%의 수익률을 올리며 우승했다. 단순하고 강한 전략이라 오늘날에도 많은 사람들이 이 전략을 적극 활용하고 있다. 무작정 이 전략을 따라한다기보다 자기 자신의 전략에 변동성 돌파 전략을 활용하여 수익을 내는 사람들이 많다. 주변 사람들이나 주식시장만 봐도 증가와 시가를 잘 살피면서 단순하게 시장 심리에 휘둘리지 않고 마음 편하게 하는 사람이 많다.

2.3 가상화폐

2.3.1 가상화폐 개념

가상화폐는 컴퓨터상에 기록되어 실물 없이 디지털 상에서만 거래되는 화폐이다. 유럽 중앙은행이 2012년에 가상화폐에 대해 정의한 내용을 차용하면, 가상화폐란 “정부에 의해 통제받거나 법적 규제를 받지 않는 디지털 화폐로, 가상화폐 발행자가 발행·관리하고, 특정 커뮤니티의 구성원들 사이에서 이용되는 화폐”이다. 가상화폐와 디지털화폐, 암호화폐가 동일한 의미로 사용되는 경우가 많은데, 엄밀히 따지면 완전히 다른 개념이다. 대부분 우리가 알고 있는 가상화폐들은 보통 암호 화폐라고 불리지만, 정작 가상화폐라고는 잘 불리지 않는다. 외국에서도 virtual currency 라는 용어보다는 crypto currency 라는 용어를 더 많이 쓴다. 하지만 굳이 관계를 따지자면 암호화폐가 가상화폐에 포함되고, 가상화폐가 디지털화폐에 포함된다. 결국, 디지털화폐가 가장 크고 보편적인 개념이라 볼 수 있다.

용어	설명
채굴	암호화폐의 거래 내역을 기록한 블록을 생성하여 그 대가로 코인을 얻는 행위
반감기	채굴에 성공한 보상으로 지급되는 코인의 양을 절반으로 줄어드는 시기
디파이	중앙정부의 통제를 받지않는 탈중앙화 금융
백서	새로운 프로젝트나 사업에 대한 설명서이자 로드맵, 생성원리나 수익구조, 향후 비전, 제작사 및 투자사 등 다양한 정보를 포함함
스테이블코인	법정 화폐인 미국 달러와 1대 1로 교환이 가능한 코인
스테이킹	코인을 예치해서 이자를 받는 행위
스왑	각각 다른 블록체인 기반의 코인을 서로 교환하는 행위
NFT	대체 불가능한 토큰이라는 뜻으로, 디지털자산에 부여하는 일종의 정품인증서
민팅	디지털자산에 NFT를 부여하는 행위
에어드랍	특정 코인을 보유한 사람에게 이벤트성으로 코인을 무상 지급하는 행위
익절	이익을 보고도 매도를 하는 행위
단타	단기적으로 투자를 하고 매도하는 행위
장기투자	투자를 하고 장기적으로 기다리는 행위
변동성 돌파전략	[현재가 > 당일 시가 + (전일 고가 - 전일 저가) * k]
MDD	최대 낙폭 ((고점가격-저점가격)/고점가격)

[표1] 가상화폐에 관련된 용어들

2.3.2 가상화폐와 주식의 차이

가상화폐와 주식은 기본 원리로는 동일하다고 볼 수 있다. 주식시장이 있고 주가가 상승하고 하락하며, 각종 종목을 매매하여 차트로 흐름을 분석하는 것은 동일하다. 그러나 주식을 잘 한다고 해서 바로 가상화폐로 뛰어들어 큰 수익을 얻는 경우는 드물다. 그리고 주식을 못 한다고 해서 가상화폐 투자를 시작할 수 없는 것도 아니다. 차이점에는 다음과 같은 것들이 있다.

1) 거래소 운영시간

주식은 오전 09:30분 ~ 오후 03:30분이고 주말과 공휴일에는 거래가 불가능하다. 가상화폐는 24시간 365일 가능하다. 다만 언제나 거래가 가능하다는 것은 수익의 기회가 넘쳐난다고 볼 수 있겠지만, 아주 단편적인 부분이며 대부분은 일할 때도 밥 먹을 때도 코인에만 신경을 쓰게 되는 부작용이 훨씬 더 크다. 이를 예방하기 위해 보통 예약매수나, 예약매도 같은 기능들을 적극 활용하고, 단타 대신 중장기 투자를 바라보기도 한다.

2) 세금

주식은 배당 소득세나 양도소득세, 증권거래세 등이 있고 가상화폐는 세금을 내지 않는다. 해외의 경우 가상화폐에 대한 세금 비율이 많이 높은 편이다. 우리나라도 일정 이상에 해당하는 수익에 대해 n%의 양도세를 매기려고 하는 하지만 아직까진 적용하지 못하고 있다.

3) 상한가, 하한가

주식시장은 정해진 시간이 있기 때문에, 주식이 폭등하거나 폭락을 하더라도 최대 30%이다. 그러나 가상화폐는 그런 제한이 없다. 이 또한 양날의 검이기 때문에 가상화폐는 폭락을 하더라도 30%이상도 폭락을 할 수 있다. 따라서 단기투자라고 하면 5% 이상 10% 이상은 익절(이익을 보고도 매도)하는 것을 권장하고 있다.

3. 제안기법

3.1 가상화폐 거래소 선택

자동매매 프로그램을 만들기 전에 API를 발급받을 가상화폐 거래소를 선택해야 한다. 자동매매 프로그램으로 거래를 한다는 조건으로 생각했을 때, 최대한 거래 수수료가 저렴한 곳을 선택해야 한다. API를 제공하는 대표적인 거래소 중에 빗썸 0.15%, 코빗 0.15%, 비아낸스 0.1%, 업비트 0.05%로 수수료가 가장 싼 곳은 업비트이다. 그래서 나는 업비트로 선택을 하고 업비트 API를 사용하기로 했다.

3.2 가상화폐 거래소 가입

먼저 API를 발급받으려면 가상화폐 거래소에 가입을 해야 한다. 3.1에서 선택한 가상화폐 거래소인 업비트에 가입을 한다. 홈페이지에서 가입해도 되지만 우리는 앱을 통해서 알림을 받을 것이기 때문에 앱을 설치하고 앱에서 가입을 진행한다. 앱스토어나 구글 플레이스토어에서 업비트 앱을 다운로드한다. 앱을 실행하면 업비트 로그인, 애플 ID로 로그인, 그리고 카카오 계정으로 간편 로그인이 있다. 나중에 간편 인증을 생각했을 때, 카카오 로그인이 좋았지만 지난 10월 15일 발생한 카카오톡 먹통 사태로 인해 많은 업비트 회원들이 카카오톡을 활용한 로그인 방식을 제때 이용하지 못하면서 불편을 겪은 이후로, 업비트는 보안강화 등의 이유를 들며 로그인 방식을 변경했다. 우선 애플리케이션을 업데이트 한 뒤 실명계좌 확인을 해야 한다. 이후 6자리의 핀 번호나 생체 인증을 등록하면 QR코드를 통한 로그인을 할 수 있다. 로그인 후 닉네임을 설정하고, 고객확인 절차를 한 후 보안 등급을 설정해야 한다. 업비트의 보안 등급은 총 5개의 레벨로 구성되어 있는데 이 중에 4레벨까지 인증을 해야 원화 출금이 가능하기 때문에 레벨 4까지 인증을 해주어야 한다. 레벨 1의 이메일 인증이나 레벨 2의 휴대폰 인증은 그냥 인증하면 되는데 레벨 3 입출금계좌 인증은 계좌가 반

드시 케이뱅크 계좌와만 연동이 가능하다. 케이뱅크 계좌가 없는 사람들은 계좌를 개설해야 한다. 그 후 레벨 4는 카카오페이 간편 인증으로 편하게 인증할 수 있다. 보안등급 4레벨까지 인증을 한 사람들은 입출금 내역에 들어가 보면 계좌 개설을 한 케이뱅크 계좌에 있는 원화를 입출금하기가 가능한 것을 볼 수 있을 것이다.

3.3 API 발급

이번에는 자동매매에 사용될 API 키를 발급받을 것이다. 업비트 공식 홈페이지에 접속해서 로그인을 해준다. 그리고 고객센터로 들어가서 왼쪽에 오픈 API 안내로 들어간다. 그리고 오픈 API 사용하기 버튼을 눌러준다. 약관에 동의하면 오픈 API 키 관리 페이지가 나온다. 여기서 자산 조회, 주문 조회, 주문하기, 출금 조회, 출금하기, 입금조회, 입금하기 이렇게 7가지의 선택지가 있는데 이 중 처음 3가지인 자산조회, 주문조회, 주문하기를 체크해준다. 그리고 안전을 위해서 특정 IP에서만 실행을 체크해주고 내 컴퓨터에서만 실행이 될 수 있도록 자신의 아이피를 적어놓는다. 이 과정을 다 해주었으면 오픈 API 키 발급하기를 눌러준다. 카카오페이 인증을 해주면은 액세스 키와 시크릿 키를 발급해준다. 발급받은 키를 나중에 쓰기위해 편한 곳에 복사해놓는다.

3.4 개발 환경 세팅 후 잔고 조회하기

개발 환경은 비주얼 스튜디오 코드(vscode)와 파이썬을 이용한다. 비주얼 스튜디오와 파이썬을 다운로드해 준다. 파이썬 버전은 3점대 버전이면 괜찮고, 비주얼 스튜디오 코드는 파이썬 버전을 맞춰야 한다. 맞춰주지 않으면 진행이 되지 않는다. 파이썬을 다운로드 할 때 따로 파이썬 폴더를 만들고 다운로드 한다. 둘 다 다운로드를 했으면 vscode를 실행해주고 작업할 폴더를 선택해 준다. 왼쪽 상단 file에 오픈폴더를 눌러서 C드라이브에 작업할 새로운 폴더를 만들어준다. 새로운 폴더를 생성 후, vscode 안에서 방금만든 폴더에 새로운 파일을 생성해준다.(폴더명이나 파일명들은 자유롭게, 알아보기 쉽게 만들면 된다.) 그리고 vscode 위쪽에 terminal - new terminal을 눌러서 터미널 창을 실행시켜 준다. 터미널에서 python을 입력하여 파이썬 버전을 재 확인 후 vscode 왼쪽 하단에 보면 파이썬 버전이 설정되어있는데, 클릭 후 내가 다운로드받은 파이썬의 버전의 64비트를 선택해주면 파이썬 환경설정은 끝이 났다.

필요한 라이브러리들도 설치해 줄 것이다. 구글 검색창에 pyupbit를 검색하면 파이업비트 라이브러리 깃허브가 나와있다. pyupbit는 업비트 API를 편하게 사용하기 위한 라이브러리이다. 여기에 나와 있는 Installation에 따라서 설치를 진행해준다. 파이썬 개발환경으로 최신 버전의 Anaconda 를 설치해주는 pip install pyupbit를터미널창에 입력해서 엔터를 눌러 설치를 진행해 준다.

그리고 업비트 API를 활용해서 자신의 잔고를 조회하는 것을 해볼 것이다. 마찬가지로 깃허브에 pyupbit 문서로 들어가서 import 란에 있는 코드(import pyupbit)를 vscode에 입력해서 넣는다. 그러면 'pyupbit' 라이브러리를 가져오게 된다. 그리고 깃허브에서 스크롤을 내려 보면 로그인과 잔고 조회라는 코드가 나와 있다. 로그인 코드 import pyupbut를 아래에 입력하여 넣어준다. 시크릿 키 아래에 있는 upbit는 객체를 생성해 준 것인데 pyupbit에서 불러오도록 만들어준 것이다.

업비트 API를 활용해서 로그인을 하기 위해서 액세스 키와 시크릿 키는 자기 자신의 값으로 변경을 해줘야 한다. 전에 오픈 API 키를 발급받아서 나온 자신의 액세스 키와 시크릿 키를 대신 넣어준다.

```
import pyupbit

access = "73kVqowGQ0GEjdR31221j31j2ifekjkgjekgjekg"
secret = "egjekgj3iekeEEkej3i3j3iejjwiEejejeEeijg"
upbit = pyupbit.Upbit(access, secret)

print(upbit.get_balance("KRW-XRP"))      # KRW - XRP 조회
print(upbit.get_balance("KRW"))          # 보유 현금 조회
```

[사진 3] 로그인 코드와 잔고조회 코드

그리고 잔고 조회 코드도 가져와서 아래쪽에 붙여 넣어준다. get_balance 메서드는 입력 받은 티커의 보유 수량 정보를 조회한다.(티커는 암호 화폐 고유 이름을 식별하는 약어, 또는 심볼이다.)

비트코인 BTC/KRW	25,612,000	-0.43% -111,000	192,106백만
이더리움 ETH/KRW	1,380,000	-0.07% -1,000	152,961백만
리플 XRP/KRW	409	-1.21% -5.00	107,549백만
샌드박스 SAND/KRW	1,435	-1.37% -20.00	95,401백만

[사진 4] 가상화폐 종류와 티커

그리고 저장을 해준 후 아래 터미널 창에서 python test.py를 입력하고 실행시켜보면 값이 나온다. 이 값이 어떻게 나온 것이냐 하면 잔고 조회 코드 upbit.get_balance에 KRW를 하게 되면 보유하고 있는 현금이 조회되는 것이고, KRW 뒤에 원하는 코인의 티커를 입력하면 그 가격이 나오는 것이다. 즉 (그림 3)의 아래 두 줄은 업비트의 보유자산에서 XRP(리플)에 해당하는 양이 나오고 그 아래에는 남은 원화의 양이 나온다. 위 코드에서 XRP 대신 비트코인의 티커인 BTC를 붙여 넣으면 보유자산에서 비트코인(BTC)에 해당하는 가격이 나오는 것이다.

3.5 코드구현

코드가 어떤 식으로 짜였는지 설명을 해주는 것보다 코드 전체를 보여주고 코드 설명을 해주는 방향이 더 이해하기 쉬울 것 같아서 그렇게 하기로 했다.

```

01 import time
02 import pyupbit
03 import datetime
04
05 access = " "
06 secret = " "
07
08 def get_target_price(ticker, k):
09     """변동성 돌파 전략으로 매수 목표가 조회"""
10     df = pyupbit.get_ohlcv(ticker, interval="day", count=2)
11     target_price = df.iloc[0]['close'] + (df.iloc[0]['high'] - df.iloc[0]['low']) * k
12     return target_price
13
14 def get_start_time(ticker):
15     """시작 시간 조회"""
16     df = pyupbit.get_ohlcv(ticker, interval="day", count=1)
17     start_time = df.index[0]
18     return start_time
19
20 def get_balance(ticker):
21     try:
22         now = datetime.datetime.now()
23         start_time = get_start_time("KRW-BTC")
24         end_time = start_time + datetime.timedelta(days=1)
25
26         if start_time < now < end_time - datetime.timedelta(seconds=10):

```

```

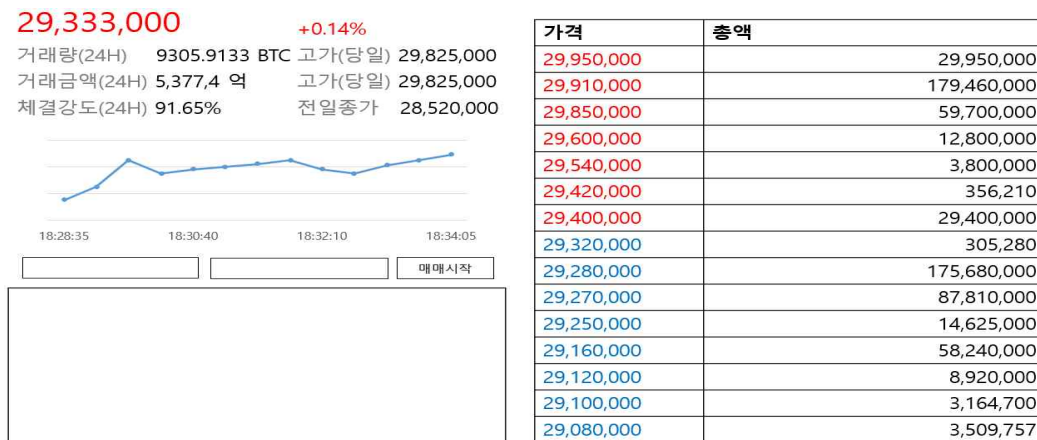
21     """잔고 조회"""
22     balances = upbit.get_balances()
23     for b in balances:
24         if b['currency'] == ticker:
25             if b['balance'] is not None:
26                 return float(b['balance'])
27             else:
28                 return 0
29     return 0
30
31 def get_current_price(ticker):
32     """현재가 조회"""
33     return pyupbit.get_orderbook(ticker=ticker)["orderbook_units"][0]["ask_price"]
34
35 # 로그인
36 upbit = pyupbit.Upbit(access, secret)
37 print("autotrade start")
38
39 # 자동매매 시작
40 while True:
41
42     target_price = get_target_price("KRW-BTC", 0.5)
43     current_price = get_current_price("KRW-BTC")
44     if target_price < current_price:
45         krw = get_balance("KRW")
46         if krw > 5000:
47             upbit.buy_market_order("KRW-BTC", krw*0.9995)
48     else:
49         btc = get_balance("BTC")
50         if btc > 0.00008:
51             upbit.sell_market_order("KRW-BTC", btc*0.9995)
52     time.sleep(1)
53 except Exception as e:
54     print(e)
55     time.sleep(1)

```

[사진 5] 코드구현

우선 새로운 파이썬 파일을 만들어준다. 코드를 설명하자면, 우선 필요한 모듈을 미리 가져와준다.(행01~03) 먼저 access와 secret 변수를 만들고 이 부분에는 자신의 업비트 API 키를 넣어 준다.(행05~06) 그리고 36행에 pyupbit에 access 키와 secret 키를 이용해서 로그인을 해주는 코드를 넣는다. 로그인을 해준 upbit를 이용해서 자동매매를 할 것이다. while True문으로 무한 루프를 반복하면서 자동매매 코드를 동작시킬 건데, 중간에 Try Except를 이용해서 오류가 발생하면 잡아주도록 만들 것이다.(행40~60) 먼저 현재시간과 시작 시간, 끝나는 시간을 받아온다.(행42~44) 주식시장과 다르게 코인 시장에서는 시작 시간이 별도로 정해져 있지 않기 때문에 get_start_time()함수(행43,46)를 활용해서 업비트 API에서 제공하는 OHLCV(Open/High/Low/Close/Volume) (행10)를 조회할 때 일봉(하루

에 이루어진 주가)으로 조회하면 그날의 시작 시간이 나오는데, 그게 9시로 설정되어 있다. 그거를 받아올 수 있도록 get_ohlcv를 이용해서 값을 받아오고 거기에 가장 첫 번째 값인 시간 값을 받아와서 start_time을 넘겨준다. 요약하면 그냥 9시를 가져오는 것이다. 그 다음으로 장 마감 시간인 end_time은 9시에서 1일을 더해 준 값인 다음날 9시로 마감시간으로 설정을 해놓았다.(행44) 그러면 시간에 따라서 if, else 구문으로 구분동작을 하게 된다. 먼저 if문은 시작 시간과 끝나는 시간에서 끝나는 시간이 9시면 무한정으로 돌아갈 것이기 때문에 10초를 빼준 8시 59분 50초까지 돌아가도록 만들어준다.(행46) 그렇게 되면 9시부터 8시59분50초 사이에 현재 시간이 이 조건을 만족하면 get_target_price()를 이용해서 매수 목표가를 설정해준다.(행47) get_target_price()함수는 변동성 돌파 전략을 이용해서 매수 목표가를 설정해주는 함수이다. 여기에 티커와 k값을 넣어주게 되면 변동성 돌파 전략 계산을 이용해서 이를 치에 해당하는 일 데이터를 조회하고(행10) 그 다음 줄인 (행11)에서 변동성 돌파 전략을 사용해 준다. 비트코인에서는 다음날 시가와 붙어 있기 때문에 변동폭을 더해줘서 목표가를 설정해 주고 리턴을 해준다.(행11~12) 그러면 결국 변동성 돌파 전략으로 목표가를 세워주고(행47) 현재 가격이 목표 가격보다 높다 면(행49) 그때 KRW인 내 원화 잔고를 조회하고(행50) 이게 최소 거래 금액인 5천원 이상이면 비트코인을 매수한다. 수수료 0.05%도 고려해서 0.9995도 써놓았다.(행51~52) 그리고 현재시간이 8시59분50초 이상이고 9시 전일 때(else에 해당할 때) 당일 증가에 비트코인을 전량 매도한다.(행54) 매도할 때, 현재 가지고 있는 BTC잔고를 가져와서 현재 잔고가 5천원 이상이고 비트코인 가지고 있는 것이 최소 거래 금액인 5천원 이상이면 9시 되기 10초 전부터 계속 전량 매도한다. 이때도 수수료를 고려한다.(행55~56) 그 후 코드를 구현한 대로 실행을 시켜주면 자동거래가 시작이 된다. 실행이 안될 때 마다 오류에 나와 있는 필요한 모듈을 vscode 터미널 창에서 pip install ~ 명령어로 다운로드 받으면 된다.



[사진 5] 자동매매 프로그램 GUI

미리 보는 GUI 디자인이다. 거래할 티커를 정하고 매매를 시작하면 자동으로 돌아가게끔 하고, 그림에 나와 있는 대로 24시간 거래량, 저가와 고가, 전일 증가, 거래금액, 체결강도가 나오고 오른쪽에는 가격과 가격별 매수가가 나오게끔 만들 것이다.

여기까지가 자동매매 코드에 대한 설명이다. 전략을 변경하고 싶다면 `target_price`를 구하는 `get_target_price` 코드를 자신의 전략에 맞게 구하면 새로운 자동매매 코드를 만들 수 있고 `k` 값을 변화시켜서 코드를 좀 더 다양화할 수 있다. 그리고 최소한의 손실을 줄인 것 뿐이지 자신만의 투자 전략을 구성해 나가면서 프로그램을 써야 효율을 낼 수 있다.

4. 결론 및 향후과제

앞서 설명했지만 이 전략이 항상 득을 볼 수 있는 것은 아니다. 단순하게 봤을 때, 전날 변동 폭을 기준으로 오늘 `k`배 상승했을 때 매수를 하는 프로그램이라 아예 하락장일 때에는 매수를 자동으로 하지 않기 때문에 손실을 줄여주는 것이다. 하지만 조건에 맞게 전날 변동 폭을 기준으로 오늘 `k`배 상승했을 때 프로그램이 매수를 했는데, 당일 종가에 자동으로 팔리기 전에 가격이 하락하면 손해를 보는 구조이기 때문에 이 프로그램만 의존한 다기 보다는 가상화폐 거래에 대해 개인적인 전략과 많은 지식을 쌓은 상태에서, 신중하게 프로그램을 쓸 티커를 선택하고 프로그램을 사용해야 할 것이다. 그렇게 되면 손해를 덜 보면서 하루 종일 핸드폰을 보지 않아도 되게끔 편하게 거래를 해주는 프로그램이 될 것이다. 항상 시장 상황과 전략에 따라 손실이 발생할 수 있어서 투자의 책임은 투자자 본인에게 있다.

이 프로그램은 24시간 컴퓨터를 켜놔야 하는 단점이 있다. 하지만 클라우드 상의 컴퓨터에 올려놓게 되면 내 컴퓨터는 꺼놔도 클라우드에서 내 코드가 스스로 돌아가게 될 것이다. AWS라는 아마존에서 만든 클라우드 서버는 AWS 프리 티어라는 것을 제공해 주는데 12개월 동안 무료로 서버를 돌릴 수 있기 때문에 이 클라우드 서버를 활용해서 클라우드 상에서도 돌아갈 수 있게끔 계속 연구를 진행해 볼 생각이다.

참고문헌

- [1] 한국의 가상화폐; 연구동향 및 향후연구 시사점, 이종수, 박현숙 한국인터넷정보학회 (2021), 111-121
- [2] 가상화폐 거래를 위한 시장안정화요인의 우선순위 도출, 강다연, 황종호, 한국콘텐츠학회 (2018) 519-526
- [3] 증권 데이터 예측/분석을 위한 효과적인 API활용 데이터 수집 방법 연구, 이승호한세대학교 대학원(2017), 1-38
- [4] PC 그라드 환경을 위한 오픈 API 개발 동향 및 사례, 한창환, 한연희, 길준민, 윤준원, 최장원, 한국콘텐츠학회지(2008) 15-23
- [5] 심층 신경망을 이용한 변동성 돌파 전략 기반 주식 매매 방법에 관한 연구, 이은우, 이원부, 한국콘텐츠학회논문지(2022) 81-93
- [6] 앱 개발자를 위한 오픈 API 서비스 플랫폼 (Open API Platform on SaaS), (주)애플런트 (2013)
- [7] 파이썬을 이용한 프레임 내 웹 페이지 스크래핑 기법, 윤수진, 승리, 우영운, 한국정보통신학회 2019 춘계학술대회(2019), 271-274