

## 파이썬 언어 기반의 퀀트 전략 개발 시스템

김상혁<sup>o</sup> 노영준 윤세영 이정준  
한국산업기술대학교 컴퓨터공학부

{tkgut0728, qkduqh, 2016156020, jjlee}@kpu.ac.kr

## Python language-based Quant Strategy Development System

Sang-Hyeok Kim<sup>o</sup> Young-Jun No Se-Young Yoon Jeong-Joon Lee  
Dept. of Computer Engineering, Korea Polytechnic University

## 요 약

국내 주식 투자 인구가 해마다 증가하고 있고 좋은 수익률을 창출하기 위해 여러 투자 방법이 등장하고 있다. 기본적인 분석은 기업의 재무제표, 기업의 실적, 시장 등의 내재적 가치를 분석하여 투자하는 방법이다. 이러한 방법은 기업의 재무제표, 기업의 실적을 단기적으로 알 수 없기 때문에 단기적인 미래의 주가를 예측하는데 한계가 있다. 기술적 분석은 과거 및 현재 주가와 거래량을 기반으로 미래의 주가를 예측하는 분석 방법이다. 기술적 분석 투자에 대한 연구와 관련 서비스가 많이 존재한다. 기존의 금융거래 전략 시스템이나 플랫폼은 거래 전략을 작성하는데 있어 자유도의 제약이 많다. 또한 외부 데이터 활용 및 다 종목 시뮬레이션 수행이 어렵다는 단점이 있다. 본 논문에서는 기존 시스템이 지원하지 않은 외부데이터를 활용한 전략 개발과 다 종목 시뮬레이션을 이용하여 거래 전략에 적합한 종목을 찾을 수 있는 시스템을 제안한다.

## 1. 서 론

투자 인구가 해가 지날수록 증가하고 있음을 뉴스나 기사를 통해서 접할 수 있다[1]. 하지만 주식 투자로 수익률을 극대화하는 것은 모든 투자자가 겪는 어려운 문제 중 하나이다. 성공적인 투자를 위한 주식예측방법으로 기본적인 분석 (fundamental analysis)과 기술적 분석(technical analysis)이 있다. 기본적인 분석은 기업의 재무제표, 기업의 실적, 시장 등의 내재적 가치를 분석하여 투자하는 방법이다. 이러한 방법은 기업의 재무제표, 기업의 실적을 단기적으로 알 수 없기 때문에 단기적인 미래의 주가를 예측하는데 한계가 있다.[2] 기술적 분석은 과거 및 현재 주가와 거래량을 기반으로 미래의 주가를 예측하는 분석 방법이다.

주가이동평균선을 중심으로 한 연구를 보면 기술적 분석을 통해 더 나은 성과를 보여주었다[3]. 관련 시스템으로는 GENPORT [4], YesTrader[5]와 같은 시스템이 존재한다. [4]의 시스템은 투자전략을 쉽게 작성하고 다양한 종목에 적용할 수 있었지만 사용자가 자유롭게 전략을 작성하기 어려운 점이 있다. [5]의 시스템은 사용자가 다양한 전략을 자유롭게 작성하여 하나의 종목에 적용할 수 있지만 생성한 전략을 한 번에 여러 종목에 대해서 적용하기 어려운 점이 있다. [4], [5]에 공통으로 기술적 지표 이외의 외부데이터(기사, 미리 정제한 데이터)를 사용하기 어려운 점이 있다.

본 논문에서 위와 같은 문제점을 극복하고 다음과 같은 추가 기능을 지원한다. 사용자가 투자 전략 작성과 외부데이터를 쉽게 조작할 수 있다. 본 시스템에서 다양한 기술적 지표와 외부데이터를 이용하여 투자 전략을 적용 및 개발할 수 있다. 거래 전략 시뮬레이션을 여러 종목에 적용하여 가장 적합한 종목을 쉽게 찾을 수 있다. 사용자가 다양한 전략을 개발하고 개발한 전략을 이용한 시뮬레이션을 통해 주가 종목에 맞는 최적의 전략을 시스템을 사용하여 개발할 수 있다.

	YesTrader	GenPort
시뮬레이션	한가지 전략에 대해 다종목 시뮬레이션 불가능	2007년 이전 과거 데이터로 시뮬레이션 불가능 시뮬레이션 사용이 유료
사용성	엑셀권지 학습 필요(난이도: 중)	사용자 수식 작성이 불필요함(난이도: 하)
사용자 정의 함수 사용	가능	불가능
외부 데이터 사용	불가능	불가능

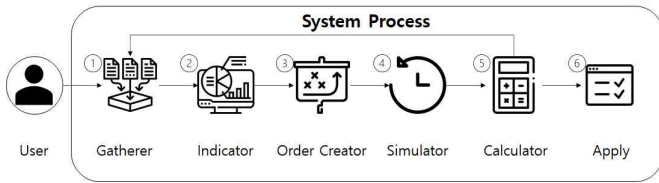
[표 1]

## 2. 시스템 개요

본 논문에서 구현한 시스템 구성도는 [그림 1]과 같다. [그림 1]의 ① Gatherer는 적용할 주가 데이터의 일봉 및 주봉 데이터를 수집한다. [그림 1]의 ② Indicator는 주가 데이터를 기반으로 기술적 지표 값을 생성한다. [그림 1]의 ③ Order Creator는 Python 코드 형식의 전략 조건식을 입력 받아 과거 데이터에 대해 전략을 적용하여 매수/매도 신호를 기록한 주문 파일을 생성한다. [그림 1]의 ④ Simulator는 주문 파일을 과거 주가 데이터에 적용하여 자산, 거래 횟수, 매수/매도 시점 등을 기록하는 거래 시뮬레이션을 수행하고 기록한 내용을 파일로 저장한다. [그림 1]의 ⑤ Calculator는 ④ Simulator에서 저장한 파일을 입력받아 일간, 주간, 월간, 연간 수익률, 누적 수익률, 현재 자산을 계산한다하고 계산한 수익률과 자산을 파일로 저장한다. ⑥ Apply는 유효한 결과를 얻은 전략을 다수의 종목에게 적용한 후, 종목 별 전략 적용 결과를 비교하여 기록한다.

사용자는 ① Gatherer부터 ⑤ Calculator까지 진행한 후 전략의 유효성을 결정한다. 전략이 유효한 결과를 가지지 못한다면

① Gatherer부터 다시 전략 개발을 진행한다. 좋은 성능을 가진 전략을 개발한 후, ⑥ Apply를 이용하여 다수의 종목에 개발한 전략을 적용한 후 백테스팅 결과를 비교 및 분석하여 개발한 전략에 가장 효과적인 결과를 가진 종목을 선정할 수 있다.



[그림 1] 시스템 구성도

### 3. 개발환경

본 논문에서 설계하고 구현하고자 하는 시퀀스 레이블러 및 금융거래전략 개발 시스템의 운영체제는 windows 10, 개발 언어는 python 3.8이다. 기술적 지표를 생성할 때 쓰는 라이브러리는 TA-Lib[6]를 사용했으며, 주가 데이터 수집은 FinanceDataReader[7]를 사용한다. gui 프레임워크는 pyside2[8] IDE(통합 개발환경)은 spyder3와 vscode로 개발하였다.

### 4. 시스템 구현

본 장에서는 구현한 시스템을 화면과 함께 설명한다. 사용자가 프로그램을 실행하면 [그림 2]의 화면이 나온다. 시스템의 모든 메뉴는 프로젝트 단위로 수행이 가능하며, 각 메뉴의 기능들은 서로 파일을 통하여 연결이 가능하게 구현 하였다. [그림 2]의 ①, 사용자가 원하는 주식 종목의 가격 데이터를 다운로드 하여 캔들 차트 형식으로 시각화 할 수 있다. 가격 데이터는 종목 코드를 이용하여 수집할 수 있으며, 로컬 파일로 저장한다. 또한, 가격 데이터에 각종 기술적 지표를 적용할 수 있다. 기술적 지표는 시가, 종가, 저가, 고가, 거래량을 사용하며, 원하는 파라미터로 변경 하여 다양한 형태의 지표를 제공할 수 있다. [그림 2]의 ②, 두 개의 시계열 데이터 파일을 하나의 파일로 병합 할 수 있다. 파일 병합 과정에서 데이터들의 조인 옵션과 결측값 처리 옵션을 선택할 수 있다. [그림 2]의 ③, 거래전략 메뉴에서는 개발하고자 하는 거래 전략을 편집할 수 있다. 사용하고자 하는 기술적 지표를 정의하는 지표 선언식 편집창과, 선언한 지표들을 사용하여 전략을 개발할 수 있는 거래 전략 편집기가 있다. 전략 편집기의 문법은 파이썬의 문법을 이용하고, 상향돌파, 하향돌파, 가격, 지표를 사용할 수 있다. 편집이 완료된 거래 전략은 전략 조건식에 맞춰 도출된 매매지점을 기반으로 그래프 형식으로 시각화하여 보여준다. [그림 2]의 ④, 백테스트 메뉴에서는 종목과 기간을 설정할 수 있는 기능을 지원한다. 백테스트 메뉴에서는 거래 전략을 기반으로 백테스팅을 수행한다. 매수횟수, 매도횟수, 최저손익률, 최고손익률, 최종손익률을 계산하여 표 형태로 제공한다. 또한, 거래 전략을 적용하여 매매를 수행 하였을 때의 누적 및 기간별 수익률과 주식 가격의 상승률을 그래프로 시각화 하여 비교 및 분석을 할 수 있다. [그림 2]의 ⑤, 종합차트 메뉴에서는 두 개의 시계열 데이터들을 병합 및 합성하여 그래프로 시각화 하여 보여주는 기능을 가지고 있다. [그림 2]의 ⑥, 종목찾기 메뉴는 개발한 거래 전략의 성능을 최대화 하는 종목을 찾는 기능을 가지고 있다. 즉, 하나의 전략에 대한 다 종목 백테스팅을 수행하여 종목 별 수익률을 비교 및 분석할 수 있다. [그림 3]은 백테스팅 완료 후 매수횟수, 매도횟수, 최저/최고/최종 손익률로 구성된 통계 자료 표와 전략 수익률 그래프를 보여주는 화면이다.



[그림 2] 시스템 화면



[그림 3] 백테스팅 화면

### 5. 결론

본 논문에서는 기술적 분석과 외부데이터를 이용하여 거래를 수행하는 전략을 생성, 거래전략의 수익률을 검증 위한 투자 시뮬레이션 등의 기능을 제공하는 전략개발시스템을 제안하였다. 종목차트, 자료정제, 거래전략, 백테스트, 종합차트, 종목찾기로 구성된 메뉴들의 기능을 순서대로 사용하여 거래전략 개발 및 적합 종목 탐색을 지원한다.

본 시스템을 통해 기존 시스템과 달리 기술적 지표와 외부 데이터를 결합하여 파이썬 언어로 자유도가 높은 거래 전략을 개발할 수 있고 하나의 거래 전략으로 단일 종목 시뮬레이션 뿐만 아니라 다 종목 시뮬레이션을 이용하여 전략에 적합한 종목을 찾을 수 있다.

향후 연구로 서로 연관된 두 개 이상의 종목을 전략에 적용할 수 있는 연구와, 대용량으로 진행되는 시뮬레이션 시간을 단축할 수 있는 연구가 앞으로 수행되어야 할 과제이다.

### 참고문헌

- [1] 김영배, (2021) “주식 인구 1천만’ 시대”, 한겨레, 4월 8일.
- [2] 최진영, 김민구, “기술적 지표 기반의 주가 움직임 예측을 위한 모델 분석”, 한국정보처리학회 26권 2호, 885p, 2019.
- [3] 이윤선, “한국증권시장에서 기술적 분석의 경제적 효과 -주가 이동평균선을 중심으로-”, 한국금융공학학회 금융공학연구 2권, 2호, 1p, 2003.
- [4] GENPORT, <https://genport.newsysstock.com/Main.aspx/>.
- [5] YesTrader, <https://www.yesstock.com/>.
- [6] TA-Lib, <https://mrjbq7.github.io/ta-lib/>.
- [7] FinanceDataReader, <https://financedata.github.io/>.