대시보드 설명

퀀트응용경제학 3 기 배이한

데이터분석입문 과제를 수행하는 동안 중점을 둔 곳은 최대한 다양한 기능을 활용해보는 것이었습니다. 이러한 의도를 가지고 수집한 미국 대형주의 종목 지표 및 애널리스트의 실적 예상 데이터로 구성한 shiny 대시보드를 제출합니다.

데이터 수집한 과정은 2 단계로 나뉩니다. 우선 (1) read_html 코드를 활용하여 https://www.slickcharts.com/sp500 에서 시가총액 순서로 종목의 티커코드를 수집하고, (2) 이를 활용해 Selenium 으로 Seeking Alpha 의 종목별 a) Earnings Estimate, b) Valuation Metrics, c) Dividends Estimate 페이지 (종목당 3 개의 웹페이지) 를 접속하여 애널리스트의 전망, 시킹알파 퀀트 랭킹, 과거 PER, PSR 등의 지표를 포함한 총 3 개 시점에 대한 14 개의 데이터를 가져옵니다.

(1)은 수업과 같이 문제가 없었으나, (2)에서는 문제가 상당수 발생했습니다. 시킹알파 페이지에서 데이터를 수집할 경우, 2~3 개의 페이지만 방문하여도 접속자를 로봇으로 분류하여 "Are you a robot?" 와 같은 화면을 표시하여 추가적인 데이터 수집을 방지했습니다. 차단을 우회하기 위한 다양한 실험을 통해 해답은 한 페이지 방문 후 selenium 종료 및 재가동하기, 매 접속 간 Sys.sleep( ) 명령어를 특정 기간 수행하기, 그리고 다음 접속 시 포트값 변경하기라는 것을 발견했습니다. 하지만 과도하게 할 경우 데이터 크롤링에 오랜 시간이 들어간다는 문제가 있었고, 반복적인 실험을 통해 시킹알파의 차단 알고리즘을 우회할 수 있는 최단시간은 사이트 접속하고 크롤링 실행 전 15 초, 실행 후 46 초 대기하는 것임을 발견했습니다. 하지만 다른 날 같은 과정을 반복했으나 데이터가 수집되지 않았음을 알게 되었습니다. 시킹알파는 html 의 pattern 값을 난수로 구성하여, 매 3 시간 변경되도록 설계되어 있었고, 이는 결국 3 시간마다 크롤링을 위해 14 개 데이터에 대한 pattern 을 수기로 변경해야함을 의미했습니다. 또한, 포트 번호를 계속하여 변경하다 보니 간혹 이미 사용 중인 포트 번호 도달 시 에러가 발생하였습니다. 이로 인해 수집의 속도가 저하되었고, 잠재적으로 기간 내 수집 가능한 종목의 수는 크게 감소했습니다.

결국, 50 개 종목에 대한 데이터를 수집하였고, 이를 활용해 대시보드 제작을 시작했습니다. 처음에는 Flex dashboard 로 작업을 시작했으나, 제가 원하는 "선택 종목에 따라 페이지 구성을 변경"하는 것이 불가능했습니다. 이를 위해 input 에 따라 output 이 변경되는 반응형 대시보드 구성이 가능한 shiny 대시보드를 선택했고, 이를 통해 대상 종목 전체의 실적 전망을 보여주는 'Overall' 페이지와 원하는 종목을 input 으로 넣어 해당 종목에 대한 output 만 표기되는 'Stock' 페이지를 구성할 수 있었습니다. 해당 대시보드를 제작하는데 소요된 시간 중 상당 시간이 Stock 페이지를 구성하는 데에 쓰였고, 특히 배당금을 지급하지 않는 종목 (e.g. AMZN, GOOGL, TSLA 등)에 대해서는 배당금 전망을 선택했을 때 차트 대신 문구가 나오도록 하는 작업이 가장 오래 걸렸던 것 같습니다. 또한 이번 대시보드 제작 과정을 통해 html 형태로 표기되는 output 을 원하는 형태와 위치로 수정하는데 필요한 다양한 렌더링 문법(reactive, renderPlot, renderDataTable 등)의 활용법 학습할 수 있었습니다.