

코스피 시장과 코스닥 시장 내 기업들의 현재가 분석

60152177 김현목

목차

명지대학교
2019 2학기
빅데이터 프로그래밍 프로젝트



01 주제 선정동기



02 시스템 아키텍처



03 프로젝트 전개과정



04 프로젝트 결과

주제 선정 동기

01





01 프로젝트 선정 동기

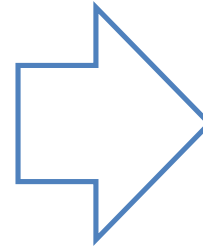
1. 추후 빅데이터로의 확장 가능성 염두
2. 시계열 분석을 위한 RealTime 갱신되는 데이터 선정
3. 타 데이터셋과의 융합을 통한 새로운 가치 창출 여부

아키텍처 02

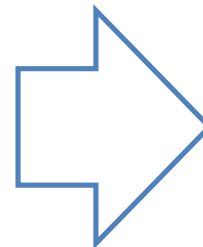


02 시스템 아키텍처

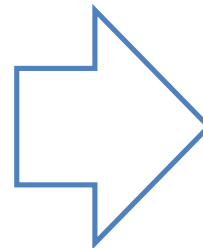
1. 데이터 스크래핑



2. 데이터 전처리



3. 데이터 분석



결과 보고서 작성

전개 03





03 프로젝트 시나리오

목표

데이터 수집 건수 - 15만 Row 이상
수집된 데이터 양 - 1000 KB 이상

1. 데이터 스크래핑 코드 작성
2. 윈도우 스케줄러를 이용한 데이터 스크래핑 - 자동화
3. 수집된 데이터 전처리를 위한 코드 작성
4. 전처리 코드를 이용한 데이터 통합 및 전처리 -> final_dataset 구축
5. HDFS에 final_dataset 업로드 및 HIVE 테이블 구성
6. HIVE QL을 이용한 데이터 분석
7. 결과 해석



03 프로젝트 전개

1. 데이터 스크래핑 코드 작성 – BeautifulSoup and Selenium

* BeautifulSoup을 이용한 데이터 스크래핑

```
url = 'https://finance.naver.com/sise/sise_market_sum.nhn?sosok='+str(sosok)+'&page='+str(page)
result = requests.get(url)
soup = BeautifulSoup(result.content, 'html.parser')

stock_table = soup.find("table", {"class": "type_2"})
summary_stock = stock_table.find('tbody')
data_list = summary_stock.find_all('tr')
data_list = data_list[1:]
```

* Selenium을 이용한 마지막 페이지 번호 알아내기

```
# 코스피 목록의 마지막 페이지 넘버를 알아낸다.
driver1 = webdriver.Chrome('C:\\Users\\HyunMok\\Desktop\\projectforhadoop\\chromedriver_win32\\chromedriver.exe')
driver1.get('https://finance.naver.com/sise/sise_market_sum.nhn?sosok=0')
# 맨 마지막 페이지를 클릭한다.
driver1.find_element_by_css_selector('#contentarea > div.box_type_1 > table.Nnavi > tbody > tr > td.pgRR > a').click()
# url에서 'page=' 뒤에 있는 숫자를 슬라이싱으로 가져온다.
reference_num = driver1.current_url.find('page=')
last_num_kospi = int(driver1.current_url[reference_num+5:])

driver1.close()
```

03 프로젝트 전개

2. 윈도우 스케줄러를 이용한 데이터 스크래핑 - 자동화

새 트리거 만들기

X

작업 시작(G): 예약 상태

설정

☒ 한 번(N) 시작(S): 2019-12-10 오전 9:30:00 ☐ 표준 시간대 간 동기화(Z)

☐ 매일(D)

☐ 매주(W)

☐ 매월(M)

고급 설정

☐ 작업이 지연되는 최대 시간(임의 지연)(K): 1 시간

☒ 작업 반복 간격(P): 30 분 기간(F): 무기한으로

☐ 반복 기간이 종료될 때 실행 중인 모든 작업 중지(I)

☐ 다음 기간 이상 실행되는 작업 중지(L): 3 일

☐ 만료(X): 2020-12-15 오후 4:27:11 ☐ 표준 시간대 간 동기화(E)

☒ 사용(B)

확인 취소

새 동작 만들기

X

실행할 작업을 지정해야 합니다.

동작(I): 프로그램 시작

설정

프로그램/스크립트(P):

C:\Users\Whyunmok\Anaconda3\pythonw.exe

찾아보기(B)...

인수 추가(옵션)(A):

stock_crawling.py

시작 위치(옵션)(O):

ok\Desktop\stock_data

확인



















취소

CHEDIVE SC...	준비	2019-12-10 오전 9:30에 - 트리거된 후 무기한으로 30 분마다 반복합니다.
stock_scrapp...	준비	2019-12-10 오전 9:30에 - 트리거된 후 무기한으로 30 분마다 반복합니다.
Update Chec...	준비	여러 개의 트리거가 정의되었습니다.
User_Feed_S...	준비	매일 오후 9:41에 - 2029-12-15 오후 9:41:51에 트리거가 만료됩니다.



03 프로젝트 전개

2. 윈도우 스케줄러를 이용한 데이터 스크래핑 - 자동화

 stock_20191210_9시30분	2019-12-10 오전 9:30	Microsoft Excel 실행...
 stock_20191210_9시30분	2019-12-10 오전 9:30	Microsoft Excel 워크...
 stock_20191210_10시0분	2019-12-10 오전 10:00	Microsoft Excel 실행...
 stock_20191210_10시0분	2019-12-10 오전 10:00	Microsoft Excel 워크...
 stock_20191210_10시30분	2019-12-10 오전 10:30	Microsoft Excel 실행...
 stock_20191210_10시30분	2019-12-10 오전 10:30	Microsoft Excel 워크...
 stock_20191210_11시0분	2019-12-10 오전 11:00	Microsoft Excel 실행...
 stock_20191210_11시0분	2019-12-10 오전 11:00	Microsoft Excel 워크...
 stock_20191210_11시30분	2019-12-10 오전 11:31	Microsoft Excel 실행...
 stock_20191210_11시30분	2019-12-10 오전 11:31	Microsoft Excel 워크...
 stock_20191210_12시0분	2019-12-10 오후 12:00	Microsoft Excel 실행...
 stock_20191210_12시0분	2019-12-10 오후 12:00	Microsoft Excel 워크...
 stock_20191210_12시30분	2019-12-10 오후 12:30	Microsoft Excel 실행...
 stock_20191210_12시30분	2019-12-10 오후 12:30	Microsoft Excel 워크...
 stock_20191210_13시0분	2019-12-10 오후 1:01	Microsoft Excel 실행...
 stock_20191210_13시0분	2019-12-10 오후 1:01	Microsoft Excel 워크...
 stock_20191210_13시30분	2019-12-10 오후 1:30	Microsoft Excel 실행...
 stock_20191210_13시30분	2019-12-10 오후 1:30	Microsoft Excel 워크...



03 프로젝트 전개

3. 수집된 데이터 전처리를 위한 코드 작성
4. 전처리 코드를 이용한 데이터 통합 및 전처리 -> final_dataset 구축

* 주식 시장이 개장하는 시간과 영향받는 시간에 스크래핑 된 파일만 load

```
# 파일 목록 중 확장자가 csv인 파일만 따로 저장하기
stock_csv = [file for file in file_list if file.endswith(".csv")]

# 주가가 변동하는 시간에 크롤링된 데이터만 추출한다.
time = ['_9시 30분', '_10시 0분', '_10시 30분', '_11시 0분', '_11시 30분', '_12시 0분',
        '_12시 30분', '_13시 0분', '_13시 30분', '_14시 0분', '_14시 30분', '_15시 0분', '_15시 30분', '_16시 0분']

stock_time_csv = list()
for i in stock_csv:
    for j in time:
        if j in i:
            stock_time_csv.append(i)
```



03 프로젝트 전개

3. 수집된 데이터 전처리를 위한 코드 작성
 4. 전처리 코드를 이용한 데이터 통합 및 전처리 -> final_dataset 구축
- * 데이터를 통합하고, 계산에 필요한 데이터는 numeric 타입으로 변경

```
# 수집한 데이터의 컬럼 목록을 가져온다.
tmp_columns = pd.read_csv('C:\\Users\\hyunmok\\Desktop\\stock_data\\'+stock_csv[0]).columns

# 최종 데이터셋을 구성할 데이터프레임 선언
final_dataset = pd.DataFrame(columns=tmp_columns)

# final_dataset에 모든 데이터를 병합한다.
for i in stock_time_csv:
    data = pd.read_csv('C:\\Users\\hyunmok\\Desktop\\stock_data\\'+i)
    final_dataset = pd.concat([final_dataset, data])

# 불필요한 열을 제거한다.
final_dataset.drop(['Unnamed: 0', 'NO'], axis=1, inplace=True)
```

```
# 계산에 필요한 컬럼들은 천의자리의逗를 지우고 데이터 타입은 numeric으로 변경
final_dataset['현재가'] = final_dataset.현재가.str.replace(',', '').astype('int64')
final_dataset['전일비'] = final_dataset.전일비.str.replace(',', '').astype('int64')
final_dataset['액면가'] = final_dataset.액면가.str.replace(',', '').astype('int64')
final_dataset['시가총액'] = final_dataset.시가총액.str.replace(',', '').astype('int64')
final_dataset['거래량'] = final_dataset.거래량.str.replace(',', '').astype('int64')
final_dataset['상장주식수'] = final_dataset.상장주식수.str.replace(',', '').astype('int64')

final_dataset['외국인비율'] = final_dataset.외국인비율.astype('float')
final_dataset['PER'] = final_dataset.PER.str.replace(',', '').astype('float')
final_dataset['ROE'] = final_dataset.ROE.str.replace(',', '').astype('float')
```



03 프로젝트 전개

5. HDFS에 final_dataset 업로드 및 HIVE 테이블 구성

```

DROP TABLE stock;
CREATE TABLE IF NOT EXISTS stock(
  no INT,
  date_yyyy_mm_dd STRING,
  time STRING,
  market STRING,
  company_name STRING,
  now_price INT,
  per_yesterday INT,
  in_decrease STRING,
  per_value INT,
  market_cap INT,
  listed_stock INT,
  foreign_rate DOUBLE,
  trading_volume INT,
  per DOUBLE,
  roe DOUBLE
)

```

stock.no	stock.date_yyyy_mm_dd	stock.time	stock.market	stock.company_name	stock.now_price	stock.per_yesterday	stock.in_decrease	stock.per_value	stock.market_cap	stock.listed_stock	stock.foreing_rate	stock.trading_volume	stock.per	stock.roe
0	2019-12-10	09:30	코스피	삼성전자	50900	300	-0.59%	100	3038619	5969783	56.97	1083521	8.45	19.63
1	2019-12-10	09:30	코스피	SK하이닉스	80100	600	-0.74%	5000	583130	728002	49.98	335388	3.75	38.53
2	2019-12-10	09:30	코스피	삼성전자우	41800	100	-0.24%	100	343967	822887	92.2	90090	6.94	null
3	2019-12-10	09:30	코스피	NAVER	173500	1500	-0.86%	100	285951	164813	58.68	32597	44.07	12.97
4	2019-12-10	09:30	코스피	현대차	120000	500	+0.42%	5000	256402	213668	41.58	50625	22.42	2.2
5	2019-12-10	09:30	코스피	삼성바이오로직스	385500	2500	-0.64%	2500	255066	66165	9.77	10729	113.82	5.51
6	2019-12-10	09:30	코스피	현대모비스	254000	2000	+0.79%	5000	242079	95307	47.74	26055	13.09	6.3
7	2019-12-10	09:30	코스피	셀트리온	167500	500	-0.30%	1000	214966	128338	20.04	69066	81.75	10.84
8	2019-12-10	09:30	코스피	LG화학	298500	3500	+1.19%	5000	210718	70592	37.95	36863	15.87	8.86
9	2019-12-10	09:30	코스피	신한지주	43750	350	-0.79%	5000	207462	474200	64.77	90553	6.57	9.21
10	2019-12-10	09:30	코스피	POSCO	231500	1500	-0.64%	5000	201838	87187	51.9	14080	11.94	3.88
11	2019-12-10	09:30	코스피	KB금융	48050	0	0.00%	5000	200903	418112	66.7	100855	6.56	8.78
12	2019-12-10	09:30	코스피	LG생활건강	1264000	3000	-0.24%	5000	197414	15618	45.15	1864	32.8	20.98
13	2019-12-10	09:30	코스피	SK텔레콤	238000	1000	-0.42%	500	192175	80746	37.58	10815	6.14	15.52
14	2019-12-10	09:30	코스피	삼성물산	99000	800	-0.80%	100	187793	189690	13.59	33109	11.06	8.06
15	2019-12-10	09:30	코스피	한국전력	28600	0	0.00%	5000	183602	641964	25.15	196796	-13.96	-1.86
16	2019-12-10	09:30	코스피	SK	260000	0	0.00%	200	182937	70360	25.33	5134	8.18	14.88
17	2019-12-10	09:30	코스피	기아차	43850	500	+1.15%	5000	177752	405363	42.26	65534	15.38	4.27
18	2019-12-10	09:30	코스피	삼성SDI	223000	0	0.00%	5000	153345	68765	43.49	18234	22.39	6.05
19	2019-12-10	09:30	코스피	삼성에스디에스	191000	1500	-0.78%	500	147792	77378	12.77	2858	23.48	10.91
20	2019-12-10	09:30	코스피	삼성생명	72700	300	-0.41%	500	145400	200000	15.7	15590	8.74	5.95
21	2019-12-10	09:30	코스피	SK이노베이션	146000	500	+0.34%	5000	135000	92466	35.02	28496	8.29	9.12
22	2019-12-10	09:30	코스피	KT&G	95500	0	0.00%	5000	131114	137292	47.45	40219	14.54	11.38
23	2019-12-10	09:30	코스피	카카오	149000	2500	-1.65%	500	128396	86172	30.33	50548	243.07	1.05
24	2019-12-10	09:30	코스피	LG	72600	0	0.00%	5000	125276	172557	34.48	17472	6.85	10.96
25	2019-12-10	09:30	코스피	엔씨소프트	539000	1000	-0.19%	500	118332	21954	49.81	9443	28.28	16.44
26	2019-12-10	09:30	코스피	LG전자	70000	900	+1.30%	5000	114553	163648	33.91	50055	10.21	9.03
27	2019-12-10	09:30	코스피	삼성화재	234000	2000	-0.85%	500	110857	47375	47.66	3019	11.05	8.81
28	2019-12-10	09:30	코스피	아모레퍼시픽	189500	3500	+1.88%	500	110779	58458	31.71	24142	39.37	7.75
29	2019-12-10	09:30	코스피	하나금융지주	36500	250	-0.68%	5000	109588	300242	67.11	110156	4.89	8.88



03 프로젝트 전개

5.1 LEFT SEMI JOIN을 사용하기 위한 테이블을 만들어서 저장하기

```
SELECT best_top5.company_name
FROM(SELECT kspi.company_name,kspi.now_price
FROM(SELECT stock.company_name,stock.date_yyyy_mm_dd,stock.time,stock.now_price
FROM stock
WHERE stock.market == '코스피') AS kspi
WHERE kspi.date_yyyy_mm_dd == '2019-12-10' and kspi.time == '09:30'
ORDER BY kspi.now_price DESC LIMIT 5) AS best_top5
```

Filter columns



best_top5.company_name

LG생활건강

태광산업

LG생활건강우

영풍

엔씨소프트

```
SELECT worst_top5.company_name
FROM(SELECT kspi.company_name,kspi.now_price
FROM(SELECT stock.company_name,stock.date_yyyy_mm_dd,stock.time,stock.now_price
FROM stock
WHERE stock.market == '코스피') AS kspi
WHERE kspi.date_yyyy_mm_dd == '2019-12-10' and kspi.time == '09:30'
ORDER BY kspi.now_price ASC LIMIT 5) AS worst_top5
```

Filter columns



worst_top5.company_name

미래산업

키위미디어그룹

서울식품

이아이디

KR모터스






코스닥 내 현재가가 높은 5개 기업과 낮은 5개 기업도 위 코드와 같이 진행



03 프로젝트 전개

5. HDFS에 final_dataset 업로드 및 HIVE 테이블 구성

<결과>

 kospi_best_top5
 kospi_worst_top5
 kosdac_best_top5
 kosdac_worst_top5
 stock



03 프로젝트 전개

6. HIVE QL을 이용한 데이터 분석

- kospi 시장 내 현재가가 가장 높은 5개 기업의 일평균 현재가 추이를 분석

```
SELECT stock.company_name, stock.date_yyyy_mm_dd, AVG(stock.now_price)
FROM stock LEFT SEMI JOIN kospi_best_top5 on (stock.company_name = kospi_best_top5.best_company_name)
GROUP BY stock.company_name, stock.date_yyyy_mm_dd
```

Kospi의 현재가가 가장 낮은 5개 기업 분석

Kosdac의 현재가가 가장 높은 5개 기업과
가장 낮은 5개 기업 분석

=> 모두 동일한 방법으로 접근

LG생활건강	2019-12-10	1265357.142857143
LG생활건강	2019-12-11	1251142.857142857
LG생활건강	2019-12-12	1264285.7142857143
LG생활건강	2019-12-13	1244428.5714285714
LG생활건강우	2019-12-10	734642.8571428572
LG생활건강우	2019-12-11	735214.2857142857
LG생활건강우	2019-12-12	748571.4285714285
LG생활건강우	2019-12-13	743857.1428571428
엔씨소프트	2019-12-10	538785.7142857143
엔씨소프트	2019-12-11	544785.7142857143
엔씨소프트	2019-12-12	539214.2857142857
엔씨소프트	2019-12-13	536071.4285714285
영풍	2019-12-10	628642.8571428572
영풍	2019-12-11	632428.5714285715
영풍	2019-12-12	655142.8571428572
영풍	2019-12-13	639500.0
태광산업	2019-12-10	983071.4285714285
태광산업	2019-12-11	986571.4285714285
태광산업	2019-12-12	989642.8571428572
태광산업	2019-12-13	987000.0

결과 03





02 프로젝트 결과

기업 \ 시장	코스피	코스닥
현재가가 높은 5개 기업	안정 추구	대개 기업 꾸준한 증가 추이 보임
현재가가 낮은 5개 기업	대개 기업 꾸준한 증가하는 추이 보임	현재가 변동폭 크지 않음

한계

1. 데이터를 수집하는 기간이 짧음
-> 시계열 분석을 위한 충분하지 못한 데이터 양
2. 단순 지표를 통한 기업 가치 평가의 오류 -> 다양한 지표 활용 X

확장 가능성

1. 데이터 수집 기간 증가 + 스크래핑 간격 축소 -> 빅데이터로 발전 가능성
2. 타 데이터셋과의 융합을 통한 새로운 가치 창출 및 정확한 결과 제시
(EX. 기업 신용평가 모델 + 해당 기업 주식 변동률 => 추후 더 정확하고 안전한 신용 등급 제시)



Thank U :D