



양성심 포트폴리오

신입 개발자를 꿈꾸는 금융권 인재입니다.

Developer

INDEX



01

Experienc

- 맡은 업무
- 개발



02

Data Science

- 데이터 분석
- Python Lover



03

Current study

- Web
 - Java
 - Django & Flask



04

Finance

- 공부(자격증)
- Fintech

■ 양성심

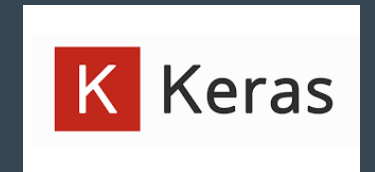
- 1993.01.03 생(만 27세)
- 홍익대학교 금융보험학과 (2017.03 졸업)
- 자산 운용사 2년 재직(시스템 트레이더)
- Python Data Analyst
- 관리 : Github
- 파이썬 데이터 분석(머신러닝, 딥러닝)
- 파이썬 고속화 (Cython)
- Web(Django, Flask, JSP)



INTERESTED COUNTRIES

- KOREA
- JAPAN
- CHINA

기술 스택



01 Experience

2017



입사

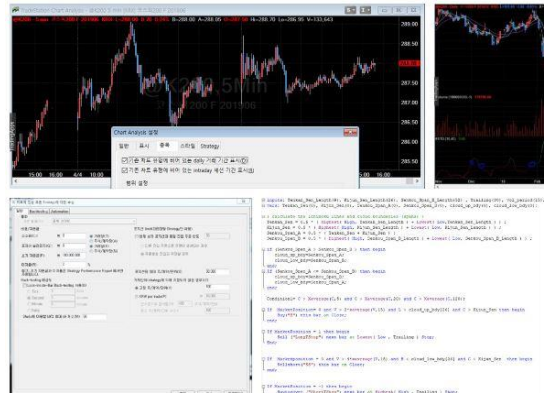
- 파생상품 운용팀
- 달러 및 주가지수 선물 시스템 트레이더

2018



개발

- 프랍 트레이딩 용 매매 규칙 데이터 분석



2019

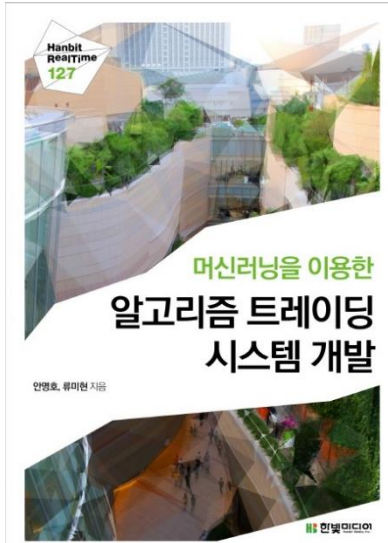


개발

- 백 오피스 정산 작업 자동화
 - 시세 크롤링
 - 계좌 스크래핑
 - 엑셀 매크로
- 암호화폐 트레이딩 로직
- 평균 주가 회귀 종목 분석
- 금융 시세 데이터 크롤링

회귀 종목 추출 및 트레이딩 과정

참고자료



처음 파이썬을 배우기 시작했던 동기

“나만의 로직으로 트레이딩 시스템을 개발해보자”

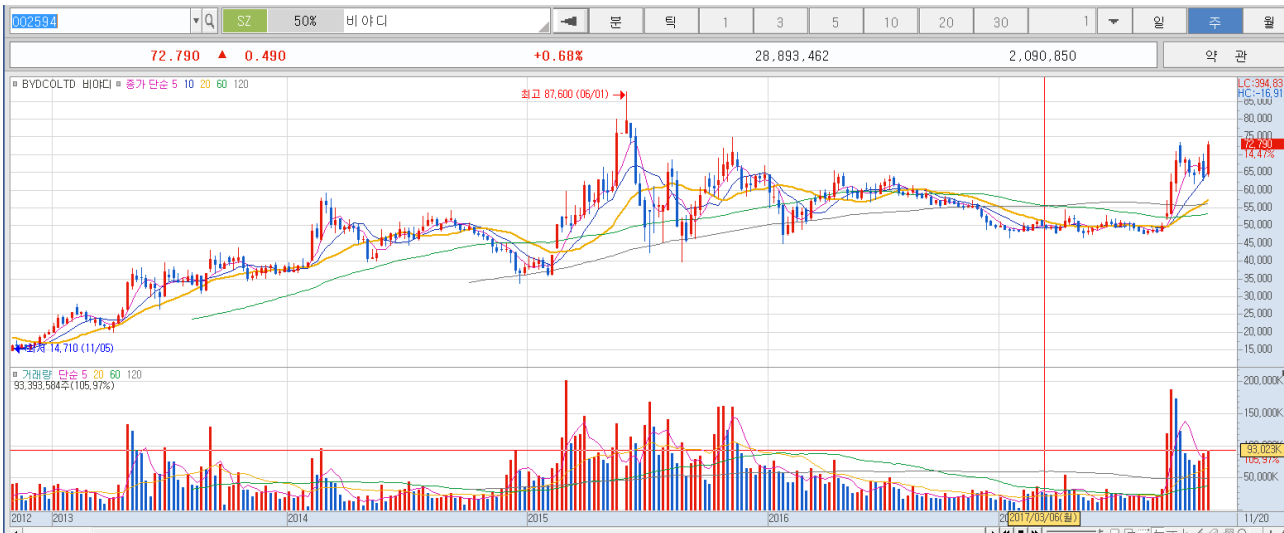
-퀀트 트레이딩 아이디어를 머신러닝, 파이썬으로 구현

- 서적1. 머신러닝 개념, 평균회귀모델로 포트폴리오 빌더 구현
- 서적2. 증권사 api에서 데이터를 불러와 거래량, 배당률 기반 투자 알고리즘을 구현하고 실제매매까지 다룬 서적



평균회귀를 보이는 종목들을 추출하기 위해 두 서적의 기술 아이디어를 혼합.

- +평균회귀 성향을 보이는 종목을 추출하게 전 종목 대상으로 확대 재검색 및 주가 특이점 필터 추가
- +데이터 수집 경로를 크롤링과 Api로 다각화



회귀 종목 추출 및 트레이딩 과정

전종목 1년치 데이터 (2019,03,20~ 2018,03,20) 의 증가와 거래량 데이터를 가지고 주가방향예측을 한다.
중간에 거래가 정지되거나 거래량이 전무해서 데이터가 전부 똑같은 3가지 종목(드림텍, 세화이앤씨..) 등을 제외시
키고 진행했다.

```
#[기본 import]
import bs4
from urllib.request import urlopen
import webbrowser
import requests
import sqlite3
import win32com.client
import timeit
import datetime
import statsmodels.tsa.stattools as ts
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import numpy as np
import pandas as pd
from warnings import simplefilter
import time
```

1.단계 전종목의 이름과 코드 넘버를 따올 웹사이트를 찾아서 크롤링 해온다. 저장은 딕셔너리에 한다.

```
stockcode = 'http://vip.mk.co.kr/newSt/rate/item_all.php'
source = urlopen(stockcode).read()
source = bs4.BeautifulSoup(source,'lxml')

td = source.findAll('td',class_='st2')
td1 = [str(i)[101:].replace("</a></td>","",) for i in list(td)]
code = {}
for i in td1:
```

→ 전처리 과정 생략 사용한 3가지 모형지표



5. ADF , Hurst, Halflife 지수 계산하는 3가지 함수

- ADF: 검정통계량이 5% , 10 % 보다 작아야 평균회귀 의미가 있다
- Hurst : 0 에 가까울 수록 평균회귀현상을 보임
- Half : 평균으로 되돌아오는 데 걸리는 시간. 짧을 수록 회귀성향이 강하다고 봄.

```
def calcADF(df):
    df.fillna(method = 'bfill')
    df.dropna(how='any')
    try:
        adf_result = ts.adfuller(df)
        critical_value = adf_result[4]
    except ZeroDivisionError:
        adf_result = 1
        critical_value = 1
    return [adf_result[0], critical_value['1%'],critical_value['5%'],critical_value['10%']]

def calcHurstExponent(df):
    lags = range(2,100)
    ts = np.log(df)

    tau = [np.sqrt(np.std(np.subtract(ts[lag:],ts[:-lag]))) for lag in lags]
    # x축                      y축                      1차방정식
    poly = np.polyfit(np.log(lags),np.log(tau),1)
    result = poly[0]*2.0    기울기의 2배는?

    return result

def calcHalfLife(df):
    price = pd.Series(df)
    lagged_price = price.shift(1).fillna(method='bfill')
    delta = price - lagged_price
```


회귀 종목 추출 및 트레이딩 결과

결과

```
cursor1.execute('select * from stocks')
rows1 = cursor1.fetchall()
stock_data1 = pd.DataFrame(rows1, columns=['id', 'code', 'company', 'logistic', 'rf', 'svm', 'total_score', 'rank', 'way'])
stock_data1.set_index('id')
del stock_data1['id']
stock_data1['svm'].fillna('0')
stock_data1['total_score'] = stock_data1['logistic'] + stock_data1['rf'] + stock_data1['svm']
stock_data1['rank'] = stock_data1['total_score'].rank(ascending=False)
stock_data1['way'] = np.where(stock_data1['lrpred'] + stock_data1['rfpred'] + stock_data1['svmpred'] > 0, 'B', 'S')
print(stock_data1[stock_data1['rank'] <= 10])
```

[결과]

	code	company	logistic	rf	svmpred	total_score	rank	way
137	000370	한화손해보험	0.5517	0.5230	...	1.0	1.6149	10.0 B
240	180640	한진칼	0.5230	0.5632	...	1.0	1.6322	9.0 B
321	298690	에어부산	0.5152	0.5758	...	1.0	1.6365	8.0 B
505	103130	웅진에너지	0.5515	0.5515	...	1.0	1.6545	6.0 B
509	298040	효성중공업	0.5892	0.5021	...	1.0	1.6888	4.0 B
678	074610	나노메딕스	0.5599	0.5543	...	1.0	1.6741	5.0 B
708	006110	삼아알미늄	0.5320	0.5710	...	1.0	1.6490	7.0 B



결과로 나온 종목 10개

1위: 삼화 페인트
5위: 한진칼

해석:
주가 움직임을 보면
모델의 정확도가 50%인
것처럼 반은 우상향하는
움직임을 보이고
반은 우하향하는 확률로
나뉘는 것을 확인.

백오피스 정산 자동화

1. 매일 시세를 수집하는 웹크롤러
2. 엑셀 매크로 자동화
3. 계좌 정산용 스크래핑

3files	20191125
LastdayOfMonth	20191125
afternoon_setting	20191125
cwarling	deleted
daeshin	20190617
dollarPrice_DaeshinAPI	add newone
kospi_dollar	20191125
maincom	deleted
morning_setting	moved
readme	Update readme
shinhan	20191125
sinNong	deleted
slippage	deleted
sum12	20191125

readme
- 엑셀 자동화
1. 01_05dollarAccount
2. 3files
3. LastdayOfMonth
4. maincom
5. sum12
- 증권사 API 시세 및 크롤링
1. cwarling - 네이버 해외 주가지수
2. daeshin - 대신 api 로그인
3. dollarPrice_DaeshinAPI - 대신 api 달러 시세 수집(월별)
4. kospi_dollar - 대신 api 시세 수집(분봉)
5. shinhan - 신한 api에서 계좌 상태 조회
- HTS 로그인
1. afternoon_setting
2. morning_setting

해외 주가지수 네이
버에서 시세 크롤링



엑셀 매크로 자동화

신한 Api 로 계좌
스크래핑 작업

```
from urllib.request import urlopen
import bs4
from bs4 import BeautifulSoup
import webbrowser
import datetime as dt
import selenium
from selenium import webdriver
import time

#Basic setting
def date_format(d):
    d = str(d)
    yyyy = int(d.split('.')[0])
    mm = d.split('.')[1]
    dd = d.split('.')[2]

    rule = '{}/{}{}'.format(mm,dd,yyyy)
    return rule

index_cd1 = ['DJI@DJI','NAS@IXIC']
index_cd2 = ['NII@NII225']

def dow_nas(i):
    naver_index = 'https://finance.naver.com/world/sise.nhn?symbol=' + i

    source = urlopen(naver_index).read()

    import win32com.client
    import time
    import xlrd
    from xlrd import xldate_as_tuple
    from datetime import date
    import easygui

    try:
        pathToExcel = '/Desktop/달러피치.csv'
        pathToMacro = '../Templates/달러피치정산용.txt'
        macroname = 'autoarrange'

        with open(pathToMacro, "r") as myfile:
            macro=myfile.read()

        excel = win32com.client.Dispatch("excel.Application")
        xl_open = excel.Workbooks.Open(pathToExcel)

        #insert the macro-string into the excel file
        excelModule = xl_open.VBProject.VBComponents.Add(1)
        excelModule.CodeModule.AddFromStr(macro)

        #run the macro
        excel.Application.Run(macroname)

    finally:
        # save the workbook and close
        excel.Workbooks(1).Close(SaveChanges=1)
        excel.Application.Quit()

#오늘 날짜를 2019-01-01 형식
a = time.strftime('%Y-%m-%d',time.localtime(time.time()))

try:
    pathToExcel2 = '/달러피치관리.xlsm'
    xl = win32com.client.Dispatch("excel.Application")
    xl.DisplayAlerts = False
    xl.Visible = True
    wh = xl.Workbooks.Open(pathToExcel2)

    import sys, os
    import PyQt5.QtGui import *
    import PyQt5.QtCore import *
    import PyQt5.QAxContainer import *
    import PyQt5.QtWidgets import QApplication
    import PyQt5.QtWidgets import QMainWindow
    import PyQt5.QtWidgets import QPushButton
    import PyQt5.QtWidgets import QLineEdit
    import win32com.client
    import datetime

    class IndiWindow(QMainWindow):
        def __init__(self):
            super(IndiWindow, self).__init__()

            self.setWindowTitle("IndiExample")

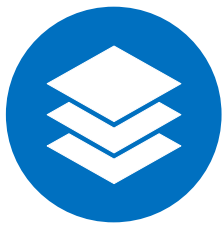
            # self.IndiTR = QAxWidget("GIEXPRTCONTROL64.GIExpertControl64Ctrl1.1")
            self.IndiTR = QAxWidget("GIEXPRTCONTROL.GIExpertControl1Ctrl1.1")

            # Indi API event
            self.IndiTR.ReceiveData.connect(self.ReceiveData)
            self.IndiTR.ReceiveSysMsg.connect(self.ReceiveSysMsg)
            self.rqidD = {}

            btnResearch = QPushButton("397", self)
            btnResearch.setGeometry(20, 50, 50, 20)
            btn_Series = [self.btn_Search,self.btn_Search1,self.btn_Search2]
            for i in btn_Series:
                btnResearch.clicked.connect(i) # 버튼을 누르면 'btn_Search' 함수가 실행됩니다.

            def btn_Search(self):
                ret = self.IndiTR.dynamicCall("SetQueryName(QString)", "SABA655Q1")
                ret = self.IndiTR.dynamicCall("SetSingleData(int, QString)", 0, "*****")
                ret = self.IndiTR.dynamicCall("SetSingleData(int, QString)", 1, "****") # 사보그브 하사 하사
```


02 DATA SCIENCE



뉴스 기사 와 주가 상관관계
N LP 처리

01

네이버 뉴스 크롤링과 종가 데이터를 Trade Station 틀에서 가져오기

02

K200 ETF 의 시장 수익률은 kospi 를 추종하는 것,
뉴스와 kospi 와의 연관도가 얼마나 되는 지 분석

03

공휴일 제거 및 형태소 분해 및 불용어 제거 처리.

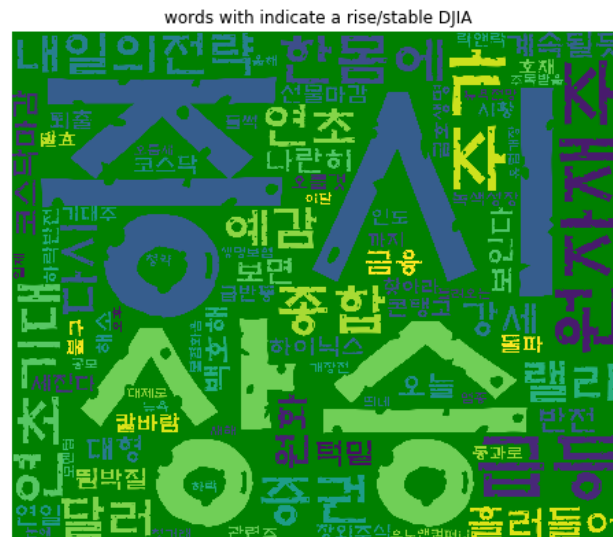
04

LogisticRegression , KNeighborsClassifier ,
RandomForestClassifier 세 가지 모델을 사용.



2010.01.01~2019.08.14까지
네이버 증권 섹션에서 selenium
크롤링을 함

Word cloud
주가 상승 시 키워드



주가 하락 시 키워드



뉴스 기사와 주가와 상관도

참조 자료: Kaggle 의 뉴스와 주가 데이터를 분석하는 대회에서 아이디어를 착안.

미국 주가 데이터와 텍스트 데이터를 바탕으로 제시한 train data Feature에서 감성 분석한 결과 이동평균 열을 발견.

한국 주가 데이터와 주요 포털 사이트의 뉴스 기사를 바탕으로 분석하고자 함.

베타 조정된 이동평균선 계산 위한 밑작업

A	B	C	D	E	F	G	H	I	J	K	L
Date	Close					Date	Close			베타조정된 수익	베타계수
2009-12-30	1682.77					2009-12-30	225.75				
2010-01-04	1696.14	0.79%				2010-01-04	227.1	0.60%		0.27%	0.73
2010-01-05	1690.62	-0.33%				2010-01-05	226.9	-0.09%		-0.32%	
2010-01-06	1705.32	0.87%				2010-01-06	228.45	0.68%		0.25%	
2010-01-07	1683.45	-1.28%				2010-01-07	225.65	-1.23%		-0.08%	
2010-01-08	1695.26	0.70%				2010-01-08	226.5	0.38%		0.44%	
2010-01-11	1694.12	-0.07%				2010-01-11	226.5	0.00%		-0.09%	
2010-01-12	1698.64	0.27%				2010-01-12	226.75	0.11%		0.21%	
2010-01-13	1671.41	-1.60%				2010-01-13	223.05	-1.63%		0.04%	
2010-01-14	1685.77	0.86%				2010-01-14	225.7	1.19%		-0.45%	
2010-01-15	1701.8	0.95%	0.011309			2010-01-15	226.75	0.47%	0.44%	0.66%	
2010-01-18	1711.78	0.59%	0.009221			2010-01-18	227.9	0.51%	0.35%	0.11%	
2010-01-19	1710.22	-0.09%	0.011593			2010-01-19	228.1	0.09%	0.53%	-0.24%	
2010-01-20	1714.38	0.24%	0.005313			2010-01-20	228.35	0.11%	-0.04%	0.18%	
2010-01-21	1722.01	0.45%	0.022905			2010-01-21	230.3	0.85%	2.06%	-0.56%	
2010-01-22	1684.35	-2.19%	-0.00644			2010-01-22	224.35	-2.58%	-0.95%	0.54%	
2010-01-25	1670.2	-0.84%	-0.01412			2010-01-25	223.5	-0.38%	-1.32%	-0.63%	
2010-01-26	1637.34	-1.97%	-0.03609			2010-01-26	218.85	-2.08%	-3.48%	0.15%	
2010-01-27	1625.48	-0.72%	-0.02748			2010-01-27	217	-0.85%	-2.71%	0.16%	
2010-01-28	1642.43	1.04%	-0.02571			2010-01-28	219.8	1.29%	-2.61%	-0.34%	
2010-01-29	1602.43	-2.44%	-0.05839			2010-01-29	213.85	-2.71%	-5.69%	0.37%	
2010-02-01	1606.44	0.25%	-0.06154			2010-02-01	214.65	0.37%	-5.81%	-0.17%	
2010-02-02	1595.81	-0.66%	-0.0669			2010-02-02	213.4	-0.58%	-6.44%	-0.11%	
2010-02-03	1615.02	1.20%	-0.05796			2010-02-03	215.85	1.15%	-5.47%	0.08%	
2010-02-04	1616.42	0.09%	-0.06132			2010-02-04	215.6	-0.12%	-6.38%	0.28%	
2010-02-05	1567.12	-3.05%	-0.0696			2010-02-05	209.45	-2.85%	-6.64%	-0.27%	

거래량과 이동평균선 feature 계산

```
import pandas as pd
import numpy as np
import datetime
from keras.models import Sequential
from keras.layers import Activation, LSTM, Dense, BatchNormalization
from keras.optimizers import sgd
import pandas as pd
import numpy as np

data = pd.read_csv('C:/Users/FOS_08/Documents/k200.csv')
pd.options.display.max_columns=20

windows = [5,10,20,60,120]
for i in windows:
    data['close_ma{}'.format(str(i))] = data['Close'].rolling(window=i).mean()
    data['volume_ma{}'.format(str(i))] = data['Vol'].rolling(window=i).mean()
```

거래량과 종가의 이동평균선을 계산

뉴스 기사와 주가와 상관도

```
from sklearn.metrics import roc_curve
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split, KFold

#한국어 텍스트 konlpy 로 형태소 추출한다음
#불용어 리스트에서 전부 제거시키기
okt = Okt()

trainheadline = []
for i in range(0, len(before_train.index)):
    trainheadline.append(' '.join(str(x) for x in before_train.iloc[i, 1:39]))

#형태소를 다 분리시킴 불용어 제거

stopwords = []
with open('../Documents/stop_words.txt', 'r') as res:
    k = res.readlines()
    for i in k:
        i = i.replace('\n', '')
        stopwords.append(i)

trainhead = []
for i in trainheadline:
    letters_only = okt.morphs(i)
    temp = []
    for ii in letters_only:
        if ii not in stopwords:
            letters = re.sub("[ㄱ-ㅎ-|가-힣]", " ", ii)
            letterss = re.sub('[\s]', '', letters)
            if letterss != '':
                temp.append(letterss)
    temp2 = ' '.join(str(x) for x in set(temp))
    trainhead.append(temp2)

#test 도 전처리

testheadline = []
```

3. 모델 학습

LogisticRegression, kNeighborsClassifiers, RandomForestClassifier 이 세모델로 학습 시킨 후 가장 정확도가 높게 나온 모델을 채택하기

```
basicvectorizer = CountVectorizer()
basictrain = basicvectorizer.fit_transform(trainhead)
basictest = basicvectorizer.transform(testhead)

#tfidfvectorizer로 할때
basicvectorizer = TfidfVectorizer()
basictrain = basicvectorizer.fit_transform(trainhead)
basictest = basicvectorizer.transform(testhead)

Classifiers = [
    LogisticRegression(C=1, solver='liblinear', max_iter=5000),
    KNeighborsClassifier(3),
    RandomForestClassifier(n_estimators=2000, max_depth=9)]
```

불용어 사전을 텍스트 파일로 만들어서 거른후

정규식으로 2차 필터

Countvectorizer 위해 리스트를 문자열로 묶기.

결론

feature_importance

	imp	col
8	2	sentiment
11	18	close_rolling10
7	19	close_beta10
10	20	open_raw10
12	20	close_rolling5
3	43	beta_close_return01
5	57	beta_open_return01
0	59	Open
9	64	close_raw10
2	70	Vol
14	88	JustCloseReturn01
1	94	Close
13	97	BetaCloseReturn01
16	171	JustOpenReturn
6	184	just_open_return
15	259	BetaOpenReturn01
4	485	just_close_return01

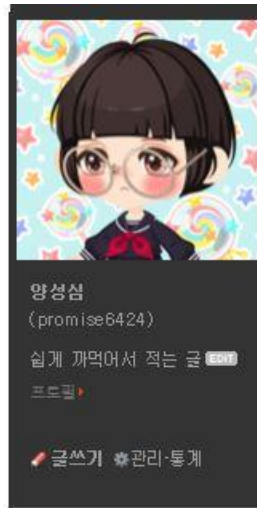
결론: 하지만, 뉴스 기사 분석을 feature로한 분석에서 정확도가 제일 낮게 나온.
하루 전 수익률이 다음날 방향과 제일 유사도가 높다고 나온

하루 전의 종가가 다음날의 움직임에 영향을 미친다.

Titanic_korean	deleted
je_date	New_folder
fifa	New_folder
funda_sales	add
jeju_transportation	add newone
job_recommndation	newly added
keras_image_classification	newly added
movie	add
smishing	newly added

Readme

- Dda-reong**
서울시 날씨와 따릉이 데이터로 1시간 후의 자전거 대수를 예측
- Financial_visualization**
네이버 각종 상품선물 및 국제주가지수 등의 데이터를 시각화
- NLP_pizza**
kaggle pizza 데이터를 자연어 처리해서 주문에 성공여부를 예측
- NLP_practice1~ NLP_stock_news**
네이버 신문기사를 크롤링해서 실제 종합주가지수와 얼마나 연관성이 있는지 예측
- Titanic_Korean**
타이타닉 승객 데이터로 생존여부 예측
- Fifa**
해외 축구 선수 이적료 예측 미션
- Funda_sales**
편다 상점의 고객 매출 데이터를 바탕으로 1분기 매출액 예상
- jeju-transportation**
제주도 버스 승차시간 별 승객 수 예측
- keras-image-classification**
케라스로 사람 얼굴 이미지 분류
- Movie**
영화 감독, 관객 수등의 데이터를 가지고 영화 관객 수 예측
- smishing**
금융 문자 텍스트를 분석하여 스미싱 여부 판단



- ### 카테고리
- 전체 보기 (239) **EDIT**
 - python (154)
 - Django (19)
 - data science (52)
 - tensorflow (17)
 - GUI
 - 그린 연습 (35)
 - 크롤링 (21)
 - 정규표현식 (4)
 - 웹
 - Excel (22)
 - VBA (8)
 - SQL (14)
 - 기타
 - hack (3)
 - JAVA (14)
 - jsp (0)
 - 안드로이드 (6)
 - 리눅스(백엔드) (1)



data science 52개의 글	조회수	작성일
LSTM 시계열 예측 검색비허용	1	2019. 11. 1.
naive bayes 분류기 검색비허용	0	2019. 9. 18.
movie review prediction 검색비허용	0	2019. 9. 6.
손실함수와 활성화 함수의 조합 검색비허용	0	2019. 9. 6.
stock_news_final 검색비허용	1	2019. 9. 5.

글관리 열기
5줄 보기

data science

movie review prediction

양성심

2019. 9. 6. 14:06

URL 복사
통계

#karas의 LSTM과 embedding을 어떻게 쌓아올리는 것인지 알기 위함.
natural language processs 가공을 어떻게 해서 모델에 집어넣어야 하는 지 알기 위함.

데이터 구성

train.csv
test.csv

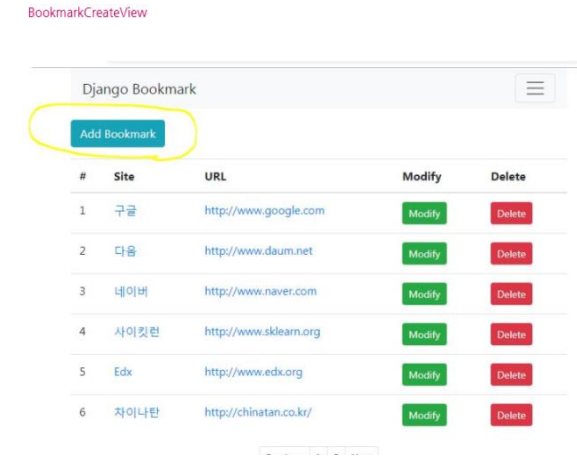
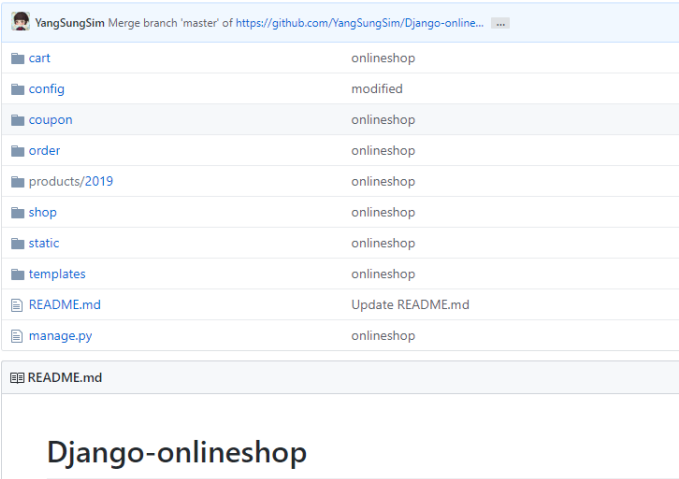
로 구성

```
import numpy as np
```

데이터 분석 학습 말미에 결과물 배포가 필요.
 ->파이썬 web과 밀접하게 연결된 장고를 사
 용 시작.



그 외 기술 블로그
 I love stackoverflow ♡



구현 포인트

- 네이버와 다음 아이디로 로그인 기능
- 댓글 달기 api 적용
- PG사 테스트 결제 기능 적용
- Sqlite3 이 아닌 AWS S3 데이터 베이스 연결 구현

- 1.북마크 페이지
- 2.쇼핑몰 페이지
- 3.기초 인스타그램 페이지

자유도가 높은 파이썬 백엔드 구현 언어
백엔드 기초로 미니 트위터 api 구현

가상 머신(vmware)으로 리눅스 우분투 환경
에서 Miniconda로 파이썬 환경을 갖추.

Mysql로 테이블 생성 및 데이터 입력 및
조회 쿼리를 각 url 주소로 연결 되는 함수에
연결.

로그인 시 비밀번호 암호화 및 프론트 엔드
와 연결.



리눅스의 자유롭고 빠른 실행속도 Good!

```
>>> app = Flask('test')
>>> exit()
(api) root@bt:~# conda deactivate
(base) root@bt:~# mkdir -p ~/Projects/api
(base) root@bt:~# cd ~/Projects/api
(base) root@bt:~/Projects/api# vim app.py
(base) root@bt:~/Projects/api# conda activate api
(api) root@bt:~/Projects/api# FLASK_APP=app.py FLASK_DEBUG=1 flask run
* Serving Flask app "app.py" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 285-043-953
127.0.0.1 - - [04/Dec/2019 13:00:37] "GET /ping HTTP/1.1" 500 -
Traceback (most recent call last):
```

```
(api) root@bt:~# http -v GET http://localhost:5000/ping
GET /ping HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: localhost:5000
User-Agent: HTTPie/1.0.3

HTTP/1.0 200 OK
Content-Length: 4
Content-Type: text/html; charset=utf-8
Date: Wed, 04 Dec 2019 04:16:59 GMT
Server: Werkzeug/0.16.0 Python/3.7.5

pong
```


자바 서버페이지를 통해 데이터베이스의 데이터를 웹으로 입출력 구현,
게시판, 회원 관련, 로그인 관련 웹 어플리케이션 제작

구현 포인트

실무형 코드를 기반으로 Eclipse 개발
툴과 Java Server Page를 통해 웹으로
입출력

```
package jsp_20191209;

import java.lang.reflect.Field;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

public class MemberDTO {
    /* 회원 아이디 */
    private String id;
    /* 회원 패스워드 */
    private String pw;
    /* 회원 이름 */
    private String name;
    /* 회원 성별 */
    private String gender;
    /* 회원 주소 */
    private String address;

    public MemberDTO() {
        System.out.println("기본생성자");
    }

    // 오버로딩 생성자 setter "s" 의미. 일체형: 인스턴스 생성전에 멤버필드 초기화
    public MemberDTO(String id, String pw, String name, String gender, String address) {
        this.id = id;
        this.pw = pw;
        this.name = name;
        this.gender = gender;
        this.address = address;
    }

    // Map -> MemberDTO
    public MemberDTO(Map<String,String[]> map) {
```

```
1. <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2. <%@ taglib prefix="c" uri="http://java.sun.com/jsp/ist/core" %>
3. <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/ist/fmt" %>
4. <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/ist/functions" %>
5. <!DOCTYPE html>
6. <html lang="ko">
7. <head>
8. <meta charset="UTF-8">
9. <title>전체 회원정보 조회(페이지)</title>
10. <!-- bootstrap CSS -->
11. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
12. <!-- jQuery -->
13. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
14. <!-- popper.js -->
15. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
16. <!-- bootstrap javascript -->
17. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
18. <style>
19. table#view_paging {
20.     font-size:9pt;
21. }
22. </style>
23. </head>
24. <body>
25.
26. <!-- 회원정보 출력부 -->
27. <div id="wrap" style="width:80%;margin:auto">
28.     <table id="view_paging" class="table table-hover">
29.         <!-- 표제 -->
30.         <tr>
31.             <th>번호</th>
```

회원 정보							
번호(idx)	이름(논리 필드명)	필드 형태	아이디(id)	이름(name)	기본값	필수여부	정규표현식(regex)
1	아이디	text	memberId	memberId		필수	[a-zA-Z]{1}\w{?,19}
2	패스워드	password	memberPassword	memberPassword		필수	(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*\W){8,20}
4	이름	text	memberName	memberName		필수	[가-힣]{2,25}
5	성별	text	memberGenderMale(남) memberGenderFemale(여)	memberGenderMale	남(m)/여(f)	필수	
6	메일	text	memberEmail	memberEmail		필수	-Z0-9_+.-]+@([a-zA-Z0-9-]+\.)+[a-zA-Z0-
7	연락처	text	memberPhone	memberPhone		필수	01\d{1}-\d{3,4}-\d{4}
8	생년월일	text	memberBirth	memberBirth		필수	\d{4}-\d{2}-\d{2}
9	우편번호	text	memberZip	memberZip		선택	\d{5}
10	주소 검색	button	joinAddressSearchBtn	joinAddressSearchBtn	주소검색		
10	기본주소	text	memberAddressBasic	memberAddressBasic		선택	[w \W 가-힣 / - <]{2,200}
11	상세주소	text	memberAddressDetail	memberAddressDetail		선택	[w \W 가-힣 / -]{2,100}
12	회원가입	submit	joinSubmitBtn	joinSubmitBtn	회원가입		
13	가입취소	reset	joinResetBtn	joinResetBtn	가입취소		

업비트 개발자 센터에서 발급받은 API.
웹소켓 등으로 실시간 데이터 스트림
mysql DB 동기화. 거래 기록 및 주문을
redis, celery, supervisord 등으로 주기적
job 생성 고민. 해당 거래 및 자산 현황 보
고서를 telegram 알림 메시지로 주기적인
알림 로직까지 구현.



향후 목표:
데이터 파이프라인 안정적 확보를 위한
서버 지식 강화, Spark 및 kafka, ELK 같은
빅데이터 처리 지식 습득

```

Cdrive > yangsungsim > PycharmProjects > upbit > 35_version > scheduler > invalid > web
1 import db.mysql_handler
2 import websocket
3 import json
4 import pymysql
5
6 try:
7     import thread
8 except ImportError:
9     import _thread as thread
10 import time
11 result = []
12 def on_message(ws, message):
13     get_message = json.loads(message.decode('utf-8'))
14     result.append(get_message)
15
16 def on_error(ws, error):
17     print(error)
18
19 def on_close(ws):
20     print("close")
21
22 def on_open(ws):
23     def run(*args):
24         with open(r'/code_list_krw.txt', 'r') as f:
25             for line in f:
26                 line = line.strip()
27                 if line:
28                     line = line.split(',')
29                     code = line[0]
30                     type = line[1]
31                     if type == 'buy':
32                         ws.send(json.dumps({'code': code, 'type': 'buy'}))
33                     elif type == 'sell':
34                         ws.send(json.dumps({'code': code, 'type': 'sell'}))
35
36     ws.on_message = on_message
37     ws.on_error = on_error
38     ws.on_close = on_close
39     ws.on_open = on_open
40     ws.run_forever(sslopt={'cert_reqs': ssl.CERT_NONE})
41     run()

```

```

def top_30_currency(k, cost):
    volume_dict = {}
    volume_cost_dict = {}
    for i in Search.code_list:
        try:
            volume = Search.day_to_month_candle('daily', i)
            volume_cost = Search.day_to_month_candle('daily', i)

        except IndexError:
            time.sleep(2)
            volume = Search.day_to_month_candle('daily', i)
            volume_cost = Search.day_to_month_candle('daily', i)

        volume_dict[float(volume[0])] = i
        volume_cost_dict[float(volume_cost[0])] = i

    sorted_key = sorted(volume_dict.items(), key=operator.itemgetter(0))
    sorted_key2 = sorted(volume_cost_dict.items(), key=operator.itemgetter(0))

    hubo1 = [i[1] for i in sorted_key[:k]] #거래량 상위 k개
    hubo2 = [i[1] for i in sorted_key2[:k] if i[0]>cost] #종목리스트가 반환 될거
    return set(hubo1)&set(hubo2)

```



04 Finance



전공 : 금융보험학과

- VBA 프로그래밍 수업 이수
- MSSQL 데이터 베이스
- 가상의 편의점 객체 구조 를 만들어 고객 정보를 입력 조회 하는 쿼리문 작성
- 과생상품 , 경제학 원론, 손해보험 외 수강 (총 학점 4.36 / 4.5)



핀테크 아카데미

- 데이터 분석 및 창업관련 지식 습득
- 머신러닝,딥러닝, 블록체인 기술 동향
- 카카오 기업 가치 모형 평가
- 핀테크 기업 재무 분석(30 % 기여 및 발표)



금융자격증

- 투자자산운용사
- 외환전문역 1종
- 은행 FP
- 증권투자상담사
- 전산세무2급

홍익대학교
서울캠퍼스: 04066 서울특별시 마포구 와우산로 94
Tel. 02-320-1114
세종캠퍼스: 30016 세종특별자치시 조치원읍 세종로 2639
Tel. 044-860-2114

성 적 증 명 서

홍익대학교
서울캠퍼스: 04066 서울특별시 마포구 화우산로 94
Tel. 02-320-1114
새종캠퍼스: 30016 세종특별자치시 조치원읍 새종로
Tel. 044-860-2114

성적 증명서

HONGIK UNIVERSITY
Seoul Campus:
94 Wusang-ro, Mapo-gu, Seoul, 04066, Korea
Sejong Campus:
2639 Sejong-ro, Jochiwon-eup,
Sejong, 30016, Korea

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

호

<[illegible]

위의 사실을 증명합
2017년 01월 11일

홍익대학교 교

[인터넷 발급] 우리 대학 및 www.certpia.com 원본대조 메뉴에서 인터넷 발급번호로 입학(발급일로부터 90일 이내) 또는 스마트폰 원본대조앱으로 증명서 하단의 QR코드를 촬영하여 원본대조 확인 가능, 스마트폰 원본대조앱은 앱스토어/구글에서 원본대조로 검색 다운로드할 수 있습니다.

(Internet no) 3734307031952276

위·변조방지용
원본대조코드
특허 10-1039390

1.

중 격 합

Certificate

성명 : 양성심
Full Name : Yang SungSim
생년월일 : 1993. 01. 03
Date of Birth : January 03, 1993
자격시험명 : 투자자산운용사
Name of Examination : Certified Investment Manager
합격증번호 : 17-017923
Certificate Number :

위 사람은 한국금융투자협회가 실시한 제18회 투자자산운용사 자격시험에 합격하였기에 본 증서를 수여함.

This is to certify that the above mentioned person has passed the 18th Certified Investment Manager Qualification Examination conducted by the Korea Financial Investment Association.

2017년 12월 08일
December 08, 2017

한국금융투자협회 회장
Korea Financial Investment Association


발행일(Date of Issue) : 2019. 12. 19 (December 19, 2019)
 발행번호(Issue Number) : 2017120214105907255207255272951
 상기 내용을 본잡지(本誌)에 앞서 투고한 내물이므로, 반드시 사용될 한자 한글출력 후호
 어부출판사(주)에서 발행되는 잡지(雜誌)에 사용될 한자 한글출력 후호(後號)를 확인하시기 바랍니다.

100-2111 3000 5564

제 1609018831 호

자 격 증


격 증
Certificate


 종 목 국가공인 외환전문역 | 종
 성 명 양성심
 생년월일 1993. 01. 03
 유효기간 2016.12.09 ~ 2019.12.31

위 사람은 우리 원 소정의 자격검정시험에 합격하고 자격 취득에 필요한 전치료 미청으로 인하여 자격정지 상태에 있습니다.


- 시험합격일 : 2016년 12월 09일 (제 27 회 시험)
- 자격등록일 : 2016년 12월 09일

2017년 01월 11일

 한국금융연수원장

2.


자 격 증

	종 목	국가공인 자산관리사(FP)
	성 명	양성심
	생년월일	1993. 01. 03
	유효기간	2016.11.18 ~ 2019.12.31

위 사람은 우리 원 소정의 자격검정시험에 합격하고 자격 취득에 필요한 절차를 마쳤으므로 이 자격증을 수여합니다.

- 시험합격일 : 2016년 11월 18일 (제 39 회 시험)
- 자격등록일 : 2016년 11월 18일

2017년 01월 11일

 한국금용연수원장

4

합 격 증

성명 : 양성심
Full Name : Yang Sangsim
생년월일 : 1993. 01. 03
Date of Birth : January 03, 1993
자격시험명 : 펀드투자상담사 (한국투자증권)
Name of Examination : Certified Fund Investment Advisor
합격증번호 : 14-026995
Certificate Number

위 사람은 한국금융투자협회가 실시한 제25회 펀드투자상담사(전과목 응시) 자격시험에 합격하였기에 본 증서를 수여함.

This is to certify that the above mentioned person has passed the 25th Certified Fund Investment Advisor Qualification Examination conducted by the Korea Financial Investment Association.

2014년 10월 02일
October 02, 2014

한국금융투자협회 회장
Korea Financial Investment Association

발행일(Date of Issue): 2017. 01. 11 (January 11, 2017)
발행번호(Issue Number): 2014090251651321346632346632346421
본 내용은 발간일(출판일) 당시 유효한 내용이며, 반드시 서문일 전체 판권표에 유효
무효 공표주지참의자서함접수번호를 기재하시(license holder, or so)해서 확인하시기 바랍니다.

MSSQL 실습팀과제

데이터베이스 학기과제

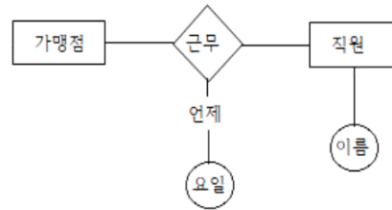
-GS25편의점 사례연구-

과목명 : 데이터베이스
담당교수 : 정일주 교수님
제출일자 : 2015.11.30.(월)
제출자 : 검은콩 조
조원 : 양성식, 유상민

3. 지역 개체연관 모델의 설계

앞서 제시한 Q1에서 Q9까지의 가시적 문의에 대하여, 각각 L1에서 L9까지의 지역 개체연관모델을 작성한다.

Q1 : 매장에서 근무하는 이름이 '전예은'인 직원이 근무하는 요일을 제시하라.
L1:



Q2 : 제품명이 첫술을 취급하고 있는 가맹점의 이름과 제품의 가격을 제시하라.
L2:



```
use GS편의점
go
insert into 가맹점
values ('송탄우림','031-691-7082')
insert into 가맹점
values ('평택합정','031-652-7627')
insert into 가맹점
values ('원평행복','031-661-3451')

use GS편의점
go
insert into 가맹점
values ('송탄우림','031-691-7082')
insert into 가맹점
values ('평택합정','031-652-7627')
insert into 가맹점
values ('원평행복','031-661-3451')
```

핀테크 아카데미 기업가치

카카오 기업가치 평가:

2016~2018 까지의 재무 정보 수집 (Dart.fss.co.kr 외)
유동자산, 유형 자산을 기초로 Noplat 및 FCF 도출

카카오의 사업 구조 및 영업 수익 구성
비율 분석을 통한 비정상 이벤트 반영

● 요약 재무상태 (단위 : 백만원)

구분	2016	2017	2018
유동자산	511,067	632,508	1,261,480
비유동자산	4,318,230	4,324,369	4,941,359
자산총계	4,829,297	4,956,878	6,202,839
유동부채	813,859	948,931	1,250,706
비유동부채	640,280	450,237	200,790
부채총계	1,454,139	1,399,168	1,451,496
자본금	33,858	34,004	41,744
자본총계	3,375,158	3,557,710	4,751,343

● 요약 손익계산서 (단위 : 백만원)

구분	2016	2017	2018
매출액	861,205	978,700	1,173,925
매출총이익	861,205	978,700	1,173,925
영업이익	70,995	97,551	122,366
영업외수익	36,229	287,929	138,785
영업외비용	54,599	138,054	78,485
법인세비용차감전순이익	52,625	247,426	182,665
법인세비용	14,453	29,872	92,435
당기순이익	38,171	217,554	90,230

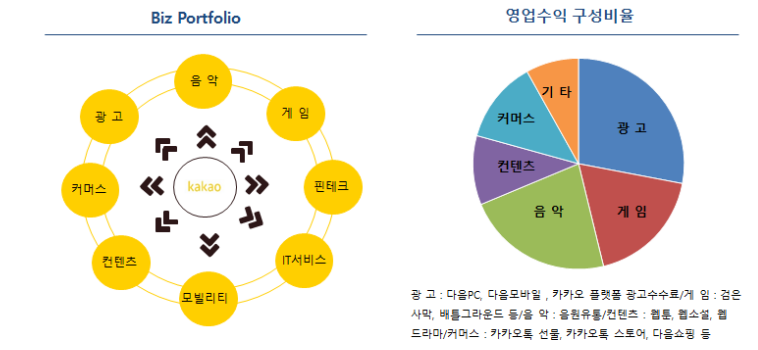
■ 전체재무비율

(단위 : %)				
구분	계정명	2016-12-31	2017-12-31	2018-12-31
성장성	매출액증가율	-0.10	13.64	19.95
	영업이익증가율	-41.13	37.41	25.44
	순이익증가율	-47.64	469.94	-58.53
	총자산증가율	54.93	2.64	25.14
	자기자본증가율	31.08	5.41	33.55
	법인세차감전순이익증가율	-48.00	370.17	-26.17
	유동자산증가율	-28.87	23.76	99.44
	유형자산증가율	-3.10	2.07	13.42
	재고자산증가율	441.28	58.15	-52.07
	종업원수증가율	9.01	-5.49	6.12
	매출원가율	0.00	0.00	0.00
	물류원가및관리비율	91.76	90.03	89.58
	영업이익률	8.24	9.97	10.42
수익성	법인세차감전순이익률	6.11	25.28	15.56
	당기순이익률	4.43	22.23	7.69
	ROIC	27.88	42.34	46.69
	ROE	1.28	6.28	2.17
	EBITDA마진율	16.71	17.11	17.19
	총자산법인세차감전순이익률	1.32	5.06	3.27
	순금융원가부담율	-0.96	0.87	-3.76
	자본금법인세차감전순이익률	164.57	729.20	482.30
	감가상각률	3.81	3.69	3.17
	법인세차감전순이익이자보상율	4.37	16.63	13.91
	그외유가비율			



Biz Portfolio

□ 카카오는 ICT기술 및 Platform 부문에서 다양한 Biz portfolio 보유



2017년 종속기업 투자 처분이익 발생.

재무 비율의 증가율의 기하 평균을 사용한 미래 재무 수치 임의 예상.

핀테크 아카데미 기업가치

3

과거 재무제표 분석

□ 플랫폼, 콘텐츠 부문 등 매출 증가로 영업이익 증가추세

재무제표

(단위 : 백만원)

재무상태표	2016	2017	2018
유동자산	511,067	632,508	1,261,480
비유동자산	4,318,230	4,324,369	4,941,359
자산총계	4,829,297	4,956,878	6,202,839
유동부채	813,859	948,931	1,250,706
비유동부채	640,280	450,237	200,790
부채총계	1,454,139	1,399,168	1,451,496
자본총계	3,375,158	3,557,710	4,751,343

손익계산서	2016	2017	2018
매출액	861,205	978,700	1,173,925
영업이익	70,995	97,551	122,366
영업외수익	36,229	287,929	138,785
영업외비용	54,599	138,054	78,485
법인세비용	14,453	29,872	92,435
당기순이익	38,171	217,554	90,230

재무비율

(단위 : %, 배, 회, 백만원)

구분	2016	2017	2018
성장성			
총자산증가율	54.93	2.64	25.14
매출액증가율	-0.10	13.64	19.95
순이익증가율	-47.64	469.94	-58.52
수익성			
영업이익률	8.24	9.97	10.42
ROE	1.28	6.28	2.17
ROIC	18.79	26.33	23.64
안정성			
부채비율	43.08	39.33	30.55
이자보상배수	4.54	6.16	8.65
차입금의존도	20.10	15.56	9.75
활동성			
매출채권회전율	9.40	11.25	15.73
재고자산회전율	417.44	217.72	288.17
총자산회전율	0.22	0.20	0.21

고속성장 및 성장률이 할인율 초과.
2018.12.31 기준 기업가치 계산

4

미래 재무제표 추정을 위한 가정

□ 재무제표 추정을 위해 2017년, 2018년 증감율 평균을 사용

(단위 : %)

구분	2017.12.31	2018.12.31	Input	비고
매출액	114%	120%	117%	
영업관련비용	101%	131%	115%	
감가상각비	253%	57%	120%	
영업이익(EBIT)	137%	125%	131%	
법인세 @40%	207%	309%	253%	
세후영업이익(NOPLAT)	570%	41%	154%	
유동자산(영업관련)	124%	199%	157%	
유동부채(영업관련 무이자발생 부채)	117%	132%	124%	
유형자산(취득원가)	102%	113%	108%	
감가상각충당금	53%	39%	45%	
할인율			5%	

5

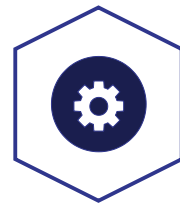
미래 재무제표 추정 결과

□ DCF법으로 산정한 카카오의 기업가치

(단위 : 백만원)

구분	2016.12.31	2017.12.31	2018.12.31	2019.12.31	2020.12.31	2021.12.31	2022.12.31	2023.12.31	2024.12.31
매출액	861,205	978,700	1,173,925	1,370,589	1,600,200	1,868,276	2,181,263	2,546,683	2,973,321
영업관련비용	735,611	743,095	973,074	1,119,166	1,287,193	1,480,445	1,702,712	1,958,349	2,252,365
감가상각비	54,599	138,054	78,485	94,100	112,821	135,266	162,178	194,443	233,127
영업이익(EBIT)	70,995	97,551	122,366	160,649	210,908	276,892	363,519	477,247	626,556
법인세 @40%	14,453	29,872	92,435	233,763	591,173	1,495,044	3,780,884	9,561,644	24,180,862
세후영업이익(NOPLAT)	38,171	217,554	90,230	138,727	213,289	327,927	504,180	775,166	1,191,800
유동자산(영업관련)	511,067	632,508	1,261,480	1,381,900	3,113,744	4,891,974	7,685,735	12,074,988	18,970,301
유동부채(영업관련 무이자발생 부채)	813,859	948,931	1,250,706	1,550,452	1,932,035	2,382,672	2,983,707	3,661,596	4,539,138
유형자산(취득원가)	38,171	217,554	90,230	138,727	213,289	327,927	504,180	775,166	1,191,800
유형자산(장부가액)	1,129,051	698,107	425,122	339,441	311,889	311,468	324,226	343,928	367,826
영업유무투자자산(Invested capital)	826,259	381,684	435,896	770,888	1,503,598	2,820,770	5,056,255	8,757,320	14,799,589
투자자산이익률(Return on invested capital)	18.8%	26.3%	23.6%	31.8%	27.7%	21.8%	17.9%	15.3%	13.6%
재투자율(Plowback rate)		-204.4%	60.1%	241.5%	343.5%	401.7%	443.4%	477.5%	507.0%
세후영업이익(NOPLAT)	38,171	217,554	90,230	138,727	213,289	327,927	504,180	775,166	1,191,800
(-) 영업유무투자자산 증가분		-444,575	54,212	334,992	732,710	1,317,172	2,235,486	3,701,064	6,042,269
= 잉여현금흐름(Free cash flow)	38,171	662,129	36,018	-196,266	-519,421	-989,245	-1,731,305	-2,925,899	-4,850,470
잉여현금흐름 증가율		1635%	-95%	-645%	165%	90%	75%	69%	66%
잔존가치(Terminal value)			122,934						
잉여현금흐름 + 잔존가치	38,171	662,129	158,951						
현재가치 계수	95.24%	90.70%	86.38%						
현재가치	36,353	600,571	137,308						
누적현재가치	36,353	636,924	774,232						
기업가치(Value of the firm)		1,447,510							

□ 능력 스킬



80

금융 전공입니다.
자산운용 업종 근무
경력 2년.



60

프로그래밍
파이썬 자동화
분석 과정 메모리 한계
극복 위해 spark, 클라
우드 컴퓨팅 학습.



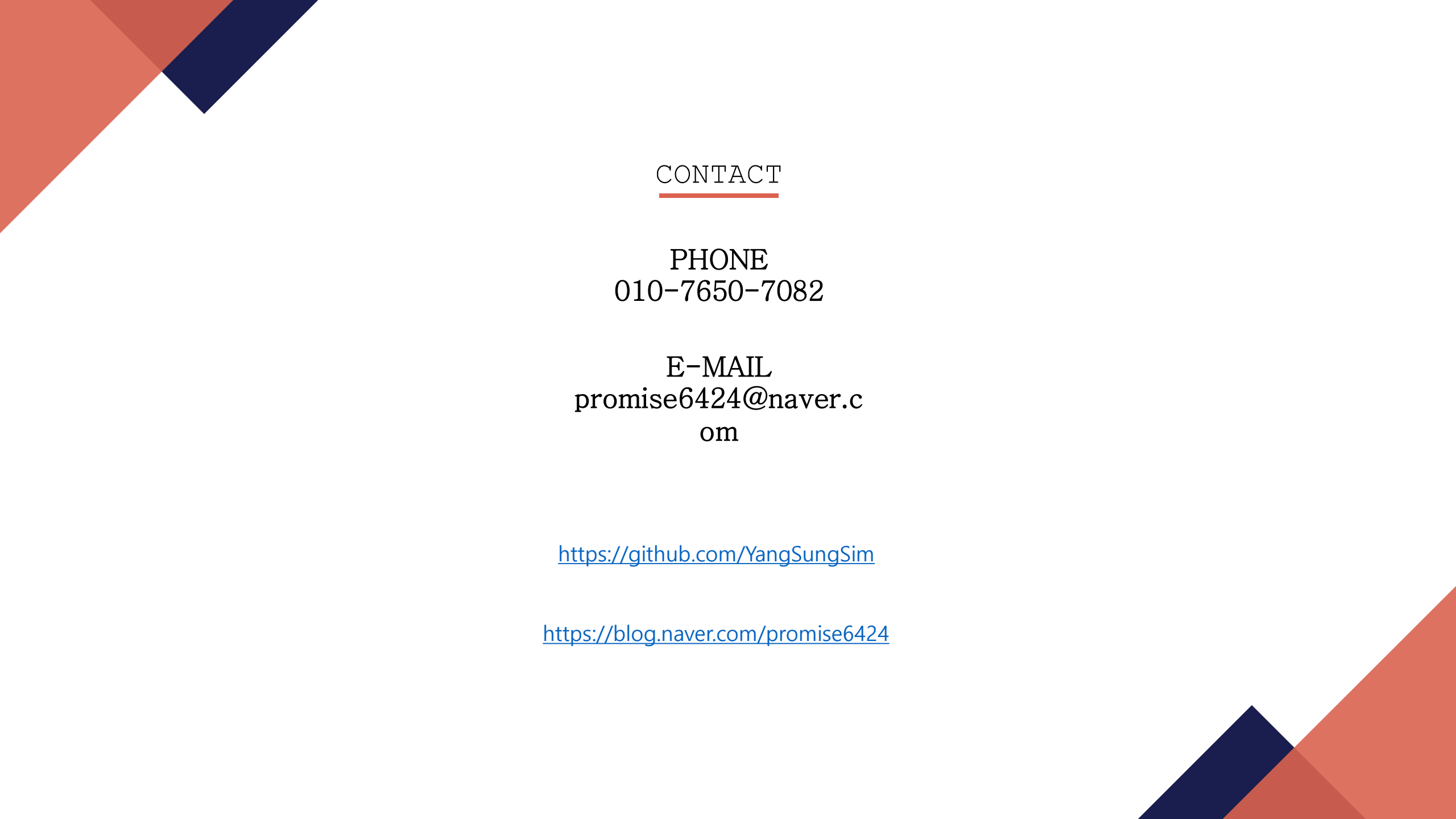
50

장고 ,flask 로
back-end 구현.
Java_jsp, oracle로
구현



70

데이터 사이언스
데이터 분석 개인 연
구 분야(자연어처리,
추천분석)



CONTACT

PHONE
010-7650-7082

E-MAIL
promise6424@naver.c
om

<https://github.com/YangSungSim>

<https://blog.naver.com/promise6424>



THANK YOU

Promising Developer