피처 엔지니어링

```
import FinanceDataReader as fdr
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

pd.options.display.float_format = '{:,.3f}'.format
```

가설 검정에서 만들었던 모든 피쳐(변수)를 정리해 보겠습니다. 이제 예측 모델링을 위한 데이터가 준비되었습니다. 예측모델링에 활용한 데이터의 기간은 2021년 1월 5일부터 2022년 3월 24일까지입니다.

```
mdl_data = pd.read_pickle('mdl_data.pkl') # 수익률 결과값이 있는 데이터
mdl_data.head()
print(mdl_data.index.min(), mdl_data.index.max())
```

```
2021-01-05 2022-03-24
```

가설검정에서 만들었던 모든 피쳐를 정리합니다. 단, *"5일 이동평균선이 종가보다 위에 있다"* 는 유의미하지 않았으므로 제외입니다. 결과를 feature_all 이라는 데이터프레임에 저장합니다.

```
kosdaq_list = pd.read_pickle('kosdaq_list.pkl')

feature_all = pd.DataFrame()

for code, sector in zip(kosdaq_list['code'], kosdaq_list['sector']):

    data = mdl_data[mdl_data['code']==code].sort_index().copy()


    # 가격변동성이 크고, 거래량이 몰린 종목이 주가가 상승한다
    data['price_mean'] = data['close'].rolling(20).mean()
    data['price_std'] = data['close'].rolling(20).std(ddof=0)
    data['price_z'] = (data['close'] - data['price_mean'])/data['price_std']
    data['volume_mean'] = data['volume'].rolling(20).mean()
    data['volume_std'] = data['volume'].rolling(20).std(ddof=0)
    data['volume_z'] = (data['volume'] - data['volume_mean'])/data['volume_std']


    # 위꼬리가 긴 양봉이 자주 발생한다.
    data['positive_candle'] = (data['close'] > data['open']).astype(int) # 양봉
    data['high/close'] = (data['positive_candle']==1)*(data['high']/data['close'] >
1.1).astype(int) # 양봉이면서 고가가 종가보다 높게 위치
    data['num_high/close'] =  data['high/close'].rolling(20).sum()
    data['long_candle'] = (data['positive_candle']==1)*(data['high']==data['close'])*\
    (data['low']==data['open'])*(data['close']/data['open'] > 1.2).astype(int) # 장대 양봉
을 데이터로 표현
    data['num_long'] =  data['long_candle'].rolling(60).sum() # 지난 20 일 동안 장대양봉의
갯 수


     # 거래량이 종종 터지며 매집의 흔적을 보인다
    data['volume_mean'] = data['volume'].rolling(60).mean()
    data['volume_std'] = data['volume'].rolling(60).std()
    data['volume_z'] = (data['volume'] - data['volume_mean'])/data['volume_std'] # 거래량은
종목과 주가에 따라 다르기 때문에 표준화한 값이 필요함
    data['z>1.96'] = (data['close'] > data['open'])*(data['volume_z'] > 1.65).astype(int)
# 양봉이면서 거래량이 90%신뢰구간을 벗어난 날
    data['num_z>1.96'] =  data['z>1.96'].rolling(60).sum()  # 양봉이면서 거래량이 90%신뢰구
간을 벗어난 날을 카운트

    # 주가지수보다 더 좋은 수익율을 보여준다
    data['num_win_market'] = data['win_market'].rolling(60).sum() # 주가지수 수익율이 1 보다
작을 때, 종목 수익율이 1 보다 큰 날 수
    data['pct_win_market'] = (data['return']/data['kosdaq_return']).rolling(60).mean() #
주가지수 수익율 대비 종목 수익율


    # 동종업체 수익률보다 더 좋은 수익율을 보여준다.
    data['return_mean'] = data['return'].rolling(60).mean() # 종목별 최근 60 일 수익율의 평
균
    data['sector'] = sector

    data['max_close']  =
data[['close_r1','close_r2','close_r3','close_r4','close_r5']].max(axis=1) # 5 영업일 종가
수익율 중 최고 값
    data['mean_close']  =
data[['close_r1','close_r2','close_r3','close_r4','close_r5']].mean(axis=1) # 5 영업일 종가
수익율 중 최고 값
    data['min_close']  =
data[['close_r1','close_r2','close_r3','close_r4','close_r5']].min(axis=1) # 5 영업일 종가
수익율 중 최저 값

    data = data[(data['price_std']!=0) & (data['volume_std']!=0)]

    feature_all = pd.concat([data, feature_all], axis=0)


feature_all['sector_return'] = feature_all.groupby(['sector', feature_all.index])
['return'].transform(lambda x: x.mean()) # 섹터의 평균 수익율 계산
feature_all['return over sector'] = (feature_all['return']/feature_all['sector_return']) #
섹터 평균 수익률 대비 종목 수익률 계산
feature_all.dropna(inplace=True) # Missing 값 있는 행 모두 제거


# 최종 피처 및 수익률 데이터만으로 구성
feature_all = feature_all[['code',
'sector','return','kosdaq_return','price_z','volume_z','num_high/close','num_long','num_z>
1.96','num_win_market','pct_win_market','return over
sector','max_close','mean_close','min_close']]
feature_all.to_pickle('feature_all.pkl')
```

이제 모델링을 위한 데이터 준비가 끝났습니다. 간단한 프로파일을 뽑아봅니다. 평균과 표준편차 값을 보고, 피처들이 제대로 생성되었는 지 확인합니다. 그리고 price_z 와 volum_z 는 같이 분석했을 때 유의미했다는 사실을 기억하면 좋겠습니다.

```
feature_all = pd.read_pickle('feature_all.pkl')
feature_all.describe(percentiles=[0.05, 0.1, 0.9,
0.95]).style.set_table_attributes('style="font-size: 12px"').format(precision=3)
```

	return	kosdaq_return	price_z	volume_z	num_high/close	num_long	num_z>1.96	num_win_market	pct_win_market	return over sector	max_close	mean_close	min_
count	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	329307.000	32930
mean	1.000	1.000	-0.106	-0.058	0.126	0.017	1.942	7.559	1.001	1.000	1.033	1.001	
std	0.035	0.013	1.316	1.063	0.387	0.131	2.015	2.840	0.004	0.030	0.071	0.053	
min	0.326	0.963	-4.359	-2.032	0.000	0.000	0.000	0.000	0.976	0.379	0.700	0.509	
5%	0.955	0.978	-2.083	-0.868	0.000	0.000	0.000	3.000	0.995	0.963	0.969	0.931	
10%	0.967	0.983	-1.729	-0.722	0.000	0.000	0.000	4.000	0.996	0.974	0.981	0.949	
50%	1.000	1.001	-0.226	-0.311	0.000	0.000	1.000	7.000	1.000	0.998	1.017	0.998	
90%	1.033	1.015	1.672	0.687	1.000	0.000	5.000	11.000	1.005	1.026	1.096	1.053	
95%	1.051	1.021	2.127	1.681	1.000	0.000	6.000	12.000	1.008	1.042	1.143	1.080	
max	1.300	1.046	4.359	7.617	5.000	2.000	15.000	22.000	1.043	1.399	3.703	2.346	

By KHS