

종목 추천 프로세스

완성된 모델을 이용하여 종목 추천을 받는 프로세스를 순서대로 만들어보겠습니다. 오늘이 2022년 4월 1일라고 가정하고 어떤 종목들이 추천되는 지 보겠습니다. 4월 1일 장 마감 후 프로그램을 돌려 추천 종목을 받고, 익일(4월 2일) 날 4월 1일의 종가에 매수를 하는 전략입니다.

```
import FinanceDataReader as fdr
import matplotlib.pyplot as plt
%matplotlib inline

import pandas as pd
import numpy as np
import requests
import bs4

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

오늘이 2022년 4월 1일라고 가정하고 어떤 종목들이 추천되는 지 보겠습니다. 먼저 오늘 기준으로 100 일전 날짜를 timedelta 를 이용해 찾습니다.

```
import datetime
today_dt = '2022-04-01'
today = datetime.datetime.strptime(today_dt, '%Y-%m-%d')
start_dt = today - datetime.timedelta(days=100) # 100 일전 데이터 부터 시작 - 피쳐 엔지니어링은 최소 60 개의 일봉이 필요함
print(start_dt, today_dt)
```

```
2021-12-22 00:00:00 2022-04-01
```

위 코드에서 찾은 시작일부터 오늘까지 종목별로 일봉을 가져와서 데이터셋을 구성합니다. 총 67개의 일봉이 있습니다. 입력 피쳐를 생성하기 위해서는 최소한 60일의 데이터가 필요합니다.

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

price_data = pd.DataFrame()

for code, name in zip(kosdaq_list['code'], kosdaq_list['name']): # 코스닥 모든 종목에
    # 대하여 반복
    daily_price = fdr.DataReader(code, start = start_dt, end = today_dt) # 종목, 일봉, 데이터 갯수
    daily_price['code'] = code
    daily_price['name'] = name
    price_data = pd.concat([price_data, daily_price], axis=0)

price_data.index.name = 'date'
price_data.columns= price_data.columns.str.lower() # 컬럼 이름 소문자로 변경

print(price_data.index.nunique())
```

```
67
```

주가지수 데이터를 가져오고, 일봉데이터에 추가합니다. 그리고 결과물을 merge 라는 이름으로 저장합니다. FinanceDataReader 에서 지수데이터가 수집이 안 될 경우, 야후 파이낸스를 이용할 수 도 있습니다.

```
kosdaq_index = fdr.DataReader('KQ11', start = start_dt) # 데이터 호출
kosdaq_index.columns = ['close', 'open', 'high', 'low', 'volume', 'change'] # 컬럼명 변경
kosdaq_index.index.name='date' # 인덱스 이름 생성
kosdaq_index.sort_index(inplace=True) # 인덱스(날짜) 로 정렬
kosdaq_index['kosdaq_return'] = kosdaq_index['close']/kosdaq_index['close'].shift(1)
# 수익율 : 전 날 증가대비 당일 증가

merged = price_data.merge(kosdaq_index['kosdaq_return'], left_index=True,
right_index=True, how='left')
merged.to_pickle('merged.pkl')
```

야후 파이낸스에서 지수 데이터 수집은 아래와 같이 할 수 있습니다.

```
import yfinance as yf

kosdaq_index = yf.download('^KQ11', start = start_dt) # 데이터 호출
kosdaq_index.columns = ['open', 'high', 'low', 'close', 'adj_close', 'volume'] # 컬럼명 변경
kosdaq_index.index.name='date' # 인덱스 이름 생성
kosdaq_index.sort_index(inplace=True) # 인덱스(날짜) 로 정렬
kosdaq_index['kosdaq_return'] = kosdaq_index['close']/kosdaq_index['close'].shift(1)
# 수익율 : 전 날 증가대비 당일 증가

merged = price_data.merge(kosdaq_index['kosdaq_return'], left_index=True,
right_index=True, how='left')
merged.to_pickle('merged.pkl')
```

[*****100%*****] 1 of 1 completed

주가 지수 수익률과 종목별 수익율을 비교한 결과를 win_market 이라는 변수에 담습니다.

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

return_all = pd.DataFrame()

for code in kosdaq_list['code']:

    stock_return = merged[merged['code']==code].sort_index()
    stock_return['return'] = stock_return['close']/stock_return['close'].shift(1) #
종목별 전일 증가 대비 당일 증가 수익율
    c1 = (stock_return['kosdaq_return'] < 1) # 수익율 1 보다 작음. 당일 증가가 전일 증가
보다 낮음 (코스닥 지표)
    c2 = (stock_return['return'] > 1) # 수익율 1 보다 큼. 당일 증가가 전일 증가보다 큼
(개별 종목)
    stock_return['win_market'] = np.where((c1&c2), 1, 0) # C1 과 C2 조건을 동시에 만족
하면 1, 아니면 0
    return_all = pd.concat([return_all, stock_return], axis=0)

return_all.dropna(inplace=True)
```

데이터가 잘 생성되었는 지 확인해 봅니다.

```
return_all.head().style.set_table_attributes('style="font-size:
12px"').format(precision=3)
```

	open	high	low	close	volume	change	code	name	kosdaq_return	return	win_market
date											
2021-12-23 00:00:00	3195	3260	3195	3220	104180	-0.002	060310	3S	1.003	0.998	0
2021-12-24 00:00:00	3230	3355	3220	3290	238933	0.022	060310	3S	1.004	1.022	0
2021-12-27 00:00:00	3290	3380	3275	3305	130826	0.005	060310	3S	1.004	1.005	0
2021-12-28 00:00:00	3355	3355	3180	3190	267316	-0.035	060310	3S	1.016	0.965	0
2021-12-29 00:00:00	3200	3350	3200	3330	115094	0.044	060310	3S	1.001	1.044	0

모델에 입력할 변수를 생성합니다.

```

model_inputs = pd.DataFrame()

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

    data = return_all[return_all['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 자주발생한다.
    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
양봉을 데이터로 표현
    data['num_long'] = data['long_candle'].rolling(60).sum() # 지난 60 일 동안 장대양
봉의 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'] # 거
래량은 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%
신뢰구간을 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 수익율
의 평균
    data['sector'] = sector
    data['name'] = name

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

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

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

model_inputs.to_pickle('model_inputs.pkl')

```

모델에 입력할 변수를 생성하고 X 에 담습니다.

```

# 최종 피쳐만으로 구성
model_inputs = pd.read_pickle('model_inputs.pkl')
feature_list =
['price_z', 'volume_z', 'num_high/close', 'num_win_market', 'pct_win_market', 'return over
sector']

X = model_inputs.loc[today_dt][['code', 'name', 'return'] +
feature_list].set_index('code') # 오늘 날짜 2022년 4월 1일 데이터만
X.head().style.set_table_attributes('style="font-size: 12px"]').format(precision=3)

```

	name	return	price_z	volume_z	num_high/close	num_win_market	pct_win_market	return over sector
code								
238490	힘스	0.997	-1.290	-0.510	0.000	6.000	1.000	1.002
037440	희림	0.981	0.144	-0.839	0.000	12.000	1.009	0.970
189980	흥국에프엔비	1.004	0.304	-0.555	0.000	14.000	1.002	1.000
010240	흥국	0.996	0.962	1.153	0.000	8.000	1.000	1.001
024060	흥구석유	1.011	-0.838	-0.591	0.000	15.000	1.006	1.012

저장한 GAM 모델을 불러 읽고, 입력변수를 넣어 예측값을 생성합니다. 입력변수의 순서는 모델에 사용한 입력변수와 동일해야 합니다. X 라는 데이터 프레임에 예측값 yhat 이 추가되었습니다.

```
import pickle
with open("gam.pkl", "rb") as file:
    gam = pickle.load(file)

yhat = gam.predict_proba(X[feature_list])
X['yhat'] = yhat
X.head().style.set_table_attributes('style="font-size: 12px"').format(precision=3)
```

	name	return	price_z	volume_z	num_high/close	num_win_market	pct_win_market	return over sector	yhat
code									
238490	힘스	0.997	-1.290	-0.510	0.000	6.000	1.000	1.002	0.205
037440	희림	0.981	0.144	-0.839	0.000	12.000	1.009	0.970	0.297
189980	흥국에프엔비	1.004	0.304	-0.555	0.000	14.000	1.002	1.000	0.218
010240	흥국	0.996	0.962	1.153	0.000	8.000	1.000	1.001	0.237
024060	흥구석유	1.011	-0.838	-0.591	0.000	15.000	1.006	1.012	0.290

어떤 종목이 높은 스코어를 받았는지 궁금합니다. 스코어의 내림차순 정렬한 후 종목을 확인해 봅니다.

```
X.sort_values(by='yhat',
ascending=False).head(5).style.set_table_attributes('style="font-size: 12px"').format(precision=3)
```

	name	return	price_z	volume_z	num_high/close	num_win_market	pct_win_market	return over sector	yhat
code									
056090	에디슨 INNO	1.080	-1.621	1.764	3.000	16.000	1.026	1.064	0.540
145020	휴젤	0.868	-3.379	7.109	0.000	6.000	0.997	0.868	0.497
185490	아이진	0.908	-2.249	2.269	0.000	6.000	0.991	0.920	0.490
069920	아이에스 이커머스	1.300	2.396	3.518	2.000	15.000	1.019	1.224	0.469
010280	쌍용정보통신	1.062	2.993	7.584	1.000	11.000	1.004	1.066	0.455

그리고 필터링을 적용해서 최종 종목을 선정합니다. 최종적으로 5 개의 종목이 선정되었습니다. 우리는 4월 1일 이후에 주가 흐름을 알고 있습니다. 4월 2일 이후 데이터를 추가하여 선택된 종목들이 유의미한지 점검해 보겠습니다.

```
tops = X[X['yhat'] >= 0.3].copy() # 스코어 0.3 이상 종목만
print(len(tops))
select_tops = tops[(tops['return'] > 1.03) & (tops['price_z'] < 0)]
[['name', 'return', 'price_z', 'yhat', 'return']]
select_tops.style.set_table_attributes('style="font-size: 12px"').format(precision=3)
```

	name	return	price_z	yhat	return
code					
024740	한일단조	1.062	-0.823	0.355	1.062
174880	장원테크	1.090	-0.366	0.305	1.090
056090	에디슨INNO	1.080	-1.621	0.540	1.080
122690	서진오토모티브	1.058	-0.114	0.314	1.058
083660	CSA 코스믹	1.035	-0.094	0.351	1.035

```

outcome_data = pd.DataFrame()

today_dt = '2022-04-01'
end_dt = '2022-04-08'

for code in list(select_tops.index): # 스코어가 생성된 모든 종목에서 대하여 반복
    daily_price = fdr.DataReader(code, start = today_dt, end = end_dt) # 종목, 일봉,
    # 데이터 갯수
    daily_price['code'] = code

    daily_price['close_r1'] = daily_price['Close'].shift(-1)/daily_price['Close'] #
    # 4월 1일 증가 매수한 후, 4월 4일 수익률
    daily_price['close_r2'] = daily_price['Close'].shift(-2)/daily_price['Close'] #
    # 4월 1일 증가 매수한 후, 4월 5일 수익률
    daily_price['close_r3'] = daily_price['Close'].shift(-3)/daily_price['Close'] #
    # 4월 1일 증가 매수한 후, 4월 6일 수익률
    daily_price['close_r4'] = daily_price['Close'].shift(-4)/daily_price['Close'] #
    # 4월 1일 증가 매수한 후, 4월 7일 수익률
    daily_price['close_r5'] = daily_price['Close'].shift(-5)/daily_price['Close'] #
    # 4월 1일 증가 매수한 후, 4월 8일 수익률

    daily_price['max_close'] =
    daily_price[['close_r1','close_r2','close_r3','close_r4','close_r5']].max(axis=1)
    daily_price['mean_close'] =
    daily_price[['close_r1','close_r2','close_r3','close_r4','close_r5']].mean(axis=1)
    daily_price['min_close'] =
    daily_price[['close_r1','close_r2','close_r3','close_r4','close_r5']].min(axis=1)

    daily_price['buy_price'] = daily_price['Close']
    daily_price['buy_low'] = daily_price['Low'].shift(-1)
    daily_price['buy_high'] = daily_price['High'].shift(-1)

    daily_price['buy'] =
    np.where((daily_price['buy_price'].between(daily_price['buy_low'],
    daily_price['buy_high'])), 1, 0) # 4월 2일 매수일, 4월 1일 증가에 살 수 있는 지 여부
    daily_price['target'] = np.where(daily_price['max_close']>=1.05, 1, 0)

    outcome_data = pd.concat([outcome_data, daily_price], axis=0)

```

최종 선정된 종목들의 결과가 궁금합니다. 선정된 종목 데이터에 결과 데이터를 병합합니다. 두 데이터셋의 인덱스는 종목이어야 병합이 가능합니다. 5% 익절할 확률은 83.3%로 높게 나왔습니다. 최저 수익률의 평균은 .98로 리스크도 비교적 낮은 것으로 보입니다. 2022년 4월 1일 매수한 종목은 수익권으로 예상이 됩니다. 물론 모든 날짜에 대하여 동일한 결과가 나오지는 않습니다.

```

outcome = outcome_data.loc[today_dt]
[['code','buy','buy_price','buy_low','buy_high','max_close','mean_close','min_close',
'target']].set_index('code')
select_outcome = tops.merge(outcome, left_index=True, right_index=True, how='inner')
select_outcome[['yhat','buy','max_close','mean_close','min_close']].mean()

```

```

yhat      0.375
buy        0.800
max_close  1.127
mean_close 1.055
min_close  0.983
dtype: float64

```

buy는 4월 1일 증가에 4월 2일 매수할 수 있는 기회가 있는지를 알려주는 Flag입니다. CSA 코스믹은 4월 2일 갭상승으로 시작했습니다. 4월 1일 증가에 살 수 있는 기회가 없습니다.

```
select_outcome[['name','buy','buy_price',
'buy_low','buy_high','yhat','max_close','mean_close','min_close']].style.set_table_at
tributes('style="font-size: 12px"').format(precision=3)
```

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
024740	한일단조	1	3185	3185.000	3300.000	0.357	1.057	1.025	0.983
174880	장원테크	1	1990	1985.000	2225.000	0.302	1.116	0.988	0.925
056090	에디슨INNO	1	12800	11350.000	13350.000	0.542	1.273	1.127	0.930
122690	서진오토모티브	1	3350	3330.000	3680.000	0.321	1.103	1.070	1.037
083660	CSA 코스믹	0	2080	2100.000	2270.000	0.353	1.087	1.067	1.041

2022년 4월 1일 추천받은 종목들의 일봉 차트를 보겠습니다. CSA 코스믹은 전일 증가로 당일 매수가 불가능합니다. 2022년 4월 2일 갭상승으로 시작을 했습니다. 에디슨 INNO 는 4월 2일 매수 후 익절할 기회를 제공하고 있습니다.

한일단조



장원테크

장원테크 시 1,960 고 1,960 저 1,860 중 1,870 ▼90 -4.59% 거 187,690

이동평균 5 20 60 120



에디슨INNO

에디슨INNO 시 11,550 고 12,450 저 10,600 중 11,850 ▲750 +6.76% 거 7,972,376

이동평균 5 20 60 120



서진오토모티브

일

주

월



서진오토모티브 시 3,510 고 3,550 저 3,420 종 3,460 ▼40 -1.14% 거 52,687

이동평균 5 20 60 120

Linear ▼



CSA 코스믹

일

주

월



CSA 코스믹 시 2,015 고 2,105 저 1,990 종 2,080 ▲70 +3.48% 거 142,908

이동평균 5 20 60 120

Linear ▼

