

# 해결책 테스트

오늘 날짜만 입력하면 내일 매수할 종목이 추천되도록 각 프로세스를 통합하여 함수를 구현합니다. 임의의 날짜를 넣어서 테스트 해 봅니다. 이 책에서는 2022년 4월 1일, 2022년 4월 18일, 2022년 5월 2일, 2022년 5월 9일, 2022년 5월 25일, 2022년 6월 2일, 6월 16일에 대하여 종목 선정 및 결과 수익률을 테스트 해 보았습니다. 모델을 개발하는데 사용한 날짜는 모델 검증 용도로 적절하지 않습니다. 왜냐하면 개발에서 사용한 데이터는 모델이 좋은 성과가 나오도록 최적화되어 있기 때문입니다. 참고로 모델 개발은 2021년 1월 4일부터 2022년 3월 24일까지 데이터가 사용되었습니다.

이제 종목을 추천하는 프로세스를 완성했습니다. 장 마감 후 종목 추천을 받아 익일 증권사 API를 이용해서 자동매매를 구현하고 한 달 동안의 수익이 어떤지 검증해 보겠습니다. 홈트레이딩 시스템에도 자동매매가 가능합니다. 책에서 구현할 자동매매는 홈트레이딩 감시 매매 설정으로도 충분히 가능합니다.

실전에서는 HTS 에서 제공하는 예약 매수기능과 매도 감시기능을 이용하는 것이 편리합니다. HTS 를 활용하여 자동으로 매수 매도가 가능합니다.

```
import FinanceDataReader as fdr
import yfinance as yf
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
import datetime
import pickle
import glob
```

추천 종목을 만드는 여러 개의 프로세스를 하나의 함수로 만들었습니다.

```

def select_stocks(today_dt):

    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)

    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 종
목에서 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

    # DataReader 코스닥 인덱스 조회 실패시, 야후파이낸스로 추출
    # kosdaq_index = fdr.DataReader('KQ11', start = start_dt, end = today_dt) # 데이
터 호출
    # kosdaq_index.columns = ['close', 'open', 'high', 'low', 'volume', 'change'] # 컬럼명
변경

    kosdaq_index = yf.download('^KQ11', start = start_dt)
    kosdaq_index.columns = ['open', 'high', 'low', 'close', 'adj_close', 'volume'] # 컬럼
명 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 증
가

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

    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
증가보다 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
만족하면 1, 아니면 0
        return_all = pd.concat([return_all, stock_return], axis=0)

    return_all.dropna(inplace=True)

    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 증가보다 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 20 일 동안 장
대양봉의 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% 신뢰구간을 벗어난 날을 카운트

        # 주가지수보다 더 좋은 수익율을 보여준다
        data['num_win_market'] = data['win_market'].rolling(60).sum() # 주가지수 수익
율이 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 수
익율의 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

        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', 'kosdaq_return', 'close'] +
feature_list].set_index('code')

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

        yhat = gam.predict_proba(X[feature_list])
        X['yhat'] = yhat

        tops = X[X['yhat'] >= 0.3].sort_values(by='yhat', ascending=False) # 스코어 0.3
이상 종목만
        print(len(tops))

        select_tops = tops[(tops['return'] > 1.03) & (tops['price_z'] < 0)]
[['name', 'return', 'price_z', 'yhat', 'kosdaq_return', 'close']] # 기본 필터링 조건

        if len(select_tops) > 1: # 최소한 2개 종목 - 추천 리스크 분산
            return select_tops

        else:
            return None

```

수익률 검정하는 프로세스도 하나의 함수로 구현합니다.

```
def outcome_tops(select_tops, today_dt, end_dt):

    outcome_data = pd.DataFrame()

    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']
        daily_price['close_r2'] = daily_price['Close'].shift(-2)/daily_price['Close']
        daily_price['close_r3'] = daily_price['Close'].shift(-3)/daily_price['Close']
        daily_price['close_r4'] = daily_price['Close'].shift(-4)/daily_price['Close']
        daily_price['close_r5'] = daily_price['Close'].shift(-5)/daily_price['Close']

        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) # 익일 저가
        daily_price['buy_high'] = daily_price['High'].shift(-1) # 익일 고가

        daily_price['buy'] =
        np.where((daily_price['buy_price'].between(daily_price['buy_low'],
        daily_price['buy_high']))), 1, 0) # 당일 증가로 익일 매수 가능한지 여부

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

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

    return select_outcome[['name','buy','buy_price',
    'buy_low','buy_high','yhat','max_close','mean_close','min_close']]
```

## 2022년 4월 1일 - 종목 선정 및 수익률 테스트

상당이 고무적입니다. 모든 종목이 익절이 가능합니다. 단 CSA 코스믹은 전일 증가로 당일 매수가 불가능합니다. 2022년 4월 2일 갭상승으로 시작을 했습니다.

```
select_tops = select_stocks('2022-04-01')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-04-01', '2022-04-08') # 5 영업일
    results.sort_values(by='buy').style.set_table_attributes('style="font-size:
    12px"').format(precision=3)
```

```
2021-12-22 00:00:00 2022-04-01
[*****100%*****] 1 of 1 completed
203
```

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
083660	CSA 코스믹	0	2080	2100.000	2270.000	0.351	1.087	1.067	1.041
056090	에디슨INNO	1	2560	2270.000	2670.000	0.540	1.273	1.127	0.930
024740	한일단조	1	3185	3185.000	3300.000	0.355	1.057	1.025	0.983
122690	서진오토모티브	1	3350	3330.000	3680.000	0.314	1.103	1.070	1.037
174880	장원테크	1	1990	1985.000	2225.000	0.305	1.116	0.988	0.925

## 2022년 4월 18일 - 종목 선정 및 수익률 테스트

4월 18일은 인성정보는 수익권, 웨이버스는 손절로 대응이 필요합니다.

```
select_tops = select_stocks('2022-04-18')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-04-18', '2022-04-25') # 5 영업일
results.sort_values(by='buy').style.set_table_attributes('style="font-size:
12px"]').format(precision=3)
```

2022-01-08 00:00:00 2022-04-18  
180

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
109820	진매트릭스	0	6690	6450.000	6670.000	0.311	0.978	0.943	0.903
089530	에이티세미콘	0	1940	1810.000	1915.000	0.310	1.134	1.037	0.951
033230	인성정보	1	2960	2930.000	3045.000	0.313	1.030	0.994	0.922
336060	웨이버스	1	2630	2535.000	2665.000	0.305	0.970	0.867	0.759

## 2022년 5월 2일 - 종목 선정 및 수익률 테스트

미래생명자원은 매수 후, 주가가 하락하는 것으로 나왔습니다. 다행이 급락 종목은 아니여서 손절로 대응하는 것이 좋을 것으로 판단됩니다.

```
select_tops = select_stocks('2022-05-02')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-05-02', '2022-05-10') # 5 영업일 (5월 5
일 어린이날)

results.sort_values(by='buy').style.set_table_attributes('style="font-size:
12px"]').format(precision=3)
```

2022-01-22 00:00:00 2022-05-02  
169

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
218150	미래생명자원	1	9690	9360.000	9870.000	0.430	0.991	0.918	0.863
014200	광림	1	2515	2445.000	2950.000	0.370	1.151	1.083	1.000
258610	케일럼	1	4575	4445.000	5100.000	0.327	1.045	0.999	0.954

## 2022년 5월 9일 - 종목 선정 및 수익률 테스트

5월 9일은 추천종목이 없습니다.

```
select_tops = select_stocks('2022-05-09')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-05-09', '2022-05-16') # 5 영업일 (5월 5
일 어린이날)

results.sort_values(by='buy').style.set_table_attributes('style="font-size:
12px"]').format(precision=3)
```

2022-01-29 00:00:00 2022-05-09  
348

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
218150	미래생명자원	1	9690	9360.000	9870.000	0.430	0.991	0.918	0.863
014200	광림	1	2515	2445.000	2950.000	0.370	1.151	1.083	1.000
258610	케일럼	1	4575	4445.000	5100.000	0.327	1.045	0.999	0.954

## 2022년 5월 25일 - 종목 선정 및 수익률 테스트

지더블유바이텍과 아이에스이커머스는 5영업일 이내 익절이 가능할 것으로 보입니다. 조이시티와 상지카일럼은 대응이 필요합니다.

```
select_tops = select_stocks('2022-05-25')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-05-25', '2022-06-02') # 5 영업일 (6월 1
    일 지방선거)

results.sort_values(by='buy').style.set_table_attributes('style="font-size:
12px"]').format(precision=3)
```

2022-02-14 00:00:00 2022-05-25  
144

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
069920	아이에스이커머스	1	6970	6960.000	7690.000	0.554	1.070	1.013	0.950
036180	지더블유바이텍	1	887	883.000	1045.000	0.413	1.125	1.074	1.025
005860	한일사료	1	8000	7670.000	8150.000	0.390	1.211	1.127	0.966
067000	조이시티	1	5990	5800.000	6180.000	0.335	0.990	0.977	0.967
227100	에이치앤비디자인	1	4430	4430.000	5640.000	0.335	1.411	1.341	1.221
104540	코렌텍	1	11550	11350.000	14800.000	0.327	1.139	1.099	1.065
042940	상지카일룸	1	1235	1170.000	1235.000	0.302	0.992	0.959	0.923

## 2022년 6월 2일 - 종목 선정 및 수익률 테스트

토탈소프트를 제외한 모든 종목이 익절이 가능할 것으로 보입니다.

```
select_tops = select_stocks('2022-06-02')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-06-02', '2022-06-10') # 5 영업일 (6월 6
    일 현충일)

results.sort_values(by='buy').style.set_table_attributes('style="font-size:
12px"]').format(precision=3)
```

2022-02-22 00:00:00 2022-06-02  
125

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
069920	아이에스이커머스	1	7410	7270.000	7720.000	0.419	1.076	1.006	0.961
021880	메이슨캐피탈	1	668	652.000	685.000	0.323	1.157	1.087	1.016
014200	광림	1	2100	2040.000	2185.000	0.308	1.221	1.035	0.938
045340	토탈소프트	1	6540	6450.000	6740.000	0.305	1.031	0.981	0.945

## 2022년 6월 16일 - 종목 선정 및 수익률 테스트

2022년 6월 16일 추천종목은 20 종목이 넘습니다. 종목은 모델 스코어가 높은 5 종목만 선택하도  
록 하겠습니다. 한타, 에이에프더블류, 베셀이 매수가 가능했습니다. 익절 가능할 것으로 예상됩  
니다.

```
select_tops = select_stocks('2022-06-16')

if select_tops is not None:
    results = outcome_tops(select_tops, '2022-06-16', '2022-06-23') # 5 영업일 (6월 6
    일 현충일)

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

2022-03-08 00:00:00 2022-06-16  
[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed  
395

	name	buy	buy_price	buy_low	buy_high	yhat	max_close	mean_close	min_close
code									
002680	한타	1	3155	3050.000	3595.000	0.457	1.132	1.029	0.805
177350	베셀	1	7630	7300.000	8280.000	0.413	1.054	1.006	0.920
312610	에이에프더블류	1	3505	3315.000	4555.000	0.397	1.185	1.054	0.874
317850	대모	1	10950	10250.000	11700.000	0.373	1.032	0.986	0.904
067010	이씨에스	1	3980	3800.000	4395.000	0.362	1.062	0.954	0.812