

# 통계 및 머신러닝 모델과 딥러닝 모델로 예측한 주가의 예측성능 비교

팀장 : 안희재(통계학과)  
팀원 : 김서영(통계학과), 이혜원(통계학과)

October 2020

## 1 Introduction

### 1.1 분석 목표

주식의 가격을 예측하고자하는 노력은 계속되고있다. 과거에는 정성적, 정량적인 지표를 반영한 전통적인 방법으로 주가를 예측해왔지만, 기술이 발달함에 따라 머신러닝과 딥러닝을 통한 다양한 방법론을 제시하고 있으며 예측의 정확도 또한 높아지고있다. 그 중 딥러닝은 뛰어난 예측 성능을 보이나 블랙박스 문제를 해결할 수 없다는 단점이 있다. 따라서 모델의 예측력이 높게 나왔더라도 이에 대한 원인과 결과를 명확히 규명하기가 어렵다. 이러한 블랙박스 특징 때문에 모델 성능의 인과관계와 변수의 영향력 등을 해석하기 어렵고, 모델 학습 진행 과정에 사람이 개입하여 조정할 수 없다. 반면 통계 모델과 머신러닝 모델의 경우 모델의 예측 결과에 대한 논리적인 설명이 가능하기 때문에, 실제 투자를 진행하는 투자자의 입장에서 통계, 머신러닝 모델이 더 설득력 있게 다가올 수 있다. 따라서 본 프로젝트에서는 통계 및 머신러닝 모델과 딥러닝 모델을 사용하여 주가 예측을 진행한 후, 오류지표를 기준으로 성능이 좋았던 몇가지 통계 및 머신러닝 모델들과 딥러닝 모델을 통계적인 방법과 포트폴리오로 비교해보고자 한다.

### 1.2 데이터 취득 방법

투자를 할때, 13개에서 17개 종목에 분산투자하는 것이 보통의 가치투자대가들이 권하는 정도이며 종목을 선정하는 기준은 투자자마다 다르지만, ”계란을 한 바구니에 담지말라.”라는 말이 있듯 종목의 시가총액, PBR, PER 그리고 테마등 다양한 분석을 통해 각기 다른 종목에 분산투자를 하는 것이 좋다고한다. 프로젝트에서는 15개 종목으로 포트폴리오를 작성하기로 결정하였고, 보통의 투자자들이 가장 많이 보는 종목의 시가총액과 어떤 테마에 속해있는지를 기준으로 종목들을 선정하였다. 각 종목별 일별시세 정보가 필요하다고 판단되었고, 데이터 셋을 얻기위해 HTML, XML 페이지로부터 데이터를 추출하는 파이썬 라이브러리인 ’뷰티풀수프’를 사용했다. [네이버 금융](#)에서 15개 종목의 일별 시세를 웹 크롤링했다.

### 1.3 프로젝트의 가치

본 프로젝트에서는 설명가능한 머신러닝 모델과 그렇지 않은 딥러닝 모델을 비교해보고자했다. 머신러닝 모델과 딥러닝 모델의 성능이 통계적으로 유의미하게 차이가 나지 않는 경우라면, 머신러닝 모델을 사용하는 것이 딥러닝 모델을 사용하는 것보다 분석적인 측면에서는 더 우수할 것이라고 생각했다. 선행 연구들에서 머신러닝 모델과 딥러닝 모델을 오류지표들의 대소비교를 통해 비교해보고 딥러닝 모델의 성능이 좋기때문에 딥러닝 모델을 사용하는 방식을 따라가고있지만, 본 프로젝트에서는 딥러닝 모델이 통계적인 방법으로 검증하였을때와 포트폴리오를 통해 검증해보았을때 모두 딥러닝 모델이 우수하다는 결론을 얻을 수 있었다.

## 2 Background

### 2.1 포트폴리오

#### 1. 동일비중 포트폴리오 ( Equally weighted portfolio )

- 정기적으로 투자 종목들의 비율을 늘 일정하게 맞춰주는 전략
- EX ) 1억원의 자금으로 100종목을 투자한다면, 모든 종목에 100만원씩 투자하는것.
- 2, 3개월에 한번씩 리밸런싱을 하며 시장에 계속 머물러있는 전략으로, 포트폴리오의 수익률은 구성 종목 수익률의 평균보다 대부분 높고, 최대 손실률 역시 구성 종목의 최대 손실률 평균값보다 대부분 낮다. 즉 동일비중 포트폴리오는 투자 위험은 줄여주고 수익률은 높여준다.
- 중소형주의 움직임을 잘 반영할 수 있다. 하지만 2008년 금융위기처럼 무차별적인 급락이 발생하는 시장에서는 중소형주가 대형주보다 크게 하락하는 경향이 있기 때문에 일반적인 주가지수보다 낙폭이 커지는 상황이 발생한다.

#### 2. 최소 분산 포트폴리오 ( Global Minimum Variance portfolio )

- 효율적 투자선( Efficient frontier ) : 해리 맥스 마코위츠가 제시한 평균-분산 최적화는 예상 수익률과 리스크의 상관관계를 활용해 포트폴리오를 최적화하는 기법이다. 투자자가 인내할 수 있는 리스크 수준에서 최상의 기대수익률을 제공하는 포트폴리오들의 집합으로 효율적 투자선위에 위치한 포트폴리오에 따라 자산을 배분한다.
- 그 중 가장 위험이 낮은 지점을 최소 분산 포트폴리오라고 한다.
- 현대 포트폴리오 이론에 따르면 개별 리스크가 주어졌을 때 효율적 투자선보다 높은 수익률은 기대할 수 없다.

#### 3. 최대 샤프지수 포트폴리오 (Max Sharpe Ratio Portfolio)

- 리스크를 최소화하고 수익률을 최대화하는 포트폴리오는 윌리엄 샤프가 창안한 샤프지수를 통해 찾을 수 있다. 샤프 지수는 측정된 위험단위당 수익률을 계산한다는 점에서 수익률의 표준편차와 다른 점이 있다. 샤프지수가 높을수록 위험에 대한 보상이 더 크다.

#### 4. 포트폴리오 리밸런싱 ( Re-balancing )

- 포트폴리오안에 있는 자산들의 비중을 조절하는 과정.

## 3 Data

### 3.1 데이터 스크래핑

야후 파이낸스의 주식 데이터, 네이버 금융의 주식 데이터등 한국 주식시장에 상장되어있는 주식들의 시세를 웹 크롤링해올 수 있는 곳들이 여러군데 있다. 처음에는 야후 파이낸스에서 주식 데이터를 크롤링해왔는데, 과거 데이터가 잘못되어있어서 실제 투자에 활용할 수 없다는 점을 깨닫고, 네이버 금융의 주식 데이터를 스크레이핑해서 데이터베이스  $\text{df}$ 를 구축하였다. 15개 종목에 대한 종목 번호을 우선 찾고, **네이버 금융**에서 일별 시세( 시가, 고가, 저가, 종가, 거래량 )을 크롤링해 각 종목에 대한 데이터 프레임을 생성했다.

다음은 선정된 15개 종목의 종목명과 종목번호이다.

- 삼성전자 (005930)
- SK하이닉스 (000660)
- NAVER (035420)

- 씨젠(096530)
- 우리들휴브레인 (118000)
- 셀트리온 (068270)
- 현대차 (005380)
- DGB금융지주 (139130)
- 데일리블록체인 (139050)
- 미스터블루 (207760)
- 소리바다 (053110)
- 한화솔루션 (009830)
- 아모레퍼시픽 (090430)
- CJ대한통운 (000120)
- GS건설 (006360)

구체적인 스크래핑 과정은 'SK하이닉스'로 설명하겠다. ( 15개 종목 모두 스크래핑 과정이 동일하기 때문 ) 네이버 금융에서 SK하이닉스를 검색해보았을때, 가장 밑에 일별 시세칸이 있다. 해당 날짜의 종가, 전일비, 시가, 고가, 저가 그리고 거래량이 나와있는 데이터이다. 한 페이지당 10일의 일별 시세가 나와있는 것을 볼 수 있다.

SK하이닉스의 경우 600페이지까지 일별시세 데이터가 존재했다. (이때, 종목마다 마지막 페이지의 숫자가 다른 것을 반영하기 위해, 마지막 페이지를 알려주는 숫자를 스크래핑하여 데이터마다 다르게 적용시켜주었다.) 페이지를 하나씩 증가시키면서 html문서 스크래핑을 진행하였다. ( 이때, 종목별로 상장날짜가 다른 것과 2000년대 이전의 종목 데이터는 불필요하다고 간주해 2000년 1월을 시작으로 일별 시세를 스크래핑했다. )

스크래핑한 데이터는 판다스를 이용해, csv파일로 저장해 프로젝트의 편의성을 높일 수 있었다.

아래 Table 1은 SK하이닉스 일별시세를 스크래핑 한 결과로, 5번째 행까지만 나타낸 표이다.

| 날짜         | 종가    | 전일비  | 시가    | 고가    | 저가    | 거래량      |
|------------|-------|------|-------|-------|-------|----------|
| 2000.01.04 | 23750 | 1650 | 24700 | 26450 | 24200 | 9275920  |
| 2000.01.05 | 23100 | 2650 | 24150 | 24950 | 22750 | 7414370  |
| 2000.01.06 | 21900 | 1200 | 23500 | 23800 | 21650 | 6529140  |
| 2000.01.07 | 21700 | 200  | 21200 | 22400 | 20800 | 10492270 |
| 2000.01.10 | 21500 | 200  | 22000 | 22300 | 21500 | 8041690  |

Table 1: SK하이닉스 데이터셋 일부분

### 3.2 Exploratory Data Analysis

상장시기가 2000년 이후인 종목에 대해서 결측값은 NaN값으로 채워주었다. 또한, 2020년의 주가는 코로나19라는 개입변수의 영향이 클 것이라고 예상하였다. 이는 과거의 주가들의 움직임을 반영해 주가를 예측하고자 하는 본 프로젝트의 취지와는 어긋나는 것 같아 개입효과를 분석해서 예측하기보단 제거하기로 결정하였다.

가격에 대한 변수로는 종가, 시가, 고가, 저가 총 4개가 존재한다. 4가지 변수에 대해 상관 계수를 구해보았을때 상관정도가 모든 종목에서 유의하게 높은 양의 상관정도를 보이는 것을 알 수 있었다. 따라서, 종목 종가를 일별 가격의 대표값으로 생각하고 추후 분석을 진행하기로 했다.

각 종목에 대한 시계열 그래프를 그려봄으로써 거래정지, 액면분할, 주가 등락과 폭락, 주가에 대한 간단한 추세와 계절성을 파악할 수 있었다. ( 시계열 그래프와 종목별 간단한 특징에 대해서는 부록 Figure 3 참고. )

## 4 Method

- 15개 종목의 2019년 4분기의 주가 예측하기 위해 주어진 데이터를 Train / Validation / Test 세트으로 분할

TRAIN SET - VALIDATION SET : 2013년 01월 02일 - 2019년 08월 30일 ( 8 : 2 비율 )

TEST SET : 2019년 09월 01일 - 2019년 12월 31일

( Figure 1 참고 )

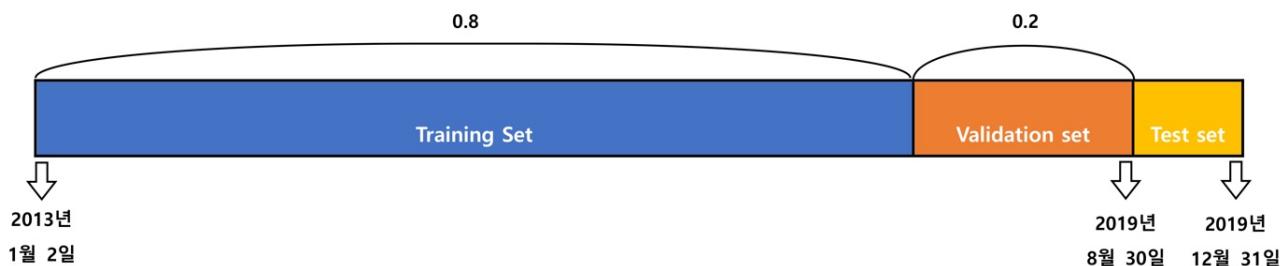


Figure 1: Train / Validation / Test

- 사용 데이터셋이 시계열 데이터이기 때문에 random shuffle을 진행하지 않고 교차검증을 진행  
이때, 각 모델에 대한 성능의 지표로 RMSE, MAPE를 사용

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Predicted_i - Actual_i)^2}{n}}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Actual_i - Predicted_i}{Actual_i} \right|$$

- 위의 성능 지표로 통계 및 머신러닝 모델에서 최종 후보군 모델 3가지 선정
- 위의 성능 지표로 딥러닝 모델에서 하이퍼파라미터 튜닝을 통해 1가지 모델 선정
- 총 4가지 모델로 15개 종목의 2019년 4분기 종목 종가 예측
- 통계 및 머신러닝 모델과 딥러닝 모델의 예측 성능 비교 이때, 통계적인 유의미성을 판단하기 위해 DM test를 사용
- 모델별로 예측된 15개 종목 종가로 포트폴리오 생성  
실제 2019년 4분기 종목 종가로 만든 포트폴리오와 예측된 종가로 만든 포트폴리오 비교  
이때, 포트폴리오 비교를 위해 수익률, 변동성 그리고 샤프지수를 사용

## 5 Experiments

본 프로젝트에서 사용한 총 12개의 모델에 대한 간단한 설명과 적용 방법

- Moving Average

단순 이동평균은 일정 기간 동안의 가격을 모두 더한 뒤 이를 가격 개수로 나누어 평균값을 구한 것이다. 이렇게 구해진 이동평균 값들을 선으로 이으면, 이동평균선이 되는데 이동평균선의 진행 방향을 보면 전반적인 가격 흐름을 예측할 수 있다. 이 프로젝트에서는 예측하고자 하는 기간의 가격 개수로 평균을 구하였다.

- Linear Regression

종속 변수  $y$ 와 한 개 이상의 독립 변수  $X$ 와의 선형 상관 관계를 모델링하는 기법이다. 이는 독립 변수와 종속 변수 간의 관계를 결정하는 방정식을 반환한다. 프로젝트에서는 train 시작날부터 날짜가 지남에 따라 1씩 증가한 값을 time이라는 변수로 만들어 선형 회귀 모델을 만들었다.

- Polynomial Regression

Linear Regression에서 만든 time 변수와 함께 time을 제곱시킨 timebytime이라는 2가지 변수를 사용해 선형 회귀 모델을 만들었다.

- Linear Regression With independent Variables

time 변수 대신 날짜로 새로운 변수들을 만들었다. 요일을 숫자로 나타내는 **weekday** 변수( 월:0,화:1, 수:2,목:3,금:4 ), 한해의 몇 번째 날이지를 나타내는 **day of year** 변수, 매달의 1일을 나타내는 **is month start** 더미 변수, 매달의 마지막날인지 나타내는 **is month end** 더미 변수, 분기의 시작을 나타내는 **is quarter start** 더미 변수, 분기의 마지막을 나타내는 **is quarter end** 더미 변수, 월요일 혹은 금요일인지를 나타내는 **mon fri** 더미 변수, 총 7가지 변수를 새로 만들어서 선형 회귀 모델을 만들었다.

- K-Nearest Neighbor (KNN)

독립 변수를 기반으로 새로운 데이터 포인트와 이전 데이터 포인트간의 유사성을 이용하여 새로운 데이터 포인트의 값을 예측한다. 이때 주변의 몇개의 데이터를 보고 결정할지 최적의 이웃 수를 선정하기 위해, 파라미터를 3, 5, 10, 15, 20, 25, 30로 설정하였고 교차검증과정을 통해 최적의 이웃수를 선정하였다.

- Hidden Markov Model (HMM)

통계적 마르코프 모형의 하나로, 시스템이 은닉된 상태와 관찰 가능한 결과의 두 가지 요소로 구성되었다고 보고 마르코프 성질을 이용하여 예측하는 확률적 통계 모델이다. 관찰 가능한 결과를 야기하는 직접적인 원인은 관측될 수 없는 은닉 상태와 오직 그 상태들이 마르코프 과정을 통해 도출된 결과들만 관찰한다. 과거와 현재 상태가 주어졌을 때 미래 상태의 조건부 확률 분포가 과거 상태와는 독립적으로 현재 상태에 의해서만 결정된다.

- Prophet Model

prophet 알고리즘은 푸리에급수(Fourier series)를 이용하여 seasonality 패턴을 추정한다.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_i$$

- $g(t)$  : 주기적이지 않은 변화인 트렌드.
- $s(t)$  : weekly, yearly 등 주기적으로 나타내는 패턴.
- $h(t)$  : 휴일과 같이 불규칙한 이벤트. 만약 특정 기간에 값이 비정상적으로 증가 또는 감소했다면, 휴일로 정의하여 모델에 반영할 수 있음.
- $\epsilon_i$  : 정규분포를 가정한 오차

- ARIMA(p,d,q)

$$w_t = c + \phi_1 w_{t-1} + \dots + \phi_p w_{t-p} + \theta_1 \epsilon_t - 1 + \dots + \theta_q \epsilon_t - q + \epsilon_t$$

( where,  $w_t = y_t(1 - B)$  )

- $p$  : 자기회귀 부분의 차수
- $d$  : 1차 차분이 포함된 정도
- $q$  : 이동 평균 부분의 차수

- RandomForest

양상을 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리로부터 회귀분석을 출력함으로써 작동한다. 랜덤성에 의해 트리들이 서로 조금씩 다른 특성을 갖는다. 이 특성은 각 트리들의 예측들이 비상관화 되게 하며, 결과적으로 일반화 성능을 향상시킨다. grid를 사용하여 각 종목마다의 최적의 하이퍼파라미터를 조정하였다.

- Support Vector Machine (Linear model)

두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만든다. 만들어진 분류 모델은 데이터가 사상된 공간에서 경계로 표현되는데 SVM 알고리즘은 그中最 가장 큰 폭을 가진 경계를 찾는 알고리즘이다.

- Support Vector Regression (RBF kernel)

kernel support vector는 데이터에서 단순한 초평면으로 정의되지 않은 더 복잡한 모델을 만들 수 있도록 확장한 것이다. 비선형의 단점을 보완하고, 수학적 기교를 사용하여 새로운 특성을 많이 만들지 않아도 고차원에서 분류기를 학습시킬 수 있다. Radial Basis Function Kernel을 사용하였다.

- Long short-term memory(LSTM)

순환 신경망 모델은 데이터가 갖고 있는 시간과 순서에 따른 의미를 학습에 반영시킬 수 있다. CNN과 같이 일반적인 신경망 모델은 데이터가 입력층에서 은닉층까지 연산이 순차적으로 진행되면서 노드를 한 번씩만 지나기 때문에 데이터의 시간적인 측면을 고려하지 못한다. 따라서 신경망 모델에 이전 은닉층의 값이 다시 해당 은닉층의 입력으로 들어가는 순환성을 추가한 것이 순환 신경망 모델이다. 순환 신경망 모델은 주식 데이터와 같이 데이터에 시간적 순서에 따른 의미가 있는 데이터를 분석할 때 사용할 수 있다. LSTM모델은 순환 신경망 모델에 장기기억을 담당하는 부분을 추가하여 데이터의 시간 정보가 길어지면 제대로 학습하지 못하는 단점을 보완한 모델이다.

## 6 Experiments

### 6.1 모델 선정 과정

Table 2는 통계 및 머신러닝 모델에서 최종 3가지 모델을 선정하기 위하여 SK하이닉스의 Training set을 통해 Validation기간의 주가를 예측하였을때 오류지표를 구한 표이다.

Table 3는 딥러닝에서 최적의 성능을 보이는 LSTM 모델을 찾기 위해 window size와 LSTM의 레이어 구성을 조절해가며 Training set을 통해 Validation기간의 주가를 예측하였을때 오류지표를 구한 표이다.

### 6.2 모델 선정 결과

통계 및 머신러닝 모델에서는 오류지표로 사용한, RMSE와 MAPE 모두 값이 작을 수록 성능이 좋은 모델이라고 판단했다. 오류지표가 낮은 2개의 모델 Moving Average, Support Vector Regression(RBF

| Model  | RMSE         | MAPE     |
|--|--------------|----------|
| Moving Average   | 10046.741364 | 0.103832 |
| Linear Regression                                      | 11253.943648 | 0.106122 |
| Polynomial Regression                                  | 25809.998395 | 0.317298 |
| Linear Regression With Different Independent Variables | 11292.062572 | 0.105975 |
| KNN  | 32800.365563 | 0.316690 |
| Hidden Markov Model                                    | 20067.372011 | 0.190747 |
| Prophet  | 50839.675248 | 0.651720 |
| ARIMA  | 14148.235478 | 0.175143 |
| RandomForest   | 8067.691542  | 0.089422 |
| Support Vector Machine (Linear model)                  | 11475.401156 | 0.107988 |
| Support Vector Regression (RBF Kernel)                 | 8122.066619  | 0.092135 |

Table 2: 11가지 모델들의 오류지표 비교

| Model                               | RMSE               | MAPE               |
|-------------------------------------|--------------------|--------------------|
| Model 1                             |                    |                    |
| Window size=20                      | 1418.4905065553346 | 1.7742903164493526 |
| LSTM(16) - Dense(1)                 |                    |                    |
| Model 2                             |                    |                    |
| Window size=60                      | 1476.528097825081  | 1.8413499929581114 |
| LSTM(16) - Dense(1)                 |                    |                    |
| Model 3                             |                    |                    |
| Window size=20                      | 1823.4186124528364 | 2.125875402037818  |
| LSTM(16) - LSTM(16) - Dense(1)      |                    |                    |
| Model 4                             |                    |                    |
| Window size=20                      | 1665.021040510825  | 1.9657217631418076 |
| LSTM(50) - LSTM(50) - Dense(1)      |                    |                    |
| Model 5                             |                    |                    |
| Window size=20                      | 1697.8539409864538 | 2.02968328793678   |
| LSTM(50) - Bidirectional - Dense(1) |                    |                    |

Table 3: 오류지표를 이용한 LSTM 파라미터 설정

**Kernel)**과 을 15개 종목의 주가를 예측하는 데 사용할 최종 모델로 선정하였다.**RandomForest**는 오류지표는 낮지만 예측한 기간동안 모두 동일한 상수값으로 예측되는 것을 확인 할 수 있었다. 주가의 흐름을 파악하기에는 무리라고 판단하여 최종 모델로는 선정하지 않기로 결정하였다. 오류지표의 크기는 다른 모델들에 비해 크지만 주가의 흐름을 잘 표현한 것처럼 보여 다른 종목에서는 좋은 예측성능을 보이지 않을까 하는 기대감으로 **prophet**도 최종 모델로 선정하였다. 선택한 모델들의 validation값과 예측값을 시계열 그래프를 그려보았다(Figure 4.).

딥러닝 모델에서는 window size를 20으로 설정하고 유닛 개수가 20인 레이어와 Dense 레이어로 구성된 LSTM 모델을 사용했을 때의 RMSE값과 MAPE값이 가장 작은 것을 확인할 수 있어 최종 LSTM모델을 선정하였다.

위에서 선택한 4가지 모델을 사용하여 15개 종목의 2019년 4분기 주가를 예측을 진행하였다. Figure 5는 최종 모델로 15개 종목의 2019년 4분기 주가를 예측한 결과를 시계열그래프로 그린 것이다. 빨간색은 실제 값, 파란색은 Moving Average모델, 보라색은 Prophet모델, 회색은 Support Vector Regression (RBF kernel) 모델, 노란색은 LSTM모델로 예측한 결과이다.

## 7 Experimental Results

### 7.1 검정통계량을 이용한 예측 성능 비교

통계 및 머신러닝 모델에서 최종으로 선정된 3가지 선정된 모델과 딥러닝 모델에서 선정된 LSTM모델을 비교하기 위해, 2019년 4분기 15개 종목의 각 종가를 예측해 예측 성능을 비교하였다.

이때, 예측 성능을 비교하기 위해 사용한 지표는 MAPE이며 검정 통계량으로는 Diebold-Mariano 통계량을 사용하였다.

*Diebold – Marianoteststatistic*

#### 1. Moving Average VS LSTM

$H_0$  : LSTM 모델의 예측 성능과 Moving Average모델의 예측 성능은 동일하다.

$H_1$  : LSTM 모델의 예측 성능이 Moving Average모델의 예측 성능보다 우수하다. (Table 4 참고.)

| 종목      | 통계량값      | P-value      | 검정결과    |
|---------|-----------|--------------|---------|
| 삼성전자    | -3.111140 | 2.583372e-03 | 귀무가설 기각 |
| SK하이닉스  | 19.516014 | 3.712042e-32 | 귀무가설 기각 |
| NAVER   | 41.564330 | 6.283349e-56 | 귀무가설 기각 |
| 씨젠      | 18.469187 | 1.384513e-30 | 귀무가설 기각 |
| 우리들휴브레인 | 12.656597 | 8.626462e-21 | 귀무가설 기각 |
| 현대차     | 15.020230 | 5.371991e-25 | 귀무가설 기각 |
| DGB금융지주 | 7.536632  | 6.498088e-11 | 귀무가설 기각 |
| 미스터블루   | 9.235813  | 3.023595e-14 | 귀무가설 기각 |
| 셀트리온    | 7.615419  | 4.564167e-11 | 귀무가설 기각 |
| 데일리블록체인 | 9.652084  | 4.619869e-15 | 귀무가설 기각 |
| 소리바다    | 8.893185  | 1.424351e-13 | 귀무가설 기각 |
| 한화솔루션   | 13.086934 | 1.405892e-21 | 귀무가설 기각 |
| 아모레퍼시픽  | 13.368917 | 4.333999e-22 | 귀무가설 기각 |
| CJ대한통운  | 10.991570 | 1.168750e-17 | 귀무가설 기각 |
| GS건설    | 20.063072 | 5.889716e-33 | 귀무가설 기각 |

Table 4: Moving Average VS LSTM

15개 종목에 대해 가설 검정을 진행해보았을때, 유의수준 0.05하에서 모두 LSTM모델의 예측 성능이 우수하다는 것을 알 수 있다.

#### 2. Prophet VS LSTM

$H_0$  : LSTM 모델의 예측 성능과 Prophet모델의 예측 성능은 동일하다.

$H_1$  : LSTM 모델의 예측 성능이 Prophet모델의 예측 성능보다 우수하다. (Table 5 참고.)

15개 종목에 대해 가설 검정을 진행해보았을때, 유의수준 0.05하에서 모두 LSTM모델의 예측 성능이 우수하다는 것을 알 수 있다.

#### 3. Support Vector Regression (RBF kernel) VS LSTM

$H_0$  : LSTM 모델의 예측 성능과 Support Vector Regression (RBF kernel)모델의 예측 성능은 동일하다.

$H_1$  : LSTM 모델의 예측 성능이 Support Vector Regression (RBF kernel)모델의 예측 성능보다 우수하다.

(Table 6 참고.)

15개 종목에 대해 가설 검정을 진행해보았을때, 유의수준 0.05하에서 모두 LSTM모델의 예측 성능이 우수하다는 것을 알 수 있다.

| 종목      | 통계량값       | P-value       | 검정결과    |
|---------|------------|---------------|---------|
| 삼성전자    | 22.518898  | 2.265212e-36  | 귀무가설 기각 |
| SK하이닉스  | 163.604757 | 8.274696e-103 | 귀무가설 기각 |
| NAVER   | 275.841604 | 6.336119e-121 | 귀무가설 기각 |
| 씨젠      | 20.515778  | 1.316680e-33  | 귀무가설 기각 |
| 우리들휴브레인 | 70.602379  | 7.832037e-74  | 귀무가설 기각 |
| 현대차     | 507.345821 | 4.395192e-142 | 귀무가설 기각 |
| DGB금융지주 | 242.360634 | 1.960269e-116 | 귀무가설 기각 |
| 미스터블루   | 89.884148  | 4.063557e-82  | 귀무가설 기각 |
| 셀트리온    | 351.784078 | 2.287987e-129 | 귀무가설 기각 |
| 데일리블록체인 | 37.261641  | 2.572581e-52  | 귀무가설 기각 |
| 소리바다    | 90.162119  | 3.181721e-82  | 귀무가설 기각 |
| 한화솔루션   | 92.832614  | 3.147966e-83  | 귀무가설 기각 |
| 아모레퍼시픽  | 149.826018 | 9.217479e-100 | 귀무가설 기각 |
| CJ대한통운  | 298.143287 | 1.268229e-123 | 귀무가설 기각 |
| GS건설    | 20.063072  | 5.889716e-33  | 귀무가설 기각 |

Table 5: Prophet VS LSTM

## 7.2 포트폴리오를 이용한 예측 성능 비교

2019년 4분기 실제 종가와 모델들로 예측된 종가를 통해 3가지 포트폴리오를 구성하였고, 모델간의 비교를 위한 지표로는 포트폴리오의 연간 수익률, 리스크 그리고 샤프지수가 있다. 포트폴리오를 이용한 예측 성능을 비교할 때에는 통계 및 머신러닝 모델과 딥러닝 모델의 비교뿐만 아니라, 실제 종가를 통해 만들어진 포트폴리오와도 비교하였다.

$$\text{Portfolio Return} = (\text{stock weight})X(\text{stock annual return})$$

$$\text{Portfolio Risk} = \sqrt{(\text{stock weight})^T ((\text{stock annual covariance})(\text{stock weight}))}$$

$$\text{Sharpe Ratio} = \frac{(\text{portfolio Return} - \text{risk free ratio})}{\text{Portfolio Risk}}$$

### 1. 동일비중 포트폴리오

모든 종목에 대하여 동일한 비중으로 투자를 하는 방식이다. 동일비중 포트폴리오를 구했을 때의 모델별 결과를 Table 7에 나타내었다. LSTM이 실제 주가의 값과 제일 비슷한 결과를 예상하는 것을 볼 수 있다.

### 2. 최소 분산 투자 포트폴리오

실제 주가와 4가지 모델을 사용하여 예측한 주가로 최소 분산 투자 포트폴리오를 만들어 보았다. 종목 비중을 다르게 한 포트폴리오 10000개를 생성하여 예상 수익률과 리스크를 비교하였다. Figure 6은 모델별 10000개의 포트폴리오를 시각화한 그림이다. x축은 해당 포트폴리오의 리스크를, y축은 예상 수익률, 점 하나가 각각의 포트폴리오를 나타낸다. 만들어진 그래프의 왼쪽 바깥라인을 효율적투자선이라고 부르는데, 현대 포트폴리오 이론에 따르면 개별 리스크가 주어졌을 때 효율적 투자선보다 높은 수익률은 기대할 수 없다. 효율적 투자선 위에 있는 점들 중 리스크가 제일 작은 포트폴리오를 최소 분산 투자 포트폴리오라고 부른다. Figure 6, Figure 7의 파란색 점이 최소 분산 투자 포트폴리오이다. 최소 분산 투자 포트폴리오를 구했을 때의 모델별 결과를 Table 8에 나타내었다. LSTM을 기준으로 표를 설명하면, Table 9의 비율로 투자를 하게 되면 4개월동안 11%의 변동을 가지며 81%의 수익률을 예상할 수 있다는 뜻이다. Table 8를 살펴보면 LSTM이 실제 주가의 값과 제일 비슷한 결과를 예상하는 것을 볼 수 있다.

| 종목      | 통계량값       | P-value       | 검정결과    |
|---------|------------|---------------|---------|
| 삼성전자    | 22.518898  | 2.265212e-36  | 귀무가설 기각 |
| SK하이닉스  | 163.604757 | 8.274696e-103 | 귀무가설 기각 |
| NAVER   | 275.841604 | 6.336119e-121 | 귀무가설 기각 |
| 씨젠      | 20.515778  | 1.316680e-33  | 귀무가설 기각 |
| 우리들휴브레인 | 70.602379  | 7.832037e-74  | 귀무가설 기각 |
| 현대차     | 507.345821 | 4.395192e-142 | 귀무가설 기각 |
| DGB금융지주 | 242.360634 | 1.960269e-116 | 귀무가설 기각 |
| 미스터블루   | 89.884148  | 4.063557e-82  | 귀무가설 기각 |
| 셀트리온    | 351.784078 | 2.287987e-129 | 귀무가설 기각 |
| 데일리블록체인 | 37.261641  | 2.572581e-52  | 귀무가설 기각 |
| 소리바다    | 90.162119  | 3.181721e-82  | 귀무가설 기각 |
| 한화솔루션   | 92.832614  | 3.147966e-83  | 귀무가설 기각 |
| 아모레퍼시픽  | 149.826018 | 9.217479e-100 | 귀무가설 기각 |
| CJ대한통운  | 298.143287 | 1.268229e-123 | 귀무가설 기각 |
| GS건설    | 20.063072  | 5.889716e-33  | 귀무가설 기각 |

Table 6: Support Vector Regression (RBF kernel) VS LSTM

| Model          | Return    | Volatility |
|----------------|-----------|------------|
| Actual         | 0.749208  | 0.162030   |
| Moving Average | -0.051136 | 0.009137   |
| RBF kernel     | 0.169570  | 0.006381   |
| Prophet        | -0.051136 | 0.009137   |
| LSTM           | 0.785706  | 0.126812   |

Table 7: Equally Weighted Portfolio

### 3. 최대 샤프지수 비율 포트폴리오

최소 분산 투자 포트폴리오와 똑같은 방식으로 시각화를 하고, 효율적 투자선을 그린다. 단, 리스크가 가장 작은 포트폴리오를 선택하는 것이 아니라 샤프지수가 제일 큰, 즉 리스크당 수익률이 가장 큰 포트폴리오를 최종 포트폴리오로 선택하는 방법이다. Figure 6의 빨간색 점이 샤프지수 포트폴리오에 해당한다. 샤프지수 포트폴리오를 구했을 때 모델별 결과를 Table ??에 나타내었다. LSTM을 기준으로 표를 설명하면, Table ??의 비율로 투자를 하게 되면 4개월 동안 9%의 변동을 가지며 47%의 수익률을 예상할 수 있다는 뜻이다. Table 8를 살펴보면 LSTM이 실제 주가와 제일 비슷한 결과를 예상하는 것을 볼 수 있다.

### 7.3 모델의 한계점 파악

- Moving Average

Moving Average를 사용한 예측값은 훈련 데이터의 평균을 이용한 값이기 때문에 훈련 데이터셋에서 관측된 값과 동일한 범위를 가지며 변동이 일어난다. 즉, 학습데이터셋에서 나오지 않은 값은 예측값으로 나올 수 없다. 따라서 값이 기존의 흐름보다 더 크거나 작은 값은 예측할 수 없어 이 경우에는 성능이 낮게 나오게 된다.

- Support Vector Regression(RBF kernel)

Support Vector Regression모델은 데이터의 특성이 몇 개 되지 않더라도 복잡한 결정경계를 만들 수 있지만, 샘플이 많을수록 속도가 느려지고 메모리 할당이 크며 성능이 잘 나오지 않는다는 단점을 가지고 있다. 이는 이 프로젝트에서 사용한 데이터와 맞지 않는 데이터임을 파악할 수 있다.

| Model          | Return    | Volatility |
|----------------|-----------|------------|
| Actual         | 0.303123  | 0.126622   |
| Moving Average | -0.006623 | 0.003919   |
| RBF kernel     | 0.000303  | 0.000409   |
| Prophet        | 0.007778  | 0.004472   |
| LSTM           | 0.470363  | 0.095709   |

Table 8: Minimum Variance Portfolio

| Stock   | Moving Average | RBF kernel | Prophet  | LSTM     | Actual   |
|---------|----------------|------------|----------|----------|----------|
| 삼성전자    | 0.136068       | 0.02776    | 0.041304 | 0.145499 | 0.029813 |
| SK하이닉스  | 0.062401       | 0.131248   | 0.120097 | 0.103967 | 0.124721 |
| NAVER   | 0.131212       | 0.040158   | 0.141953 | 0.09912  | 0.067565 |
| 씨젠      | 0.012184       | 0.133566   | 0.011332 | 0.14154  | 0.129012 |
| 유리들휴브레인 | 0.027003       | 0.033795   | 0.055966 | 0.009211 | 0.079144 |
| 현대차     | 0.016266       | 0.0066     | 0.044695 | 0.006179 | 0.008678 |
| DGB금융지주 | 0.047212       | 0.148456   | 0.115591 | 0.051737 | 0.10415  |
| 미스터블루   | 0.177814       | 0.13307    | 0.161099 | 0.022883 | 0.008699 |
| 셀트리온    | 0.179693       | 0.020036   | 0.087151 | 0.053874 | 0.121195 |
| 데일리블록체인 | 0.01835        | 0.010353   | 0.001102 | 0.037124 | 0.035367 |
| 소리바다    | 0.01835        | 0.115769   | 0.00002  | 0.013119 | 0.001613 |
| 환화솔루션   | 0.058449       | 0.02121    | 0.048013 | 0.04745  | 0.049916 |
| 아모레퍼시픽  | 0.040464       | 0.0596     | 0.046632 | 0.144292 | 0.128542 |
| CJ대한통운  | 0.015131       | 0.110649   | 0.111303 | 0.089327 | 0.111997 |
| GS건설    | 0.059461       | 0.007731   | 0.011741 | 0.034677 | 0.00859  |
| 총합      | 1              | 1          | 1        | 1        | 1        |

Table 9: Minimum Variance Portfolio Stock Weights

- Prophet

Prophet는 과거의 데이터로부터 추세와 계절성을 파악하여 적용하는 모델이다. 일반적인 시계열 데이터셋에서는 좋은 성능을 발휘한다. 하지만 주가 데이터는 현재 시장에서 어떠한 상황이 일어나고 있는지에 영향을 받지, 특별한 추세나 계절성을 가지지 않는 시계열 데이터이다. 따라서 주가예측에서 좋은 모델이라고 할 수 없다.

## 8 Appendix

- diebold-mariano test ( DM test )

Figure 2 참고.

- Figure 3

15개 종목의 2013년 1월 2일부터 2019년 12월 31일까지의 데이터의 시계열 그래프

– 거래 정리 기간

- \* 삼성전자 : 2018.04.30 - 2018.05.03
- \* 우리들휴브레인 : 2013.10.16, 2014.02.13 - 2014.03.10
- \* 미스터블루 : 2015.05.13 - 2015.07.09
- \* 셀트리온 : 2013.03.04- 2013.03.21

| Model          | Return   | Volatility |
|----------------|----------|------------|
| Actual         | 0.773325 | 0.153477   |
| Moving Average | 0.164476 | 0.018482   |
| RBF kernel     | 0.386924 | 0.014756   |
| Prophet        | 0.060980 | 0.060980   |
| LSTM           | 0.810191 | 0.113328   |

Table 10: Max Sharpe Ratio Portfolio

| Stock   | Moving Average | RBF kernel | Prophet  | LSTM     | Actual   |
|---------|----------------|------------|----------|----------|----------|
| 삼성전자    | 0.020918       | 0.011636   | 0.051206 | 0.074742 | 0.081961 |
| SK하이닉스  | 0.043154       | 0.084618   | 0.121176 | 0.014314 | 0.055721 |
| NAVER   | 0.161903       | 0.014027   | 0.145937 | 0.125455 | 0.037841 |
| 씨젠      | 0.166225       | 0.123254   | 0.090905 | 0.024286 | 0.066617 |
| 유리들휴브레인 | 0.013384       | 0.062141   | 0.05337  | 0.01351  | 0.013106 |
| 현대차     | 0.130868       | 0.137338   | 0.028158 | 0.049587 | 0.161665 |
| DGB금융지주 | 0.014165       | 0.074705   | 0.127665 | 0.177226 | 0.086497 |
| 미스터블루   | 0.006485       | 0.046199   | 0.091764 | 0.005608 | 0.054438 |
| 셀트리온    | 0.160302       | 0.072968   | 0.009721 | 0.038347 | 0.065437 |
| 데일리블록체인 | 0.000711       | 0.017299   | 0.016183 | 0.00239  | 0.007897 |
| 소리바다    | 0.014044       | 0.07771    | 0.026174 | 0.037229 | 0.072517 |
| 환화솔루션   | 0.004726       | 0.053215   | 0.03873  | 0.028853 | 0.069499 |
| 아모레퍼시픽  | 0.059277       | 0.131674   | 0.035836 | 0.090534 | 0.029101 |
| CJ대한통운  | 0.11927        | 0.019631   | 0.083091 | 0.156952 | 0.151898 |
| GS건설    | 0.084567       | 0.073585   | 0.080081 | 0.160966 | 0.045804 |
| 총합      | 1              | 1          | 1        | 1        | 1        |

Table 11: Max Sharpe Ratio Portfolio Stock Weights

\* 데일리블록체인 : 2017.02.13 - 2017.02.21

\* 소리바다 : 2016.08.19 - 2016.09.20

\* 아모레퍼시픽 : 2015.04.22 - 2015.05.07

- 액면분할 날짜

\* 삼성전자 : 2018.05.04

\* NAVER : 2018.10.12

\* 아모레퍼시픽 : 2015.05.08

- Figure 4

Moving Average, Support Vector Regression (RBF kernel), Prophet, RandomForest 모델로 예측한 SK하이닉스의 Validation 값과 실제 Validation 값 비교

- Figure 5

Moving Average, Support Vector Regression (RBF kernel), Porphet, LSTM 모델로 예측한 15개 종목의 Test 값과 실제 Test 값 비교

- Figure 6

예측한 15개 종목의 Test 값을 사용하여 만든 효율적 투자선

- Figure 7

실제 15개의 종목의 Test 값을 사용하여 만든 효율적 투자선

$$\begin{aligned} \text{actual values} &: \{y_t ; t = 1, \dots, T\} \leftarrow \\ \text{two forecast values} &: \{\hat{y}_{1t} ; t = 1, \dots, T\}, \{\hat{y}_{2t} ; t = 1, \dots, T\} \leftarrow \end{aligned}$$

$$\begin{aligned} e_{it} &= \hat{y}_{it} - y_t, i = 1, 2 \leftarrow \\ g(e_{it}) &= \exp(\lambda e_{it}) - 1 - \lambda e_{it} : \text{loss function } (\lambda = \text{positive constant}) \leftarrow \\ dt &= g(e_{1t}) - g(e_{2t}) \leftarrow \end{aligned}$$

$$\begin{aligned} d &= \sum_{t=1}^T dt \leftarrow \\ f_d(0) &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \gamma_d(k) \leftarrow \end{aligned}$$

$$\begin{aligned} H_0 &: E(dt) = 0 \leftarrow \\ \text{under } H_0, & \frac{d}{\sqrt{\frac{2\pi f_d(0)}{T}}} \rightarrow N(0, 1) \leftarrow \end{aligned}$$

Figure 2: dm

## 9 Reference

- Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. International Journal of forecasting, 13(2), 281-291  
 파이썬 증권 데이터 분석 (김황후 지음, 한빛 미디어, 2020)  
<https://github.com/HvyD/HMM-Stock-Predictor>  
<https://randerson112358.medium.com/predict-stock-prices-using-python-machine-learning-53aa024da20a>  
<https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/>

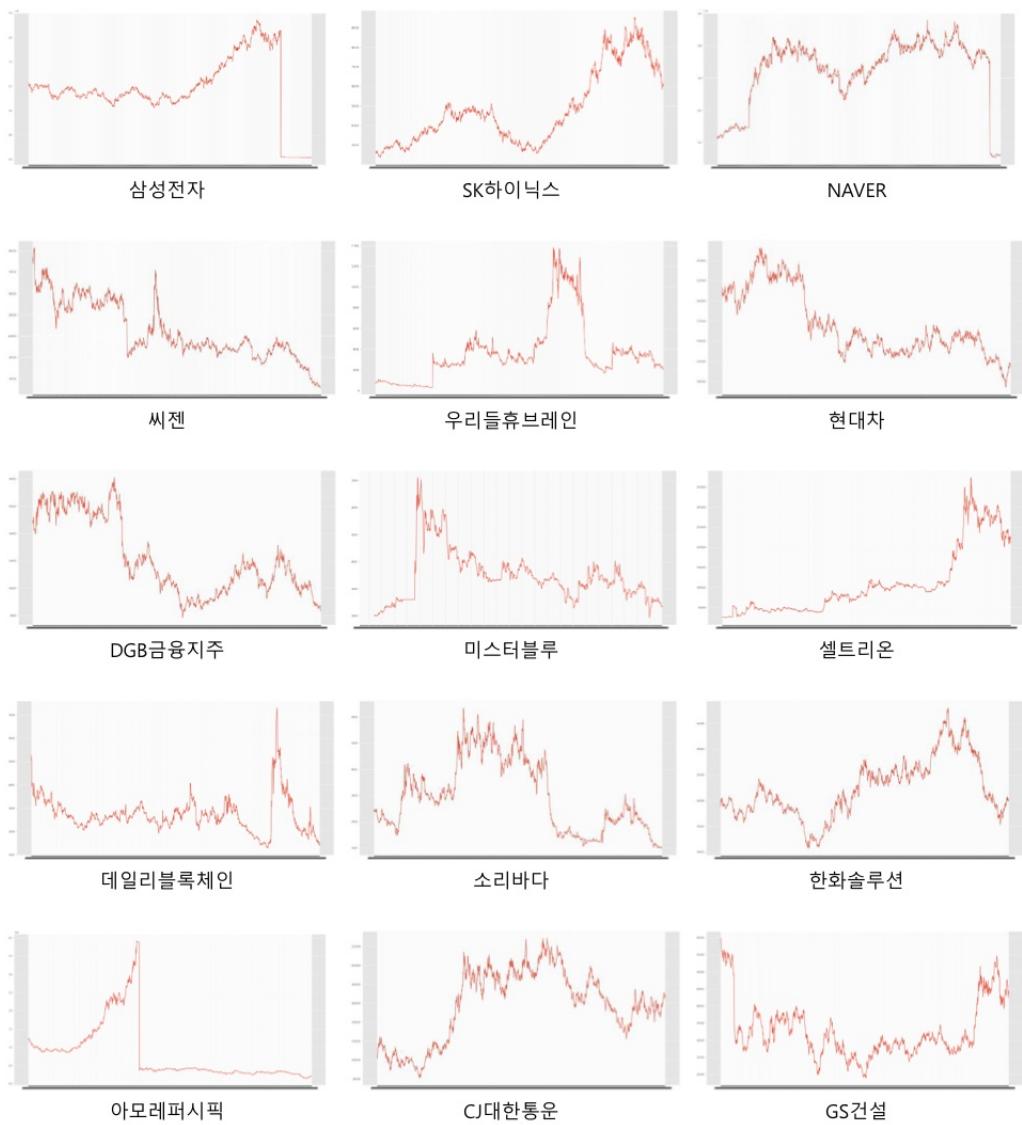


Figure 3: EDA

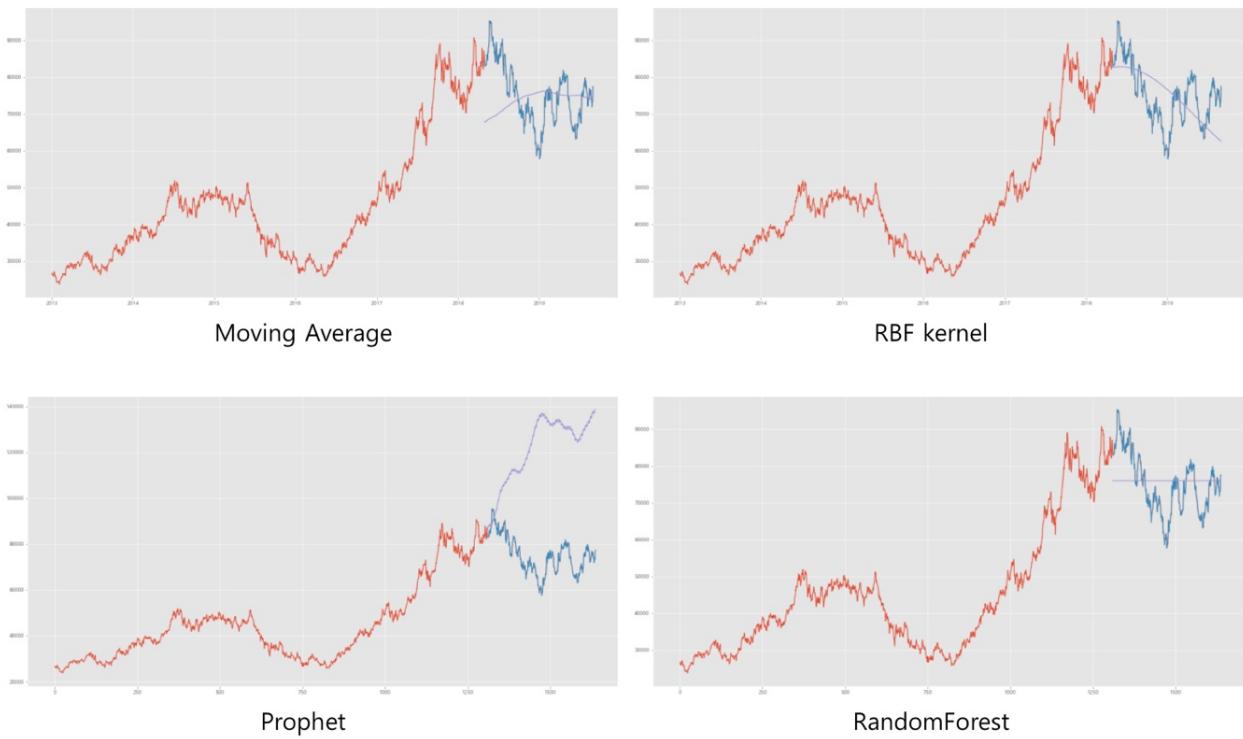


Figure 4: Actual Validation Set vs Predicted Validation set

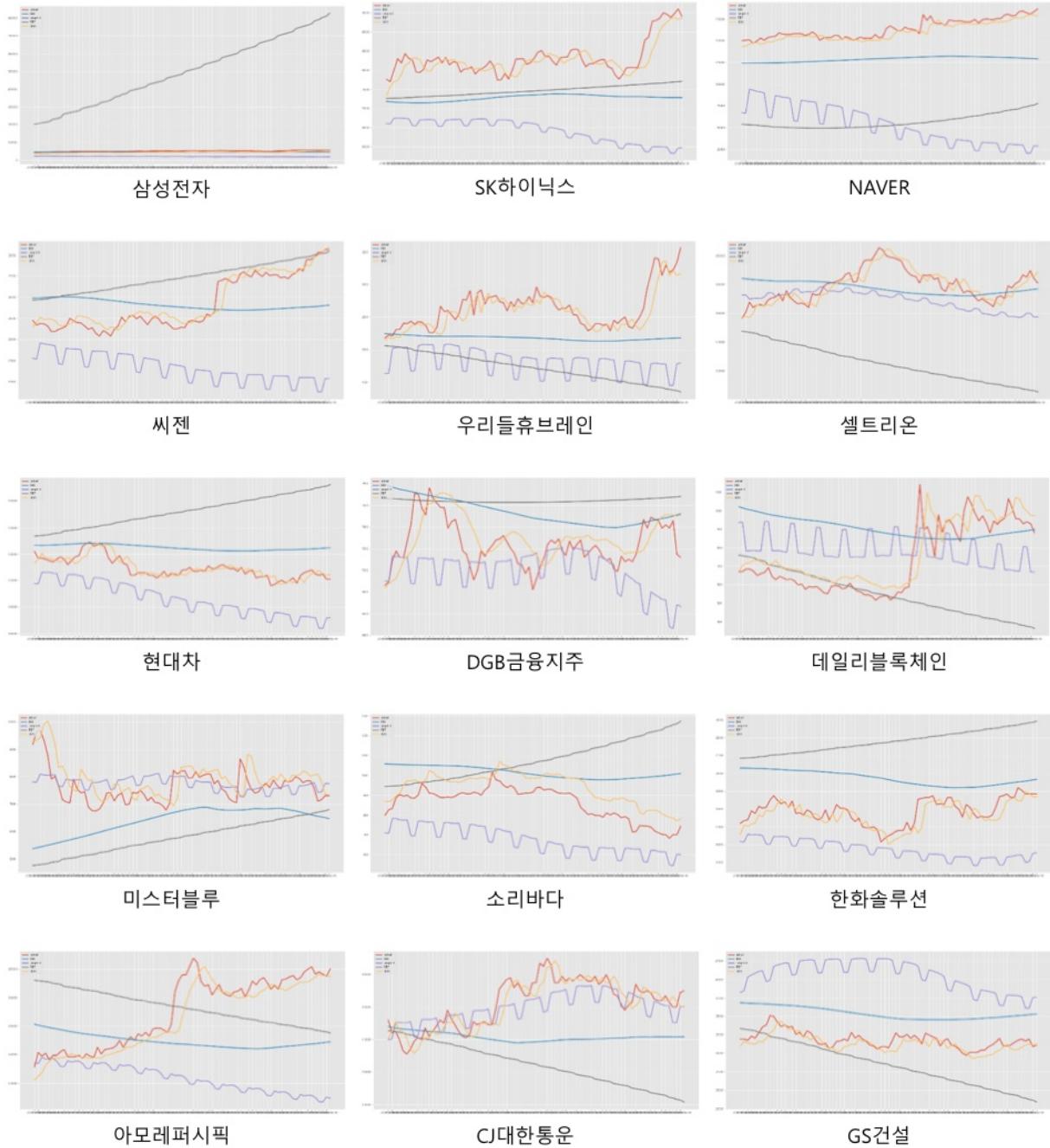


Figure 5: Actual Validation Set vs Predicted Validation set

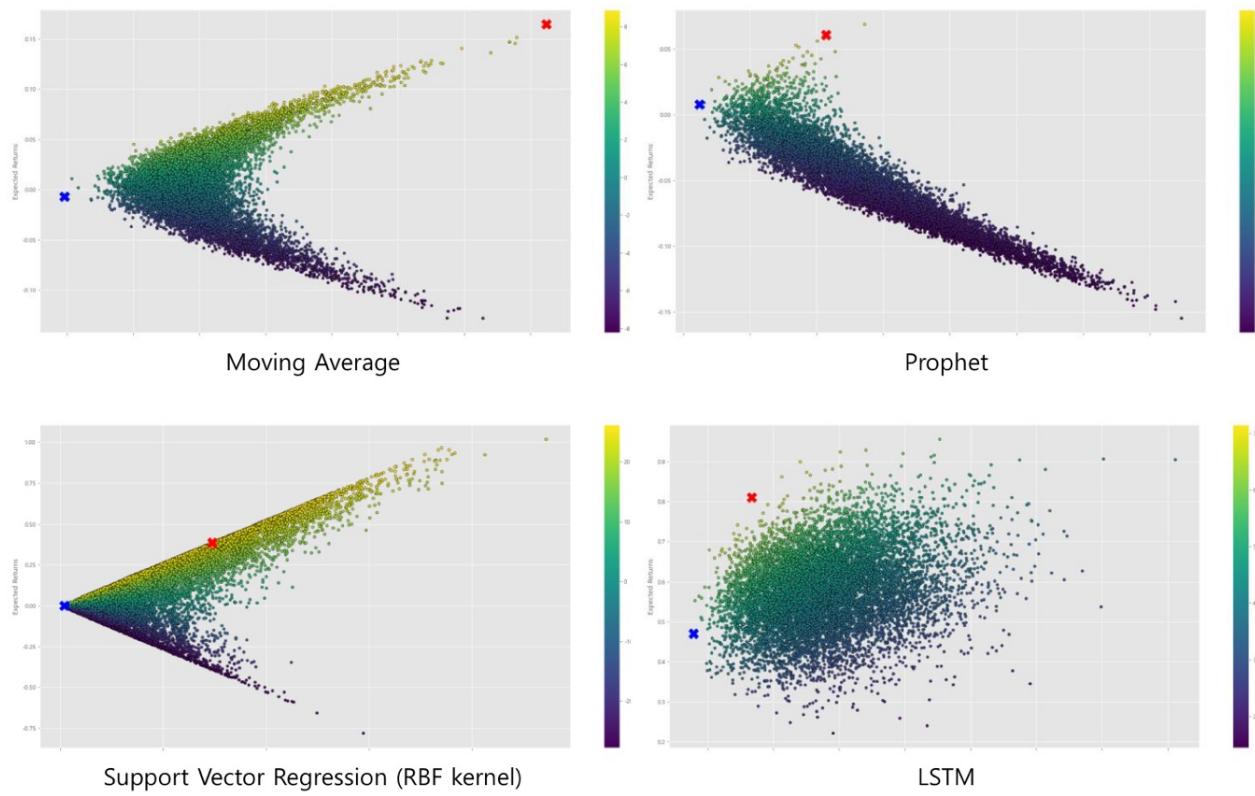


Figure 6: Global Minimum Variance Portfolio Using Predicted Stock

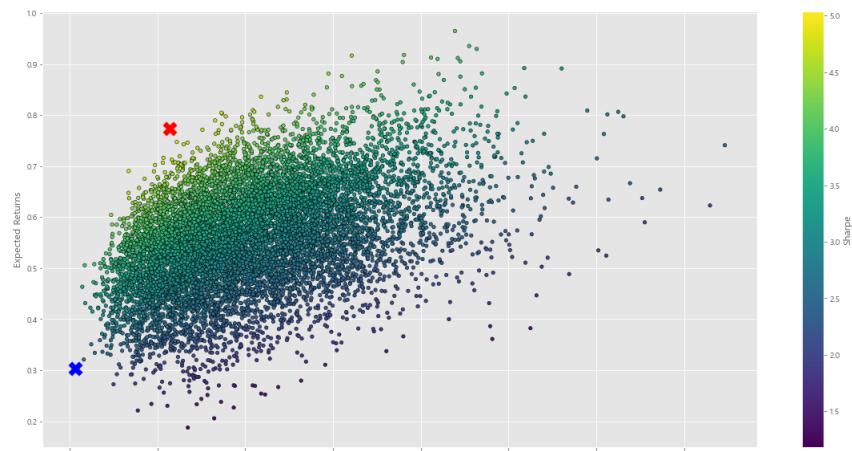


Figure 7: Global Minimum Variance Portfolio Using Actual Stock