creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

**저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.**

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

컴퓨터공학 석사 학위논문

# 암호 화폐 시장의 횡적 수익 분석에서 비선형적인 데이터 스케일링 방법의 영향에 대한 실험적 연구

2023 년 2 월

서울대학교 대학원

컴퓨터공학과

김형수

# 암호 화폐 시장의 횡적 수익 분석에서 비선형적인 데이터 스케일링 방법의 영향에 대한 실험적 연구

지도 교수 문병로

이 논문을 컴퓨터공학 석사 학위논문으로 제출함
2022 년 12 월

서울대학교 대학원
컴퓨터공학과
김형수

김형수의 컴퓨터공학 석사 학위논문을 인준함
2022 년 12 월

위 원 장 _____ 이 재 욱 _____ (인)

부위원장 _____ 문 병 로 _____ (인)

위 원 _____ 김 건 희 _____ (인)

# 초 록

자산가격을 예측하는 다양한 방법이 연구되어 왔다. 초기에는 선형적인 예측 모델들이 사용되었고, 자산 가격 데이터의 복잡성 때문에 비선형적인 알고리즘들이 적용되었다. 그 후, 기계학습 알고리즘이 데이터들의 노이즈를 줄이고 상대적으로 자산 가격의 예측 정확도를 높여 많은 관심을 받게 되었다. 최근에는, 컴퓨테이션 파워의 증가로 인하여 Deep learning model을 사용하는 방법들의 연구들이 활발하게 진행되고 있다. 자산가격을 예측하는 문제들 중, 여러 종목들의 상대적인 수익률 분석을 cross sectional analysis라고 한다. 이런 연구에서는 자산가격이 특정 요인에 의해 결정된다고 가정한다. 상대적으로 많이 상승하는 종목과 하락하는 종목들을 잘 구분하면 좋은 팩터라고 볼 수 있다. Fama-French의 3-factor 모델로 시작을 하여, 여러 factor들로 종목의 수익률을 설명하고자 시도돼 왔다. 최근에는 머신러닝과 딥러닝 등을 이용한 모델들이 이를 설명하고자 한다. 그러나, 자산 시장을 구성하는 종목군 및 국가들이 다양하고, 기간의 영향을 크게 받는다. 이런 요인들로 인하여 조건이 바뀌면 모델의 성과도 크게 바뀌게 된다.

금융 데이터들은 노이즈가 많고 여러 outlier들이 존재하는데, 그를 고려한 데이터 전처리 단계에서 Data scaling 방법에 대한 연구가 부족하다. 본 연구에서는 일반적인 자산시장보다 변동성이 큰 암호화폐 시장의 cross sectional return분석에서 데이터 스케일링 방법의 영향에 대하여 연구를 하고, 이들의 중요성을 보였다. 그리고, 여러 모델에서 공통적으로 좋은 데이터 스케일링 방법을 볼 수 있었다. 또 한, 기존 연구들에서는 제한된 데이터 스케일방법만 적용을 하였는데, data scaling method들의 최적화 방법을 통하여 feature와 class에 다양한 data scaling 방법들을 적용하여 성과의 증가를 보였다. 또 한, feature selection을 통하여 이들 최적화 방법의 타당성을 보였다. 그리고 특정 기간에 집중된 실험이 될 수도 있으므로, 이를 방지하기 위하여 rolling back testing method로 여러 기간에 대하여서 실험을 진행하였다. 이를 통해서 연구에서 제시한 최적화 방법들로 다양한 기간에서 모두 좋은 성과를 얻을 수 있었고, 기간에 민감한 금융 데이터에서 안정적인 성과를 얻을 수 있는 방법을 제안하였다.

**주요어** : 암호 화폐, 횡적 수익 분석, 데이터 스케일링 방법, 최적화
**학 번** : 2019-28216

# 목 차

제 1 장 서 론 ..... 1
제 1 절 서론 ..... 1
제 2 절 연구 동기 ..... 3
제 3 절 연구 공헌 ..... 3

제 2 장 배경 이론 ..... 4
제 1 절 데이터 스케일링 방법 ..... 4
제 2 절 기계학습 알고리즘 ..... 6

제 3 장 실험 데이터 및 방법 ..... 8
제 1 절 데이터 ..... 8
제 2 절 실험 방법 ..... 10

제 4 장 실험 결과 및 분석 ..... 11
제 1 절 Data scaling 방법 최적화 ..... 11
제 2 절 Feature selection ..... 18
제 3 절 기간 별 Data scaling 방법의 영향 ..... 20

제 5 장 토의 및 결론 ..... 25

참고문헌 ..... 26

Abstract ..... 28

# 표 목차

[표 1] Feature 계산 방법 ..... 9
[표 2] Data scaling mehtod에 따른 성과 ..... 12
[표 3] Data scaling method 최적화 ..... 15
[표 4] Feature scaling method 최적화 ..... 17
[표 5] Feature selection에 따른 성과 ..... 18
[표 6] Feature importance1 ..... 19
[표 7] Feature importance2 ..... 20
[표 8] 장기간 Data scaling method에 따른 성과 ..... 21
[표 9] 장기간 Class scaling method 최적화 ..... 23
[표 10] RF의 연도별 Feature scaling method 최적화 ..... 24

[표 11] DNN의 연도별 Feature scaling method 최적화..... 24

# 그림 목차

[그림 1] SP500과 BTC의 가격 ..... 9
[그림 2] Feature 간의 상관계수. ..... 10
[그림 3] 실험 과정 Flow chart ..... 11
[그림 4] Sample 별 Minmax scaled data 분포 ..... 12
[그림 5] 성과 비교 ..... 13
[그림 6] Feature와 class간의 상관계수..... 14
[그림 7] Saling method에 따른 data 분포..... 16
[그림 8] 일반적인 Factor들과 성과 비교..... 22

# 제 1 장 서 론

## 제 1 절 연구 배경

자산 가격의 예측 문제는 다양한 방법으로 시도가 되었다. 금융 데이터는 time series의 성격을 가지고 있기에, 자산 가격을 예측할 때는 과거 data들을 feature로 사용하여 모델을 학습시킨다. 초기에는 autoregressive같은 linear model을 사용하였다.[1] 그 이후, 금융 데이터에서의 nonlinear 특징과 time series 특징을 이용하기 위하여 GARCH 모델 등을 사용하였다.[2] 또 한, 자산 데이터의 노이즈 및 복잡성 때문에 KNN,SVM 그리고 Neural network 기계 학습들이 사용되었다.[3] 이런 모델들은 자산들의 비선형적인 pattern을 catch하며 상대적으로 높은 성능을 보여주며 각광을 받았다 특히 최근에는 computation power가 증가하여 더 많이 사용된다. 많은 연구에서 특정 종목의 output을 예측하는데 있어서, 그 종목만의 feature를 사용한다. 하지만, 자산시장 종목의 output은 그 종목뿐만 아니라 다른 종목에도 영향을 받는다. 여러 종목들을 합해 하나의 종목처럼 거래가 가능하게 한 수단인 ETF의 방향성을 예측하는 논문도 있다. 이 논문에서는 Random forest의 성능이 DNN, KNN, SVM보다 우수하다 주장하며, 개별 종목의 feature 뿐만 아니라 다른 종목의 feature를 사용할 경우 방향 예측성이 증가한다고 한다.[4] 그리고 Gaurav는 global ETF 종목들의 return을 예측할 때, 여러 global macro factor들도 feature로 사용하여 성과의 향상을 보였다.

Cross sectional return 분석은 여러 종목들의 cross sectional data를 이용하여서 분석을 한다. 이는 주식들의 상대적인 수익률을 설명하고자 한다. Fama-French의 연구가 대표적으로, 이들은 3~5개의 factor로 주식의 수익률을 설명하였다.[5,6] 그 이후로, 많은 factor들이 발표 되었다. 많은 factor들과 주식 수익률의 관계성을 설명하는데 linear regression model을 많이 사용하였지만, 최근에는 deep learning, random forest, SVM 등의 방식이 사용되고 있다.[7,8] Kelly 등은 미국 주식시장에 대하여 Neural Network, random forest, 그리고 linear regression 등의 방법을 이용하여 cross sectional return 분석을 진행하였고 Machine learning 방법을 통하여 bench mark대비 더 좋은 성과를 낼 수 있다고 주장하였다.[9]

머신러닝 및 딥러닝을 학습하기 전에는 data들의 정제 위한 preprocessing step이 필요하다. 이런 preprocessing단계는 data cleansing, feature selection 그리고 data scaling 등이 포함된다. 여러 분야에서 data scaling method 방법의 영향에 대해서 연구하였다. Ahsan 등은 heart disease detection problem에서 여러 machine learning 기법들과 data scaling method를 사용하여 scaling method의 중요성을 보였다.[10,11,12]

암호화폐 시장은 최근 성장하고 있는 자산 시장의 한 종류이다. 암호화폐 시장은 높은 변동성을 가지고 있다. 이들의 가격 예측을 위하여 전통적인 time series 예측 모델인 ARIMA 방식[13,14,15]을 사용하기도 하고, machine learning, Neural network 그리고 deep learning 방식들인 GRU, LSTM등을 이용한 방식들로 가격 예측을 하는 연구가 나오고 있다.[16]

본 연구에서는 data scaling method 방법의 최적화를 통하여서 암호화폐 시장에서 cross sectional return 분석을 할 때의 영향을 실험 및 분석하고, data scaling 방법 적용의 응용 방법에 대하여 제안하고자 한다. 이를 위하여서, Minmax scaler(MM), Max absolute value scaler(MA), Standard scaler(SS), Robust Scaler(RS), Power transformer(PT), Gaussian transformer(GT), Rank transformer(RT), Quantile transformer(QT) 등의 data scaling method들을 사용하였다. 그리고 예측 모델로는 Linear regression(LR), random forest(RF), K-Nearest Neighbors(KNN), Support Vector Machine(SVM) 그리고 Deep Neural Network(DNN) 방법들을 사용하였다. 이를 통하여서, cross sectional return 분석에서 data scaling 방법의 최적화의 중요성을 보였다. 각 machine learning algorithm에서의 data scaling 방법에 따른 성과 차이가 컸고, nonlinear data scaling 방법들의 성과가 우수하였다. 하지만, 최적화를 통하여 linear data scaling 방법들 역시 사용하는 방법에 따라 성과의 향상에 영향을 미치는 것을 보였다. 그리고 feature importance를 통하여 이런 방법의 타당성에 대하여 토의하였다. 또 한 test 기간을 변경하며 여러 기간에 대해 결과들을 보여, 기간에 민감한 금융 데이터 실험에서도, 안정적인 성과를 보이는 방법들을 제안하였다.

## 제 2 절 연구 동기

본 연구의 연구 동기는 세가지가 있다. 첫 번째, 자산시장의 cross sectional return에 대한 연구 중에서, 데이터들의 분포를 조절해주는 data scaling 방법의 영향에 대한 연구가 없다. 제 1절에서 언급한 거와 같이, 금융데이터는 노이즈가 많고 아웃라이어가 많다. 그리고 결과와 데이터간의 관계가 복잡한 경우가 많다. Data scaling은 data들을 model이 학습하기 적절한 scale로 바꾸어주고, nonlinear data scaling방법은 다양한 분포로 data를 변환한다. Feature와 output의 data분포는 모델의 학습에 영향을 끼치기에, 이들에 관한 최적화 연구는 중요할 것이다. 두 번째, 다른 분야들에서 데이터 스케일링 방법들의 영향을 살펴보았지만, 대부분 특정 데이터 스케일링 방법을 머신 러닝 모델들에 적용하며 그 결과를 본 경우들이 대부분이다. [10,11,12] 본 연구에서는 데이터 스케일링 방법들의 다양한 응용 방법을 알기 위하여, 여러 데이터 스케일링의 단순 적용뿐만 아니라, Feature와 class간의 데이터 스케일링 방법들을 hyper parameter로 고려하여 최적화를 진행하였습니다. 세 번째, 암호화폐 시장에서의 횡적 가격 분석에 대한 연구 부족이다. 암호화폐 시장은 최근에 급속도로 성장하고 있는 시장으로 연구가 많이 진행되고 있다. 하지만, 연구들은 주로 BTC, ETH과 같은 특정 종목들의 가격 예측에 집중하고 있다. cross sectional return 분석에 대한 연구는, 주식시장의 초기 cross sectional return 분석처럼 소수의 factor들이나 통계적인 방식의 접근들에 머물러 있다. [16,17]

## 제 3 절 연구 공헌

본 연구의 공헌은 다음과 같습니다. 첫 번째, 암호화폐 시장의 cross sectional return 분석에서 data scaling 방법의 영향이 크다는 것을 보였다. 그리고 이 과정에서, 예측 모델과 관계없이 일반적으로 좋은 데이터 스케일링 방법에 대하여 확인하였다. 두 번째, grid search를 통한 최적화로 feature와 class의 data scaling 방법을 다양하게 적용하는 방식에 대하여 제안하였다. Class도 그들에게 더 적합한 data scaling 방법이 존재하며, 다양한 데이터 scaling 방법의 조합을 통한 최적화로 성과의 증가가 가능한 것을 보였다. 위에서도 언급한 바와 같이 금융 데이터는 복잡하기에, feature와 class간의 관계를 더 잘

표현해주는 data 분포가 있기 때문에 이런 성과의 증가가 있었을 것이다. 세 번째, 서로 다른 특성을 가지는 data scaling method들의 조합을 통한 최적화로 성과가 증가할 수 있음을 보였습니다. 금융 데이터는 아웃라이어가 많아, 다양한 분포를 가지고 있다. 데이터 스케일링 방법들에 따라서, 데이터 분포를 잘 보여주는 Minmax scaler 같은 선형적인 방법과 데이터의 분포를 고려하지 않는 Rank transformation 같은 방법들이 존재한다. 기존 연구들에서는 데이터 스케일링을 할 시 한가지 방법으로 진행을 하였고 이들을 Feature로 사용하였다. 본 연구에서는 다양한 데이터 스케일링 방법들을 조합하여 이들간의 시너지 효과를 확인하였다. 그리고 Feature importance를 통하여서 여러 data scaling 방법들을 조합하여 최적화하는 방법의 타당성을 보였다.

# 제 2 장 배경 이론

## 제 1 절 데이터 스케일링 방법들

Data scaling 방법은 data preprocessing 과정 중의 하나로, 목적은 데이터를 모델링에 적합한 범위 및 형식으로 통일하는 것이다. 서로 다른 scale을 가지는 feature들의 scale들을 통일하여 feature들의 크기 차이로 인한 영향력을 없애려 한다. data scaling후에 학습을 더 빠르게 하고 지역 최적화에 빠지는 위험이 줄어들게 된다. 이런 이유들로 인하여적절한 scaling 과정을 거친 후 model들을 학습시킬 경우 더 좋은 성과를 나타낸다. [19]

Data scaling 방법은 변환 방법과 데이터 분포에 의한 영향 정도에 따라 나눌 수 있다. 변환 방법의 관점에서 MM, MA, SS 및 RS 방법들은 선형적인 변환 방식이다. 반면에, RT, PT, QT 및 GT는 비선형적인 변환 방법이다. 데이터 분포에 대한 영향 관점으로는 MM, MA가 가장 큰 영향을 받는다. SS, RS 와 PT는 그 영향이 상대적으로 적다. RT, QT, GT는 영향을 받지 않는다. 각 Data scaling 방법들의 특징은 아래와 같다.

2.1.1. Min-max scaler (MM)

Min-max scaler는 데이터의 크기를 0~1사이로 맞추게 된다. 데이터 중 제일 작은 값을 0으로, 데이터 중 제일 큰 값을 1으로 하여, 데이터들을 mapping 한다. Max와 Min boundary로 사용하므로 outlier가 존재할 시 그 영향을 크게 받는다.

$$MM(X) = (X - X_{min}) / (X_{max} - X_{min})$$

2.1.2. Maximum absolute value scaler (MA)

Max-absolute value scaler는 데이터 셋 중 절대값이 제일 큰 값으로 각 데이터 값을 나눠 준다. MM 방식과 마찬가지로, 제일 큰 절대값의 크기에 의존하므로 outlier의 영향을 많이 받는다.

$$MA(X) = X / X_{abs_max}$$

2.1.3. Standard scaler (SS)

Standard scaler는 dataset의 분포의 평균을 0으로 하고, standard deviation으로 나눠주는 방법이다. 이 방법 역시, outlier가 있다면 평균과 표준 편차에 영향을 미쳐 데이터 분포에 영향을 준다.

$$SS(X) = (X - mean(X))/STD(X)$$

2.1.4. Robust scaler (RS)

Robust scaler는 minmax 방법에서 min 대신 $25^{th}$ quartile값을 사용하고, max 대신 $75^{th}$ quartile 값을 사용한다. 이를 통하여 outlier의 영향을 줄이려고 한다.

$$RS(X) = (X - Q1(X))/(Q3(X) - Q1(X))$$

2.1.5 Power transformer (PT)

데이터 셋의 값을 log transformation을 진행한다. 치우친 데이터의 분포를 정규 분포로 변환한다. 그에 따라 outlier의 영향이 상대적으로 덜해진다.

2.1.6 Rank transformer (RT)

데이터 셋의 값을 크기에 따라 순위를 매긴다. 그 후 이들을 0~1 사이의 값으로 변환을 해준다. Transformation 시, 데이터의 순서만 보고 크기를 보지 않으므로 outlier에 민감하지 않다. 하지만, 데이터의 분포에 대한 정보는 없다.

2.1.7 Gaussian transformer (GT)
RT 방법으로 변환된 데이터 분포를 Gaussian 분포로 변환한다. 이 역시 nonlinear 변환 방법으로, outlier들의 영향을 받지 않는다.

2.2.8 Quantile transformer (QT)
데이터를 크기 순서에 따라서 몇 개의 그룹으로 변환한다. 본 실험에서는 5개의 group으로 나누었다. 그리고 이를 0~1 사이의 분포로 변환한다. Outlier들의 영향을 받지 않는다.

## 제 2 절 기계 학습 방법들

본 연구에서 사용하는 기계 학습 방법들은 LR, SVM, KNN, RF 그리고 DNN을 사용하였습니다. Liew and Mayster[4]는 SVM, RF 그리고 DNN을 n일 뒤의 etf들의 가격 방향성 예측 문제에 적용하였습니다. 이 실험에서 결과는, 예측 일수와 etf 종목에 따라서 다르지만, RF>DNN>SVM의 성과를 보여주었습니다. Gu와 Kelly[9]는 LR, RF, GBRT 그리고 DNN 등의 방법들을 미국 주식 시장의 cross section return analysis에 적용하였다. 이 실험에서는, DNN>RF>GBRT>LR 순서의 성과를 보여주었다. 그리고 머신러닝 방법들이 cross sectional return 분석의 성과를 높이는 것을 보였다. Abe와 Nakagawa[7]는 TOPIX500에 속한 일본 주식들의 데이터를 이용하여서 cross sectional return 분석을 진행하였다. 그들은 DNN, RF, Linear regression 방법을 사용하였고 DNN의 성능이 가장 좋게 나왔다. 각 머신러닝 방법들의 특징은 아래와 같다.

2.2.1. Linear regression
Linear regression은 제일 기본적인 regression 모델이다. 이 모델은 class와 feature간에 straight line 관계를 fit한다. $y = wX + b$로 식을 나타낼 수 있다. w' 는 가중치를 의미하고, b는 error term을 의미한다. X가 하나의 feature만을 나타낼 때는 단순 선형 회귀 분석이라 하고, 여러 개일 때는 다중 선형 회귀라고 이야기한다. 본 실험에서는 여러 Linear regression 방법들 중 Linear regression, Ridge regression 그리고 Bayesian ridge regression으로 실험을 진행하였다. 세 방법의 성과 차이가 크지 않았으며, 이들 중 Linear regression에 대한 결과를

보고하였다.

2.2.2. SVM

Support vector machine은 데이터들의 그룹간 제일 큰 차이를 만들 수 있는 support vector를 찾는 것이 목적이다. 이것을 Margin이라 하며, 이를 최대화 하려한다. SVR의 장점은 outlier에 영향을 덜 받는 것이다. 또 한, kernel trick을 통하여서 nonlinear 관계들에 대한 분석을 할 수 있다. 이는 data를 더 높은 dimension에 두고, 다시 이를 원래 dimension 공간에 projecting 하는 방법이다.

2.2.3. KNN

KNN은 distance function등의 similarity를 base로 예측을 하는 알고리즘이다. 새로운 데이터 포인트가 들어올 때, training set의 데이터들 중 가장 가까운 K개의 데이터의 평균값으로 그 값을 예측한다. 그러므로, feature간의 크기 차이가 날 경우에는 특정 feature에 대한 의존도가 커질 수 있다.

2.2.4. Random forest

의사결정나무[18]는 트리 구조의 해가 주어진 모든 데이터를 구분 기준을 통해 점진적으로 분류한다. 트리의 끝까지 데이터가 분류될수록 보다 자세한 분류가 가능해진다. 각 노드에서 분류할 때 최대한 레이블 수가 잘 나누어지도록 분류를 반복한다. 이런 decision tree는 쉽게 over-fitting이 된다. 이를 방지하기 위하여 tree의 깊이, terminal node의 개수를 방지하는 등의 방법들이 제안되었다. Random forest는 위의 over-fitting 방법들과 다른 방향의 방법을 제안하였다. 이는bootstrapping 방식으로, 주어진 데이터를 여러 번 샘플링 하고, 각각 트리를 만들어 평균을 통해 값을 구한다. 본 연구에서는, random forest의 hyper parameter인 n_estimator = {100,200,300}, max_depth = {5,10,15}를 조합하여 사용을 하였다. 그리고 그들 중 n_estimator = 200, max_depth = 10을 적용한 경우에 대하여 성과를 report 하였다.

2.2.5 Deep Neural Network

Deep neural network는 hidden layer가 1개 보다 큰 neural network를 의미하며, input layer, hidden layers 그리고 output layer로

구성되어 있다. 각 node는 weight과 bias의 parameter들을 가지고 있으며, loss function을 최소화해가며 gradient descent 방식을 통하여 parameter 최적화를 진행한다. 각 node에서 input data와 weight들의 곱이 일어나며 합쳐진다. 그리고 이를 activation function을 거쳐 다음 layer로 보내진다. 본 연구에서는, 적절한 DNN model을 구하기 위하여 hidden layer의 갯수 및 각 layer의 node 숫자 그리고 dropout 비율 등의 hyper parameter들을 바꿔가며 실험을 진행하였다. 그 중 5개의 층을 가진 DNN을 사용한 경우에 대하여 성과를 report 하였다.

# 제 3 장 실험 데이터 및 실험 방법

## 제 1 절 실험 데이터

본 연구에서는 암호화폐 시장에서 Data scaling 방법의 영향을 확인하기 위해 Upbit에 상장된 118개 암호화폐의 일별 가격 데이터를 수집하여 사용하였다. 암호화폐의 가격은 다른 자산 시장에 비해서 가격 변동이 매우 컸다. 2020년 S&P500이 16.26% 상승하는 동안 234.7%의 상승을 하였고, 2021년 S&P500이 26.89% 상승하는 동안 약 1000%에 가까운 상승을 보여주었다. 그리고 2022년에 S&P500이 −20% 정도의 하락하는 동안 약 −70%정도의 하락을 보여주었다. 그림 1은 2017.10.16~2022.08.01 기간 동안의 BTC와 sp500 index의 cumulative log price 가격을 보여준다. BTC의 변동성이 sp500에 비하여 크다는 것을 볼 수 있다. 이런 변동성으로 인하여, 암호화폐 데이터들의 분포는 outlier에 대한 영향이 더 크다. 그렇기에 data scaling 방법의 최적화는 중요한 문제일 것이다.

암호화폐의 데이터는 종가, 고가, 저가, 거래량으로 이루어져 있다. Raw data를 그대로 사용하는 경우 시장의 흐름을 잘 표현하지 못하기에 아래 표 1과 같이 전처리를 수행하여 이들을 Feature로 사용을 하였다. 그리고 각 Factor들에 대하여 t = 1~15로 계산하여, 총 60개의 feature를 사용한다.

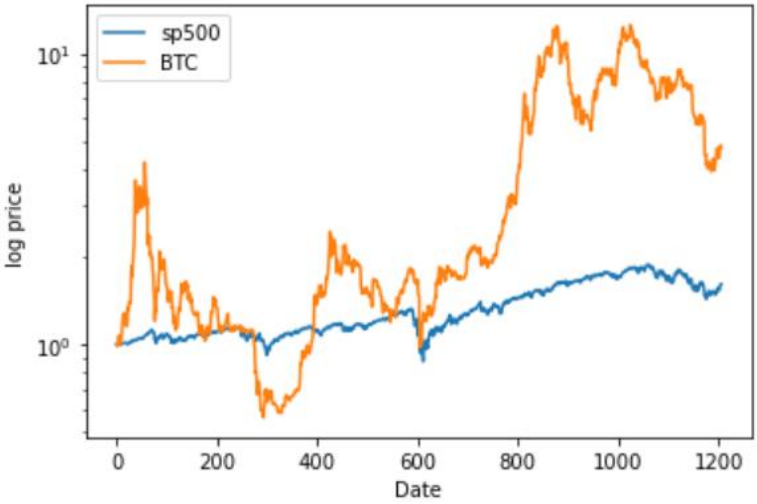


그림 1. 2017.10.16~2022.08.01 기간 동안의 sp500과 BTC 가격

| Features | Description |
|---|---|
| mom(t) | $(price - past(price,t))\ /\ past(price,t)$ |
| DMom(t) | $(past(price,t) - past(price,t-1))\ /\ past(price,t-1)$ |
| Tval(t) | $past(price,t) * past(volume,t)$ |
| IBS(t) | $(price - min(low,t))\ /\ (max(high,t) - min(low,t))$ |

표 1. 각 Feature의 계산 방법. Price, high, low는 각각 예측 시점의 종가, 고가, 저가를 의미하며 volume은 거래량을 의미한다. past(t), min(t), max(t) 함수는 각각 t일 전의 값, t일간의 최소값, 최대값을 의미한다

Mom(t)는 t일 기간 동안의 가격 변화율을 계산하며, 가격의 움직임을 표현한다. DMom(t)는 t일전의 1일 가격 변화율을 계산하며 가격의 변동성을 표현한다. Tval(t)는 t일전의 거래대금을 의미하며 종목의 유동성을 의미한다. IBS(t)는 t일간의 가격 범위에서 현재 가격의 상대적인 위치를 표현한다. 그림2에서는 이들간의 상관계수를 보여준다. Mom(t)간의 상관계수는 높은 편이며, mom(t)와 dmom(t) 및 tval(t)의 상관계수는 낮은 편이다. Dmom(t)는 daily 변화율을 보여주며 상대적으로 다른 feature들과의 상관계수가 낮다. Tval(t) 또한, tval(t)들을 제외하면 다른 feature들과 낮은 상관계수를 보인다. 이런 feature간에 낮은 상관 계수는 문제 공간에 대하여 더 다양하게 표현할 수 있을 것이다.

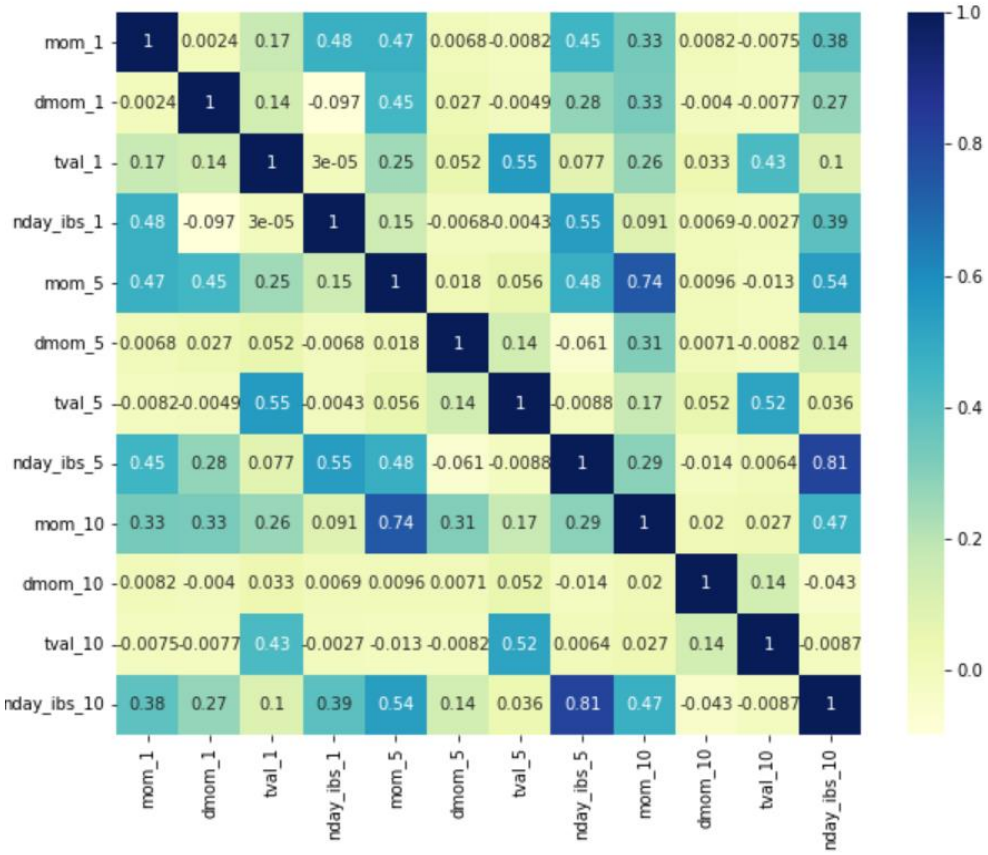


그림 2. Feature간의 상관계수

## 제 2 절 실험 방법

Cross sectional return 분석에서 성과를 측정하는 방법은 일반적으로 두 가지이다. 첫 번째는 Long portfolio의 수익률을 성과로 측정하는 것이고, 두 번째는 Long/short portfolio의 수익률을 성과로 측정하는 것이다. Long portfolio는 예상하는 수익률이 높은 상위 분위의 종목들로 구성된 portfolio로, 이들의 누적 수익률을 성과로 측정하고 비교한다. Short portfolio는 예상하는 수익률이 낮은 분위의 종목들로 구성된 portfolio다. Long/short portfolio는 Long portfolio의 수익에서 short portfolio의 수익을 빼는 방식으로 성과를 측정한다. 본 실험에서 성과의 측정 방법을 Long portfolio방식을 택하였고, 그 과정은 아래와 같다.

실험을 진행하기 위해서 먼저 제1절과 같이 데이터를 얻고 그들을

factor들로 바꿔준다. 그 후, 여러 data scaling 방법을 선택하여 feature/class의 scaling을 진행한 후 Train/Test split을 진행한다. 이때, 전체 dataset의 기간은 2017.08.30부터 2022.07.10까지이다. 이를 train/valid/test = 0.70/0.15/0.15 비율로 나눈다. 그 다음, Machine learning algorithm을 선택하여 Training을 한다. 여기서 class는 하루 뒤의 수익률로 진행을 한다. 모델을 학습 후, validation에서 제일 좋은 성과를 보이는 parameter를 선택하여 test 기간 동안의 성과를 비교한다. 그 후, 예측된 일별 수익률의 상대적인 크기를 기준으로 5개 그룹으로 나눈다. 그리고 각각 그룹에 속한 종목들의 평균 수익률로 각 그룹의 수익률을 측정하고, 제일 높은 수익률을 나타낼 것이라고 예상되는 Q5의 수익률을 성과로 측정한다. 그림 3에서 전체 실험 과정을 표현하였다.

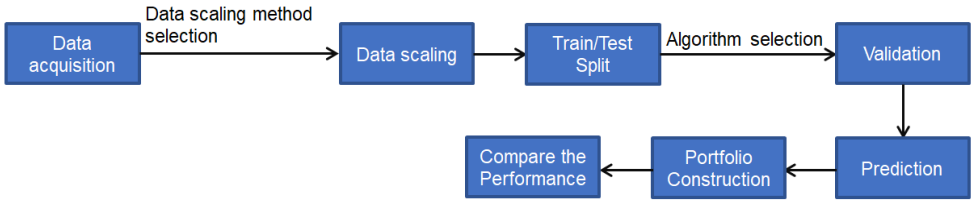


그림 3. 실험 과정 Flow chart

# 제 4 장 연구 결과

## 제 1 절 Data scaling 방법 최적화

### 4.1.1 Data scaling 방법과 model grid search

표 2는 data scaling 방법과 machine learning 모델의 grid search를 진행하였을 때의 성과를 보여 준다. 모든 모델에서 RT, QT 그리고 GT의 순서대로 성과가 좋고, MM와 MA 스케일링 방법들의 성과가 제일 안 좋다. 아웃라이어에 영향을 많이 받는 금융 데이터의 특성상, MM, MA은 그림 4와 같이 data sampling 별로 그 분포가 다양한 것을 볼 수 있다. Momentum과 trading value처럼 다른 feature들의 경우 더 큰 data 분포 차이를 보여준다. 즉, train/valid/test 데이터들 간의 분포가 다를 것이고 이는 모델의 성과에 악영향을 줄 것이다.

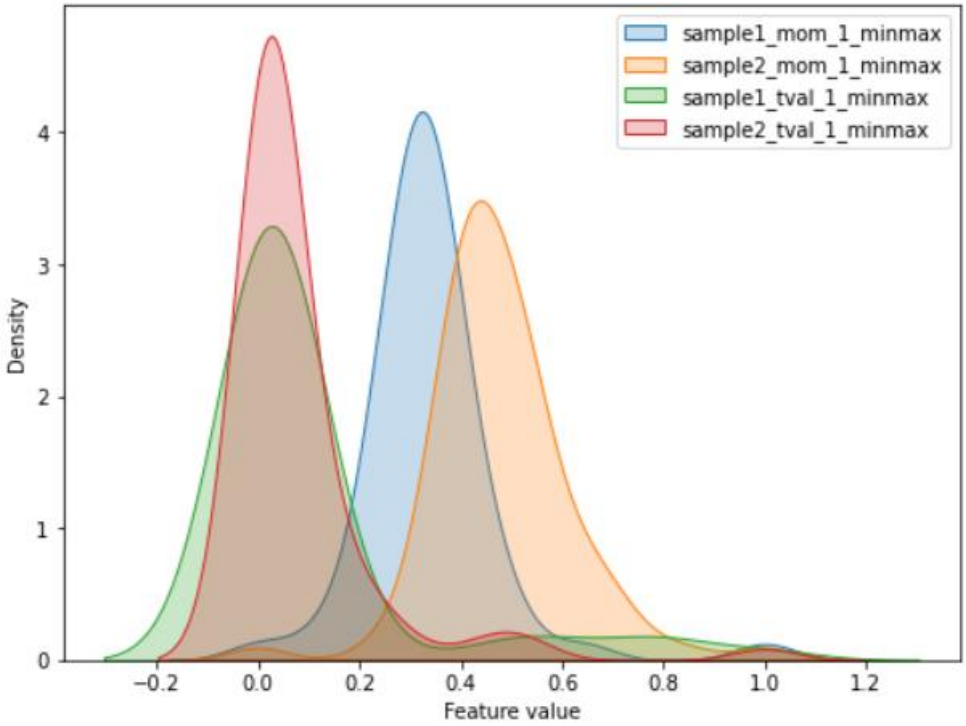

그림 4. Sample 별 Minmax scaled data 분포

대부분의 data scaling method들에서 Linear regression의 성과가 제일 좋지 않게 나왔다. Fama-French의 연구를 비롯한 여러 연구들에서 linear regression을 이용하여서 각 factor들의 영향을 설명하기에, linear regression의 성과는 base line으로 볼 수 있다. 대부분의 data scaling method은 RF를 사용하였을 때, 성과가 제일 좋았고 DNN, KNN, SVR, LR순의 성과를 보여주었다.

| Method | LR | SVR | KNN | RF | DNN |
|---|---|---|---|---|---|
| MM | (52.9) | (37.8) | (66.8) | (56.6) | (43.1) |
| MA | (56.9) | (35.7) | (47.9) | (42.3) | (55.5) |
| SS | (45.5) | (17.5) | (23.9) | (4.6) | (15.5) |
| RS | (38.4) | (9.9) | 14.6 | 23.9 | 0.0 |
| PT | (28.9) | (9.8) | (23.4) | (11.4) | (6.6) |
| GT | (11.3) | (4.5) | 20.8 | 42.0 | 23.5 |
| RT | 4.1 | 21.5 | 24.0 | 51.8 | 26.8 |
| QT | 2.9 | 16.8 | 22.5 | 42.5 | 25.4 |

표 2. Data scaling method와 model에 따른 test 기간 동안 성과 비교(%). 괄호 표시는 음수 값을 의미한다.

이들의 성과를 비교하기 위하여 market 평균 그리고 market anomalies로 알려진 momentum, liquidity 등의 factor들[21,22,23]과 비교를 해보았다. 그림 4는 그 결과를 plotting한다. Market 평균의 수익률은 −60%이고, Momentum을 factor로 쓴 경우들도 성과가 좋지 않았다. 거래대금을 사용한 경우에 성과가 좋긴 하지만, RT를 사용한 machine learning 알고리즘들 보다는 좋지 않은 것을 볼 수 있다. 특히, LR을 제외한 나머지 Machine learning 알고리즘들 보다는 성과가 매우 낮다. 이를 통하여서, 암호화폐 시장의 cross sectional return 분석에서 machine learning 알고리즘을 사용하는 경우 분석 능력의 향상을 확인하였다. 또 한, Data scaling 방법 최적화로 성과가 향상하였다. 그리고 모든 알고리즘들에서 일반적으로 RT방법을 사용하는 것이 좋음을 보였다. 이는 outlier에 영향을 많이 받는 금융데이터의 특성을 없애주기 때문이라 생각한다.

그림 5. 일반적인 factor들과 성과 비교

4.1.2 Feature 와 class의 data scaling 방법 grid search

제 1장에서 언급한 것처럼, 이전 data scaling 방법의 영향에 대한 연구들의 경우 4.1.1의 실험처럼 feature와 class의 data scaling 방법을 동일하게 적용하고, data scaling 방법들과 machine learning

방법들을 바꾸어 가며 실험을 하였다.[10,11,12] 하지만, feature와 class의 scaling 방법이 다를 경우 더 다양한 문제 공간을 만들 수 있기에 이에 대한 영향을 살펴보고자 한다. 그림 6는 feature와 class의 scaling 후 이들의 상관계수를 보여준다. 같은 scaling method을 적용한다고 해서 더 높은 상관계수를 보여주지 않는 경우도 많다. 이를 통하여서 서로 다른 scaling method를 사용하는 경우가 더 좋은 성과를 보일 수 있다 생각하였다. 이를 확인하기 위하여 Feature와 class의 data scaling 방법에 대한 최적화를 진행하였다. Hyper parameter의 최적화 방법에는 grid search, random search, 그리고 Bayesian optimization 등이 있다. Data scaling 방법의 선택은 value를 갖는 것이 아니므로 grid search를 통한 최적화를 진행하였다.

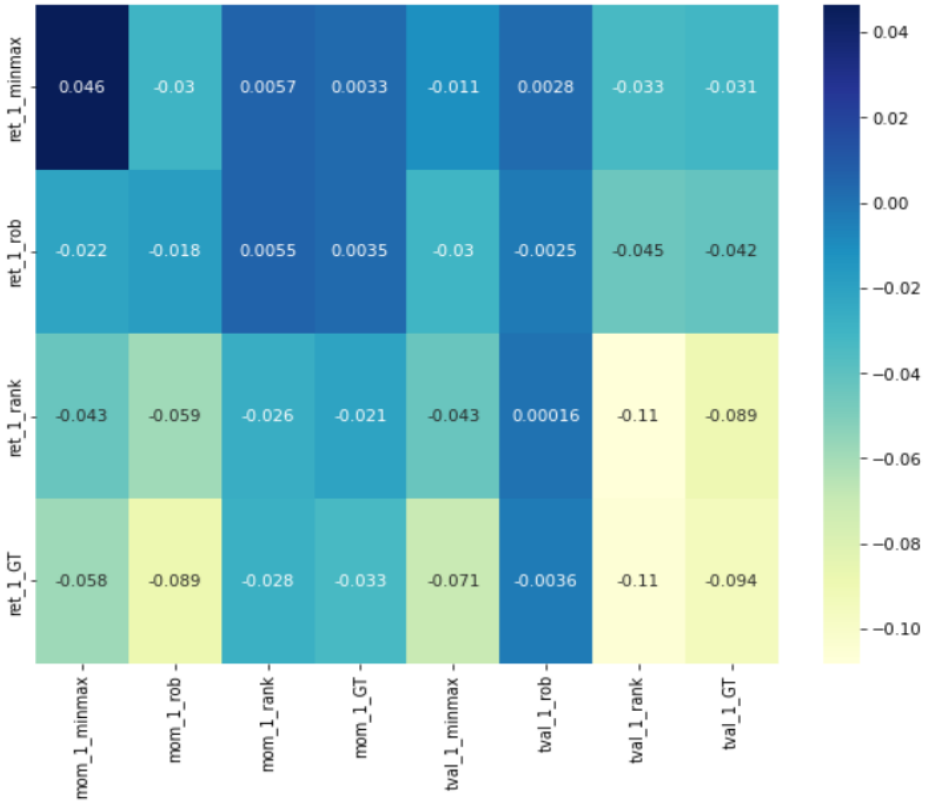


그림 6. 서로 다른 scaling method 적용 후 feature와 class의 상관계수

표 3은 그 성과를 보여준다. 많은 경우에서 feature와 class에 서로 다른 scaling method를 적용할 때 더 좋았고, Feature와 class의 data scaling 방법을 다르게 적용 가능함을 보였다. Feature scaling을

MM으로 적용하는 경우 모든 경우에서 class의 scaling method를 다른 방법으로 적용하면 성과가 더 좋았다. 그리고 Feature의 scaling method가 바뀌어도 대체로 class에 nonlinear scaling 방법을 사용하는 경우의 성과가 좋았고, MM을 사용하는 경우의 성과가 안 좋았다. 이를 통하여서, data scaling 방법을 적용할 때, 각 data의 분포와 모델에 따라 적절한 방법을 적용해야 하고, data scaling 방법의 최적화의 필요성에 대해 알 수 있다.

| Feature scaling | Class scaling | LR | SVR | KNN | RF | DNN |
|---|---|---|---|---|---|---|
| MM | MM | **(52.9)** | **(37.8)** | **(66.8)** | **(56.6)** | **(43.1)** |
| | SS | (46.6) | (17.9) | (34.6) | (6.9) | (23.1) |
| | RS | (45.1) | (16.0) | (30.7) | (15.5) | (21.8) |
| | PT | (41.3) | (10.1) | (14.5) | 22.3 | (2.6) |
| | GT | (45.3) | (16.5) | (16.7) | 24.0 | 3.8 |
| | RT | **(40.9)** | **(11.1)** | **(21.1)** | **24.2** | **10.4** |
| | QT | (40.8) | (8.8) | (23.3) | 19.2 | 5.2 |
| RS | MM | (47.1) | (54.4) | (23.6) | 4.8 | (3.5) |
| | SS | **(46.9)** | **(19.8)** | **11.9** | **31.4** | **3.9** |
| | RS | (38.4) | (9.9) | 14.6 | 23.9 | 0.3 |
| | PT | **(39.6)** | **(10.3)** | **15.1** | **36.5** | **29.0** |
| | GT | (40.8) | (0.3) | 14.4 | 36.4 | 25.5 |
| | RT | **(38.2)** | **(12.6)** | **7.5** | **46.0** | **24.5** |
| | QT | (38.4) | (8.9) | 7.5 | 47.7 | 24.3 |
| GT | MM | (17.9) | (47.3) | (6.7) | 26.4 | (11.7) |
| | SS | (16.6) | (12.6) | 8.9 | 14.2 | (3.1) |
| | RS | (17.4) | (12.8) | 23.2 | 1.9 | (6.5) |
| | PT | **(8.2)** | **(7.2)** | **30.1** | **40.4** | **27.4** |
| | GT | (11.3) | (4.5) | 20.8 | 42.0 | 23.5 |
| | RT | **(4.8)** | **(15.5)** | **15.8** | **42.6** | **26.6** |
| | QT | (3.7) | (18.9) | 19.8 | 40.8 | 24.8 |
| RT | MM | (2.3) | (17.3) | (2.1) | 15.8 | (13.2) |
| | SS | (2.1) | 13.0 | 20.0 | 2.8 | (2.8) |
| | RS | (16.1) | 15.3 | 24.0 | (0.6) | 8.8 |
| | PT | **1.6** | **20.1** | **22.2** | **45.4** | **33.0** |

| | | | | | |
|---|---|---|---|---|---|
| GT | 0.5 | 12.2 | 26.3 | 34.7 | 31.0 |
| RT | **4.1** | **21.5** | **24.0** | **51.8** | **26.8** |
| QT | 3.2 | 12.9 | 22.8 | 35.5 | 26.9 |

표 3. Data scaling method 최적화. 각 알고리즘과 scaler method에 따른 test 기간 동안 Long portfolio의 수익률(%). 괄호 표시는 음수 값을 의미한다.

4.1.3 Data scaling method 조합을 통한 최적화

위의 표 3을 통하여 feature와 class의 scaling method 최적화로 더 높은 성과를 얻을 수 있음을 보였다. 이는 더 다양한 공간을 탐색함으로써 얻을 수 있었고, 이를 feature간에도 적용시키기 위하여, scaling method들을 조합하여 모델의 최적화를 시도하였다. 서로 다른 data scaling method를 적용하였을 때, feature들의 분포는 그림 6와 같다. 서로 표현하는 데이터 분포가 크게 다르기 때문에 이들을 같이 사용하면 다양한 문제 공간을 표현할 수 있을 것이다.

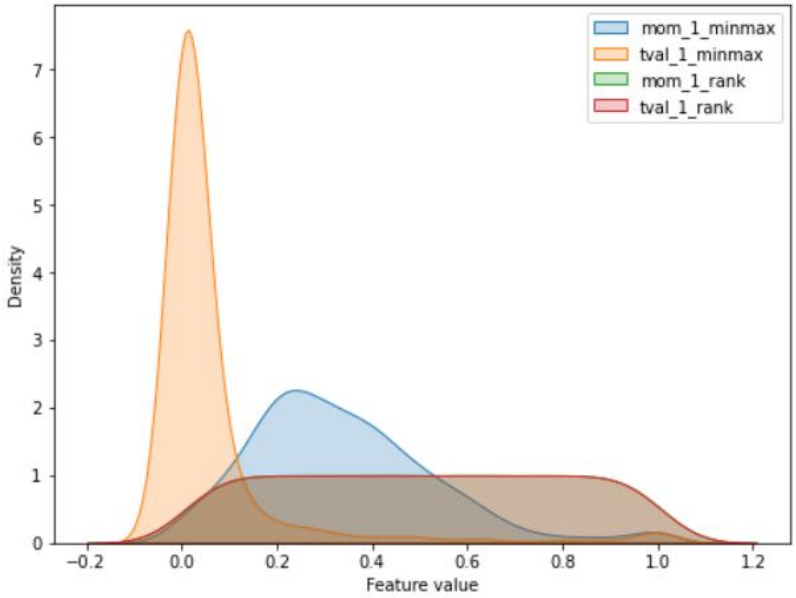


그림 7. Scaling method에 따른 data 분포

Data Scaling 방법들 중에서 MM, RT그리고 QT는 모두 data를 0~1사이의 값으로 변환을 한다. 각 scaling 방법들의 data를 표현하는 방식이 다르기에, 이런 range가 같은 scaling method들을 조합하여 모델의 성과 최적화를 진행하였다. 표 4는 그 성과를 보여준다. Feature

scaling method의 조합을 통한 최적화로 더 좋은 성과를 얻을 수 있었다. MM 방법은 liner scaling이고, outlier에 민감하며, 모든 scaler 방법들 중에서 제일 성과가 좋지 않았다. 반면에 RT 방법은 nonlinear scaling이고, outlier에 영향을 받지 않으며 성과가 좋았다. 이러한 상반된 성격을 가지고 있는 scaling method들을 조합하여 사용하였을 때 향상된 성과를 얻을 수 있었다. 즉, 가장 성과가 안 좋았던 MM 방법을 다른 data scaling 방법과 조합하여 더 좋은 성과를 얻을 수 있는 방법을 제안하였다. 또 한, 이런 Feature의 조합을 통한 최적화와 Class의 scaling method의 grid search를 통한 최적화의 결과에서 class는 다양한 feature들의 조합에도 nonlinear scaling method들의 성과가 좋았다. 이런 성과의 증가의 원인 및 조합을 통한 최적화 방법의 타당성을 알아보기 위하여서 다음 절에서 Feature selection과 Feature importance를 사용하여 살펴보았다.

| Feature scaling | Class scaling | LR | SVR | RF | DNN |
|---|---|---|---|---|---|
| MM | RT | (40.9) | (11.1) | 24.2 | 10.4 |
| RT | RT | 4.1 | 21.5 | 51.8 | 26.8 |
| MM/RT | MM | (27.4) | 2.4 | (45.6) | (33.2) |
| | SS | (10.7) | 13.8 | 10.5 | 3.2 |
| | RS | (19.7) | 16.2 | (1.9) | 9.5 |
| | PT | (5.9) | 10.4 | 48.7 | 30.0 |
| | GT | (0.1) | 13.3 | 40.9 | 27.3 |
| | RT | 2.8 | 10.5 | 48.4 | 36.4 |
| | QT | 1.0 | 13.0 | 50.0 | 32.3 |
| RT/QT | MM | 3.6 | (19.0) | 18.2 | (31.8) |
| | SS | (8.3) | 17.6 | 5.7 | 6.7 |
| | RS | (11.7) | 19.4 | (5.3) | 11.1 |
| | PT | 3.4 | 15.8 | 46.0 | 37.0 |
| | GT | 1.1 | 14.7 | 40.9 | 27.9 |
| | RT | 6.3 | 20.5 | 47.4 | 35.5 |
| | QT | 4.9 | 17.6 | 36.3 | 34.4 |
| MM/RT/QT | MM | (3.5) | (3.4) | (47.0) | **(1.7)** |
| | SS | (15.8) | 19.5 | 12.7 | **(5.3)** |
| | RS | (7.3) | 17.0 | (9.1) | **(0.8)** |

| | | | | |
|---|---|---|---|---|
| PT | (2.4) | 18.4 | 43.3 | **34.5** |
| GT | (1.3) | 22.6 | 38.4 | **32.2** |
| RT | 5.0 | 18.5 | 47.0 | **44.1** |
| QT | 2.7 | 13.4 | 44.1 | **35.0** |

표 4. Feature scaling method 최적화. 각 알고리즘과 scaler method에 따른 test 기간 동안 Long portfolio의 수익률(%). 괄호 표시는 음수 값을 의미한다.

## 제 2 절 Feature selection

4.2.1 Feature selection

표4에서 여러 data scaling method들의 조합과 grid search를 통한 최적화로 성과를 향상시켰다. 그리고 성과가 안 좋은 data scaling 방법도 최적화를 통해서, 모델의 성과를 올리는데 기여할 수 있는 것을 보였다. 이런 성과의 증가가 여러 data scaling method간에 조합을 통한 synergy효과로 인해 발생한 건지 확인하기 위하여 feature selection 방법을 통하여서 feature importance들을 구하였다. Feature들은 MM, RT, QT의 data scaling 방법들을 함께 사용하여 총 180개이다. 이들을 Random forest를 이용한 feature selection을 진행하였다. Feature importance가 높은 feature들의 수를 변화시키며 DNN 모델을 학습시켰다. 그리고 feature들의 수를 변화시켜가며 각 class의 scaling method 별 성과를 살펴보았다.

표 5에서 그 성과를 보여준다. Feature selection을 통하여서 얻은 feature들의 성과는 크게 차이가 났고, 이를 통하여서 class의 scaling method에 따라 중요하게 여겨지는 feature가 바뀜을 볼 수 있었다. 성과가 좋았던 nonlinear scaling method들은 feature의 수가 적을 때도 성과가 좋은 것을 보여주었다. 서로 다른 성과를 보일 때 어떤 feature들을 사용하였는지 보기 위하여 feature importance가 높은 feature들을 보았다.

| Class | 15 | 30 | 45 | 60 | 90 | 120 | 180 |
|---|---|---|---|---|---|---|---|
| MM | **(60.5)** | (481) | (38.9) | (26.2) | (20.1) | (22.5) | **(1.7)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| MA | 10.9 | (15.9) | (4.2) | (11.1) | (11.6) | 10.9 | (14.3) |
| SS | 32.2 | 22.2 | 0.0 | 10.0 | 6.3 | 6.6 | (5.3) |
| RS | 7.7 | 3.8 | 1.6 | 13.2 | 12.9 | 12.6 | (0.1) |
| PT | 44.6 | 49.4 | 33.5 | 40.4 | 31.8 | 38.4 | 34.5 |
| GT | 51.0 | 36.1 | 25.6 | 28.1 | 28.2 | 26.4 | 32.2 |
| RT | 35.5 | 42.0 | 31.1 | 32.3 | 44.0 | 26.2 | 44.1 |
| QT | 39.5 | 35.4 | 34.4 | 31.2 | 36.0 | 26.4 | 34.9 |

표 5. Feature selection에 따른 성과. 괄호 표시는 음수 값을 의미한다.

4.2.2 Feature importance

표 6과 표 7에서 class scaling method별로 feature importance가 높은 feature 및 feature scaling method를 보았다. 먼저 상대적으로 성과가 안 좋고, outlier들에 민감한 MM, MA, SS, 그리고 RS들의 결과는 표 6과 같다. Class의 scaling method가 바뀌면, feature importance가 높은 feature들이 많이 달라지는 것을 볼 수 있다. 제일 성과가 안 좋았던, MM 방법은 MM scaled feature들이 상위 feature들로 나타나고, 또 한 상대적으로 과거 데이터들의 중요도가 높았다. 이중 적은 수의 feature에서 성과가 제일 좋았던, SS 방법을 보면 RT scaled feature의 importance가 높은 것을 볼 수 있다. 그리고, 상대적으로 가까운 기간의 feature들의 importance가 높았다. 또 한, Tval(1)의 RT와 MM method의 importance가 모두 높고, Mom(2) 역시 RT와 MM method의 importance가 높았다. 즉, outlier의 영향을 무시하는 RT와 outlier를 포함하여 모든 분포를 고려하는 MM 모두 중요하게 작용하였다. 또 한, 다양한 조합을 통한 scaling method의 최적화의 타당성을 보여준다.

| | MM | MA | SS | RS |
|---|---|---|---|---|
| 1 | MM(DMom(12)) | MM(Mom(2)) | RT(Tval(1)) | MM(DMom(3)) |
| 2 | MM(DMom(9)) | MM(Mom(5)) | RT(Mom(1)) | RT(Dmom(3)) |
| 3 | MM(DMom(13)) | MM(Mom(1)) | RT(ibs(1)) | RT(Dmom(9)) |
| 4 | MM(DMom(2)) | MM(Mom(9)) | MM(Tval(1)) | RT(IBS(2)) |
| 5 | MM(DMom(3)) | MM(DMom(2)) | RT(Mom(2)) | RT(IBS(3)) |
| 6 | MM(Mom(1)) | MM(DMom(11)) | RT(DMom(1)) | RT(DMom(8)) |
| 7 | MM(Mom(6)) | MM(DMom(5)) | MM(Mom(2)) | RT(Tval(2)) |

| | | | | |
|---|---|---|---|---|
| 8 | MM(DMom(8)) | MM(DMom(10)) | RT(DMom(3)) | MM(Mom(2)) |
| 9 | MM(Mom(9)) | MM(DMom(14)) | MM(DMom(3)) | RT(Tval(1)) |
| 10 | MM(DMom(13)) | MM(DMom(12)) | MM(Mom(1)) | RT(DMom(1)) |

표 6. Class의 scaling method에 따른 Feature importance(1)

그리고 상대적으로 성과가 좋은 nonlinear scaling method인 PT, GT, RT, QT의 경우를 보았다. 각 Feature들의 RT와 MM 방법의 importance들이 모두 높은 것을 볼 수 있다. 그리고 이들만을 사용하여서 모델을 학습시켜도 기존과 성능이 비슷하거나 더 좋았다. 즉, Data scaling method들의 조합을 통한 최적화로 성과의 증가가 가능함을 보였다. 이는, data분포가 복잡한 여러 분야에서 적용이 가능할 것이다.

| | PT | GT | RT | QT |
|---|---|---|---|---|
| 1 | RT(Mom(2)) | RT(Mom(2)) | RT(Mom(2)) | RT(Mom(2)) |
| 2 | RT(Tval(1)) | RT(Mom(1)) | RT(Tval(1)) | RT(Tval(1)) |
| 3 | RT(Mom(1)) | RT(Tval(1)) | RT(Mom(1)) | RT(Mom(1)) |
| 4 | RT(IBS(1)) | RT(IBS(1)) | RT(IBS(1)) | RT(IBS(1)) |
| 5 | MM(Tval(1)) | MM(Mom(1)) | MM(Tval(1)) | MM(Tval(1)) |
| 6 | MM(Mom(1)) | MM(Mom(2)) | MM(Tval(4)) | MM(Tval(4)) |
| 7 | MM(Mom(2)) | MM(Tval(1)) | RT(Mom(3)) | RT(DMom(1)) |
| 8 | RT(Tval(2)) | RT(Tval(2)) | RT(DMom(1)) | RT(Tval(2)) |
| 9 | RT(Mom(3)) | RT(DMom(2)) | RT(Tval(2)) | MM(Tval(14)) |
| 10 | MM(Tval(4)) | RT(DMom(1)) | MM(Tval(14)) | RT(Mom(3)) |

표 7. Class의 scaling method에 따른 Feature importance(2)

## 제 3 절 연도별 성과 비교

고정된 train/valid/test 방법은 여러 연구분야에서 많이 사용된다. 시계열 데이터에서는 전체 데이터를 고정해서 분석할 경우 특정 시점만 분석을 하게 된다. 여러 기간에 대하여 성과를 보기 위해서는 rolling training method를 사용할 수 있다.

4.3.1 연도별 data scaling 방법의 영향

연도별 성과를 구하기 위하여 다음과 같이 실험을 진행하였다. 2018년 까지의 data를 85%의 training 기간, 15%의 valid 기간을 설정하고, 2019년의 예측을 하였다. 그 후, 매 1년마다 누적 training 학습을 하면서 test를 하였다. 즉 총 test기간은 2019~2022년이 된다. Test 기간동안 암호화폐 시장의 수익은 다음과 같다. 2019년 동안 평균 −25.5%의 수익률을 보여주었고, 2020년에는 234.7%, 2021년에는 977.0% 그리고 2022년에는 −68.7%의 수익률 보여주며, 전체기간에 106.6%의 CAGR을 보여주었다. 표8은 RF를 사용한 연도별 data scaling method의 영향에 대한 결과이다. 전체 기간에서도 GT/RT/QT 등의 nonlinear scaling method들의 성과가 좋은 것을 볼 수 있었다. 특히, 이들은 시장이 하락하는 시기인 2022년과 2019년 모두에서 좋은 성과를 보여주었다. 이를 통하여서 특정 기간뿐만 아니라 모든 기간에서 cross sectional return 분석을 할 때 data scaling 방법의 최적화가 중요하다는 것을 볼 수 있었다.

| Method | Whole | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|
| Market | 106.6 | (25.5) | 234.7 | 977.0 | (68.7) |
| MM | 391.8 | 52.2 | 1511.2 | 2161.4 | (69.4) |
| MA | 305.8 | 52.7 | 579.2 | 2432.6 | (63.3) |
| SS | 297.1 | (10.9) | 452.6 | 2955.9 | (17.8) |
| RS | 362.2 | (21.5) | 710.2 | 2945.6 | 36.2 |
| PT | 374.4 | 79.3 | 993.0 | 1370.6 | (18.1) |
| GT | 984.0 | 128.6 | 3587.0 | 4136.1 | 76.0 |
| RT | 787.4 | 108.4 | 2501.7 | 2970.1 | 89.9 |
| QT | 684.4 | 121.9 | 1839.3 | 2707.6 | 54.2 |

표 8. RF 연도별 Data scaling method에 따른 성과. 괄호 표시는 음수 값을 의미한다. 기간별 CAGR(%)을 보여준다.

그림 8은 일반적인 factor들 중 성과가 좋았던 경우들을 선택하여 성과를 비교하였다. Nonlinear scaling method를 사용하는 경우 단순 factor들에 비하여 성과가 좋음을 볼 수 있다. 특히, Market 평균이 상승할 때뿐만 아니라, 하락할 때도, 큰 하락 없는 모습을 보여준다. 이를 통해서 상승기와 하락기를 포함한 여러 특성을 갖는 기간에서 모두 machine learning이 암호화폐 시장의 cross sectional return 분석의 성과를 향상시켰고, 이때 nonlinear data scaling 방법의 적용이 좋음을

보였다.

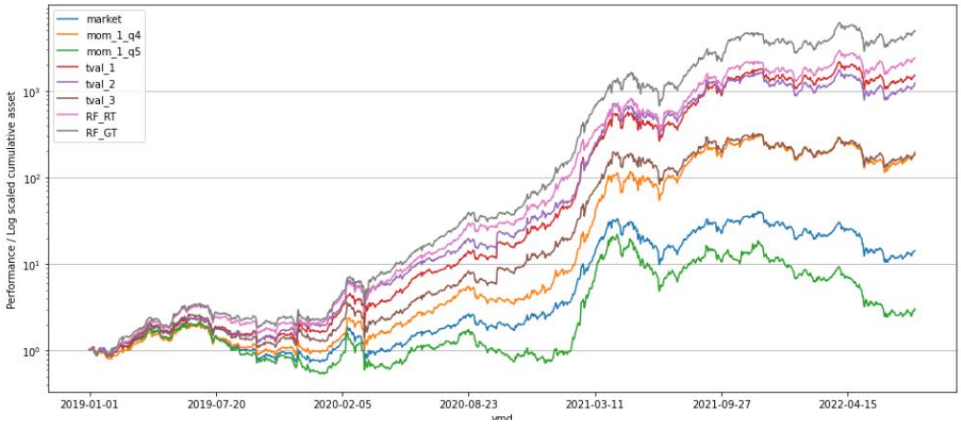


그림 8. 일반적인 factor들과 성과 비교

### 4.3.2 연도별 Feature 와 class의 data scaling 방법 grid search

4.1.2에서 grid search를 통한 feature와 class의 data scaling 방법 최적화를 통하여 성과가 향상되었다. 이런 성과의 향상이 데이터 분포가 다른 기간에서도 발생하는지 연도별 grid search를 통한 최적화를 실험하였다. 표 9은 그 성과를 보여준다. 금융 데이터는 기간에 민감하기 때문에, 기간에 따라 그 성과가 매우 상이하다. Feature의 scaling method가 MM인 경우에도, class의 scaling method가 nonlinear scaling method이면 성과가 좋은 것을 볼 수 있다. 허나, Feature가 GT나 RT를 사용했을 때와 비교하면, 상승기인 2021년을 제외하면 연도별 성과가 떨어지는 편이다. Class의 data scaling 방법을 outlier에 민감한 MM, SS를 사용하는 경우 시장의 하락기인 2019년과 2022년에서는 성과가 좋지 않았다. 하지만, 시장의 상승기인 2021년에는 그 성과가 좋다. 반면에, class의 data scaling 방법을 outlier에 민감하지 않은 PT, RT, GT로 사용하는 경우 모든 연도에서 성과가 좋은 것을 볼 수 있다. 즉 데이터의 분포에 민감한 MM, SS의 방식이 특정기간에 더 좋은 성능을 보일 수는 있으나 장기간의 안정성을 고려하면 nonlinear scaling method를 적용하는 것이 좋을 것이다. 그리고 성격이 다른 여러 기간에서도 data scaling method의 최적화를 통하여서 더 높은 성과를 얻을 수 있었다.

| Feature | Class | Whole | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|
| Market | | 106.6 | (25.5) | 234.7 | 977.0 | (68.7) |

| | | | | | | |
|---|---|---|---|---|---|---|
| MM | MM | **391.8** | **52.2** | **1511.2** | **2161.4** | **(69.4)** |
| | SS | 742.0 | 70.9 | 1159.0 | 10136.3 | (13.4) |
| | RS | 479.1 | 27.3 | 854.5 | 5501.5 | (35.8) |
| | PT | **985.7** | **128.6** | **1815.2** | **8667.8** | **60.6** |
| | GT | **854.7** | **135.7** | **1417.2** | **7350.9** | **34.3** |
| | RT | 884.6 | 120.7 | 1738.1 | 6408.9 | 66.3 |
| | QT | 801.0 | 115.9 | 1648.2 | 5409.0 | 46.1 |
| GT | MM | 925.9 | 137.3 | 2162.3 | 6524.5 | 25.6 |
| | SS | **1000.3** | **123.5** | **3599.3** | **5073.7** | **39.3** |
| | RS | 596.2 | 33.3 | 1713.9 | 3895.7 | 5.6 |
| | PT | **912.5** | **186.1** | **2154.5** | **4259.8** | **73.1** |
| | GT | **984.0** | **128.6** | **3587.0** | **4136.1** | **76.0** |
| | RT | 767.1 | 108.4 | 2091.8 | 3388.9 | 75.3 |
| | QT | 726.7 | 112.9 | 2108.7 | 2778.1 | 73.8 |
| RT | MM | 778.3 | 57.7 | 2010.7 | 5598.5 | 40.2 |
| | SS | 699.3 | 31.3 | 2289.2 | 5249.3 | (6.7) |
| | RS | 320.0 | (42.9) | 1138.7 | 2320.5 | (7.9) |
| | PT | **1079.5** | **201.9** | **2295.7** | **6290.8** | **89.6** |
| | GT | **921.8** | **172.7** | **2332.0** | **4155.3** | **83.3** |
| | RT | **787.4** | **108.4** | **2501.7** | **2970.1** | **89.9** |
| | QT | 746.1 | 118.3 | 2256.8 | 2752.8 | 74.9 |

표 9. RF의 연도별 Data scaling method 최적화. 기간별 CAGR(%)을 보여준다. 괄호 표시는 음수 값을 의미한다.

4.3.3 연도별 Data scaling method 조합을 통한 최적화

4.1.3에서 Feature에 data scaling method를 조합하여 최적화를 하였을 때 성과가 향상되었다. 이런 성과의 향상이 성격이 다른 여러 시기에서도 발생하는지 실험하였다. 표 10은 RF를 사용하였을 때 그 성과를 보여준다. 4.1.3와 마찬가지로 서로 성격이 다른 feature scaling method를 조합하였을 때, 성과의 향상을 보여준다. 특히 PT, GT, RT 및 QT 등의 nonlinear scaling method의 성과 증가가 컸다. 또 한, 기간별로 더 좋은 feature scaling method가 있는 것을 알 수 있다. 시장이 급등하는 기간에서는 MM, SS와 같은 linear scaling method의

성과가 좋고, 다른 기간에서는 nonlinear scaling method의 성과가 더 좋다.

| Feature | Class | Whole | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|
| Market | | 106.6 | (25.5) | 234.7 | 977.0 | (68.7) |
| MM | MM | 391.8 | 52.2 | 1511.2 | 2161.4 | (69.4) |
| RT | RT | 787.4 | 108.4 | 2501.7 | 2970.1 | 89.9 |
| MM/RT | MM | 464.6 | 34.5 | 1604.5 | 3748.7 | (64.4) |
| | SS | 623.2 | (16.7) | 2376.2 | 4920.6 | 17.7 |
| | RS | 272.2 | (44.1) | 786.9 | 2156.6 | (9.9) |
| | PT | 1135.5 | 186.4 | 2510.0 | 6907.9 | 103.0 |
| | GT | 970.3 | 141.5 | 2433.1 | 5382.2 | 80.9 |
| | RT | 983.4 | 165.2 | 2451.2 | 4774.4 | 100.1 |
| | QT | 910.1 | 159.7 | 2192.3 | 4846.8 | 58.0 |

표 10. RF의 연도별 Feature scaling method 최적화. 기간별 CAGR(%)을 보여준다. 괄호 표시는 음수 값을 의미한다.

표 11은 DNN을 사용하였을 때 그 성과를 보여준다. 4.1.3와 마찬가지로 서로 성격이 다른 feature scaling method를 조합하여 최적화하였을 때, 성과의 향상을 보여준다. 그리고 앞의 결과들과 비슷하게 nonlinear scaling method를 class에 적용하였을 때, 연도별 성과가 안정적이다. 이런 data scaling method의 조합을 통한 최적화 방법을 사용하면 기간에 민감한 금융데이터의 특성도 보완하며, 안정적인 성과를 얻을 수 있을 것이다.

| Feature | Class | Whole | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|
| Market | | 106.6 | (25.5) | 234.7 | 977.0 | (68.7) |
| MM | MM | 163.6 | 64.5 | 281.1 | 737.5 | (58.2) |
| RT | RT | 570.6 | 98.4 | 1225.6 | 2936.3 | 52.9 |
| MM/RT | MM | 402.9 | 44.2 | 674.2 | 4243.6 | (51.7) |
| | RT | 727.5 | 105.1 | 1456.5 | 4344.0 | 60.1 |
| | GT | 811.6 | 183.3 | 1402.2 | 4521.9 | 66.7 |
| MM/RT/ QT | MM | 711.6 | 119.0 | 1488.3 | 5044.8 | (0.9) |
| | RT | 782.4 | 125.6 | 1667.3 | 3917.0 | 94.5 |
| | GT | 810.0 | 183.6 | 1489.9 | 3712.0 | 109.6 |

표 11. DNN의 연도별 Feature scaling method 최적화. 기간별

CAGR(%)을 보여준다. 괄호 표시는 음수 값을 의미한다

# 제 5 장 결론 및 토의

본 연구에서는 암호화폐 시장에서의 cross sectional return 분석을 할 때, data scaling 방법 최적화의 중요성과 새로운 활용 방법들에 대하여 제안을 하였다. 금융 데이터는 그 특성상 noise가 많고 outlier들이 많다. 특히, 암호화폐 시장은 그 변동이 더 크다. 이는 기간에 따라 data의 분포 변화가 크다는 의미이고, 이런 관점에서 data scaling을 할 때, outlier들에 영향을 크게 받는 MM, MA 등의 scaling method들과 outlier들에 의한 영향이 적은 nonlinear scaling method의 성과가 다를 것이다. 실험 결과, outlier에 영향을 가장 많이 받는 MM, MA의 성과가 좋지 않았다. 그리고 대부분의 모델들에서 outlier에 영향을 받지 않는 nonlinear scaling method인 GT, RT, QT의 성과가 좋았다. 이를 통하여 금융 데이터의 outlier들이 학습에 방해가 되었다 볼 수 있다.

Grid search를 통한 Feature와 class의 data scaling 방법의 최적화를 통하여 서로 다른 scaling method들을 적용하였을 때, 향상된 성과를 얻을 수 있었다. 이를 통하여서 feature들의 scaling 후 데이터 분포에 따라서 더 좋은 class의 scaling 분포가 있는 것을 볼 수 있었다. 추가적인 최적화를 위하여 range가 0~1로 같은 MM, RT, QT를 조합하여 feature로 사용해 보았다. MM만 사용하는 경우 대부분의 모델에서 성과가 매우 안 좋았지만, 이들을 다른 feature들과 조합하여 사용하는 경우, 성과가 좋았던 RT를 단독으로 사용할 때보다 성과가 좋았다. Outlier를 무시하고 순서만을 나타내는 RT와 전체 data distribution에 대하여 알 수 있는 MM이 서로 시너지 효과를 발휘했다 생각하였다. 이런 부분을 밝히기 위하여, 각 class의 scaling method들에 대하여 feature scaling method들의 feature importance를 측정하였다. 성과가 안 좋은 경우들에서는 특정 data scaling method를 사용한 feature들만 사용하였다. 하지만 성과가 좋았던 경우들에서는 여러 data scaling method들을 함께 사용하며, 이들의 synergy 효과를 볼 수 있었다. 즉, data의 RT와 MM scaling후의 feature들을 모두 중요하게 생각하였다. 또 한, 이런 방법을 통하여서 모델을 학습하기 전 각 feature에 맞는 scaling method로 학습을 진행할 수 있을 것이다.

그리고 성격이 다른 여러 기간에 대하여 실험을 진행하여, 기간에 민감한 금융데이터에서도 안정적인 성과를 얻을 수 있었다. 이를 통하여서 feature와 class에 한가지의 scaling method를 사용하는 것이 아닌, grid search나 조합 최적화를 통하여 데이터 분포를 보완해주는 방법들을 제안하였다.

본 연구를 통해서 data scaling method의 최적화를 통하여 자산 시장의 cross sectional return 분석을 할 때 성과의 향상을 얻을 수 있었다. 하지만, 아직 다양한 방면에서의 연구가 부족하다. 먼저 암호화폐는 여러 국가에 여러 마켓이 있으며, 각 마켓마다 상장된 코인들의 종류가 다르다. 이런 여러 마켓에 상장된 다양한 코인들에 대하여서도 실험이 필요할 것이다. 또 한, 본 연구에서는 예측 기간을 1일로 하여 실험을 진행하였는데, 예측 기간에 따라서 class data분포 역시 달라질 것이기 때문에 scaling method의 영향 또한 변할 것이다. 이런 경우에도 앞에서 말한 feature importance와 feature들의 상태를 보면서 좋은 모델들을 선택할 수 있을지 확인해 보고 싶다.

# 참고 문헌

[1] Hamilton, J. D. 1994. Time series analysis. Princeton, NJ: Princeton university press

[2] Bollerslev, T. 1986. Generalized autoregressive conditional heteroskedasticity. Journal of econometrics

[3] Engle, R. F., Ledoit, O., and Wolf, M. 2019. Large dynamic covariance matrices. Journal of Business & Economic Statistics

[4] JKS, Liew. Boris, M. 2017. Forecasting ETFs with Machine Learning Algorithms. The Journal of Alternative Investments

[5] Fama, E. F., and French, K. R. 1992. The cross-section of expected stock returns. the Journal of Finance

[6] Fama, E. F., and French, K. R. 1993. Common risk factors in the returns on stocks and bonds. Journal of financial economics

[7] Abe, M., and Nakayama, H. 2018. Deep learning for forecasting stock returns in the cross-section. Pacific-Asia Conference on Knowledge Discovery and Data Mining

[8] Kei, N. 2020. Cross-sectional Stock Price Prediction using Deep Learning for Actual Investment Management. 2020 Asia Service Sciences and Software Engineering Conference

[9] Shihao, G., and Bryan, K. 2020. Empirical Asset Pricing via Machine Learning. The Review of Financial Studies

[10] Thara, D., and Fan, X. 2019. Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. Pattern Recognition Letters

[11] Abhilash, S. and Cheng-Chi, L. 2022. LT-FS-ID Log-Transformed Feature Learning and Feature-Scaling-Based Machine Learning Algorithms to Predict thek-Barriers for Intrusion Detection Using Wireless Sensor Network. Sensors

[12] MD, M. J., and Pritom, K. S. 2021. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. Technologies

[13] Abu Bakar, N. and Rosbi, S. 2017. Autoregressive integrated moving average (ARIMA) model for forecasting cryptocurrency

exchange rate in high volatility environment: A new insight of bitcoin transaction. Int. J. Adv. Eng. Res. Sci
[14] Hamayel, M.J. and Owda, A.Y. 2021. A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. AI
[15] Pichl, L. and Kaizoji, T. 2017. Volatility analysis of bitcoin. Quant. Financ. Econ
[16] JKS L., and Richard Z. L. 2019. Cryptocurrency Investing Examined. The JBBA
[17] Yukun, L., and Aleh, L. 2022. Common Risk Factors in Cryptocurrency. The Journal of Finance
[18] Cramer,. 1976. Estimation of toxic hazard—A decision tree approach. Food and Cosmetics Toxicology
[19] Haykin, SS. 2009. Neural Networks and Learning Machines. New Jersey: Pearson Education Upper Saddle River.
[20] ML, d. F. 2018. The 10 reasons most machine learning funds fail
[21] Ronen, I.,Tobias J. M., 2012,The role of shorting, firm size, and time on market anomalies
[22] Matteo, R., Ardi G.,2018, Efficient Market Hypothesis And Stock Market Anomalies: Empirical Evidence In Four European Countries
[23] Kai-Yin W., Chulin M., Michael M., 2020, Review on Efficiency and Anomalies in Stock Markets

Abstract

# Empirical study on the effect of Nonlinear Data scaling methods in Cross sectional return analysis of Cryptocurrency Markets

Hyoungsoo Kim
Computer Science
The Graduate School
Seoul National University

Various methods of predicting asset prices have been studied. Initially, linear predictive models were used. Due to the complexity of asset data, nonlinear algorithms were applied. After that, machine learning algorithms have received a lot of attention by reducing the noise of the data and relatively increasing the prediction accuracy of asset prices. Recently, due to the increase in computational power, studies on methods using deep learning models are being actively conducted. Among the problems of predicting asset prices, the analysis of the relative returns of various stocks is called cross sectional return analysis. In these studies, it is assumed that asset prices are determined by a specific factor. It can be seen as a good factor if the factor distinguishes between stocks that are relatively rising and stocks that are falling. In cross sectional return analysis, several factors are claimed to be significant after Fama-French's 3-factor model. Recently, machine learning and deep learning are trying to explain this. However, the group of stocks and countries

that make up the asset market are diverse and are greatly affected by the period. When the conditions change due to these factors, the performance of the model also changes significantly.

Financial data is noisy and has many outliers, but there is a lack of research on data scaling methods in the data pre-processing stage considering them. In this study, we studied the influence of data scaling methods in the cross-sectional return analysis of the cryptocurrency market, which has higher volatility than the general asset market, and showed their importance. And, good data scaling methods were seen in common in many models. In addition, in previous studies, only limited data scaling methods were applied, but through the optimization of data scaling methods, various data scaling methods were applied to features and classes to show an increase in performance. In addition, the validity of these optimization methods was shown through feature selection. In addition, since the experiment may be focused on a specific period, in order to prevent this, the experiment was conducted for several periods using the rolling back testing method. Through this, it was possible to obtain good performance in various periods with the optimization methods suggested in the study, and a method to obtain stable performance in financial data sensitive to period was proposed.

**Keyword** : Cryptocurrency, Cross sectional return analysis, Data scaling method
**Student Number** : 2019-28216