2021학년도 2학기

통계분석실습 중간프로젝트 보고서

- KOSPI 기업의 수익률 예측 -

2조 최윤서(1916129, 통계학과)

2021.11

Ⅰ. 서론

1. 분석 프로젝트 개요

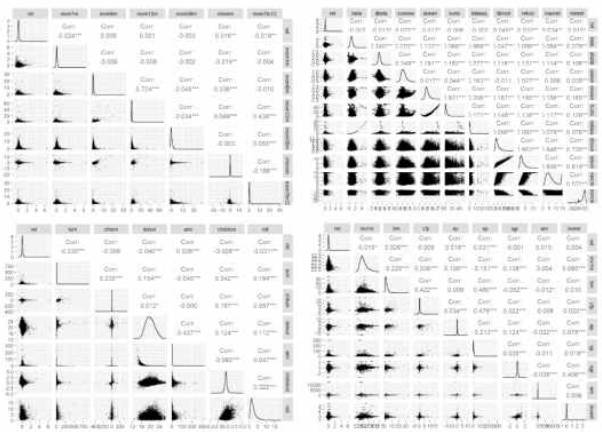
- 데이터 : 한국거래소(KRX)에 있는 KOSPI 기업의 월별 관측값
- 관측기간: 1987년 2월부터 2018년 12월까지
- 설명변수 X: 총 73개의 설명변수 (수익률 관측 시점으로부터 한 달 전까지의 정보)
- 반응변수 Y: 수익률(simple monthly return)
- 예측기간: 2007년 1월부터 2018년 12월까지의 월별수익률 (IMF 구제금융 이후의 상황)
- 평가방법: sharpe ratio

  - 각 월의 예측수익률을 기반으로 기업을 그룹화하고 상위그룹과 하위그룹을 맞교환하여 얻어진 수익률에 대한 sharpe ratio값을 계산

2. 데이터 특성

1) 데이터 탐색

분석 과정에서 특별히 고려해야 할 변수의 존재 유무를 알기 위해 반응변수 y(각 그래프의 1행 1열)와 설명변수 X의 산점도와 상관계수를 살펴보았다.



산점도와 상관계수를 살펴보았을 때 눈에 띄게 선형관계를 가진 변수를 찾기가 어렵고, 선형관계 외에도 특별히 y에 큰 영향을 주는 변수를 찾기 어려웠다.

다중공선성 문제가 심각해 보이지는 않지만, 그래도 일부 변수들 사이에 선형관계가 존재하기 때문에 주성분분석을 이용한 설명변수 변환 후 모델 적합도 고려해볼 수 있었다.

2) 주식데이터 특성

주식데이터는 시간의 흐름 순으로 작성된 데이터이기 때문에 각 시기별로 가진 특성이 다를 것으로 예상되었다. 따라서 특정 시점마다 다른 모델을 선택하여 모델을 적합하고자 하였다. 또한, 자료의 설명변수는 이전 시점의 정보를 담고 있기 때문에 훈련데이터와 검증데이터를 나눌 때 랜덤으로 나눈다면 두 데이터셋의 독립성이 보장되지 않을 것이라 생각하여 비율을 4:1로 앞 뒤로 분할하였다.

Ⅱ. 최종 모델 선정 과정

최종 모델을 선정하기까지의 프로젝트 진행 방향을 간략히 설명하자면, 우선 모든 기간에 대하여 단일 모델을 적합하여 Sharpe ratio를 측정한다. 그 중 sharpe ratio가 높게 나온 몇 개의 모델을 모델 후보로 선택하고, 기간에 따라 나누어진 시기별로 특정 성능지표를 이용하여 모델을 선택한 뒤, 선택된 모델로 월별 수익률을 예측하는 과정을 거쳤다. 그리고 가장 높은 sharpe ratio를 결과로 내는 조건을 최종 프로젝트 결과물로 선정하였다.

1. 전체 시기에 대해 단일 모델 적합

오른쪽의 그림처럼 훈련데이터 5년, 테스트데이터 1년으로 기간을 분할하고, 전체 시기에서 Linear regression, linear median regression, Random Forest, Xgboost, Ridge, Lasso 모형을 적합하였다. 이때 Random Forest는 R의 ranger 패키지의 default값을 이용하여 모델을 적합하였고, Xgboost와, Ridge, Lasso의 경우 훈련데이터를 다시 4년, 1년, 각각 앞, 뒤로 훈련데이터와 검증데이터로 분할하여 하이퍼파라미터 튜닝을 거쳤다. Xgboost의 경우 caret 패키지를 사용하였고, Ridge와 lasso의 경우 glmnet에서 기본으로 주어지는 lambda값 100개 중 검증데이터에서 최소의 mse를 가지는 lambda값을 사용하였다.

우선 예측 결과를 살펴보면 Random Forest, Xgboost, Ridge 모형이 기준점인 linear regression의 0.45보다 더 높은 sharpe ratio값을 가졌다.

lasso의 경우 73개의 설명변수 중 불필요한 정보를 담은 변수의 유무를 알아보고자 적합해 보았는데 성능이 좋지 않아 모델 선택 후보에서 제외하였다.

<단일 모형으로 적합 후 예측한 결과>

	Avg	Std	SR
Linear regression	0.87	6.73	0.45
median linear regression	0.92	7.95	0.4
Random Forest	1.31	7.31	0.62
Xgboost	1.52	7.66	0.69
Ridge	0.91	6.92	0.46
Lasso	0.7	6.65	0.36

median linear regression은 sharpe ratio값은 낮지만 평균 수익률이 linear regression을 적합했을 때의 0.87보다 더 높게 나왔기 때문에 우선 모델 후보로 선택하기로 결정하였다. Random Forest의 경우 다른 모델에 비해 훈련데이터와 검증데이터에서의 mse가 과하게 차이가 나서, 과적합이 발생하는 모델이라 생각하여 모델 후보에서 제외하였다. 각 모델의 훈련데이터와 검증데이터의 mse비교값은 하단에 첨부하였다.

	random forest		linear regression		median linear regression		xgboost		ridge	
year	train	val	train	val	train	val	train	val	train	val
2007	0.00672	0.0355	0.03410	0.02217	0.03593	0.02097	0.03456	0.02101	0.03563	0.02049
2008	0.00686	0.0391	0.03208	0.03543	0.03304	0.03656	0.03160	0.03506	0.03230	0.035149
2009	0.00708	0.0359	0.0352	0.03665	0.03638	0.03696	0.03463	0.03663	0.03519	0.03409
2010	0.00708	0.0249	0.03598	0.03529	0.03680	0.03698	0.03486	0.03526	0.03701	0.02365
2011	0.00571	0.0242	0.03056	0.02451	0.03130	0.02386	0.02795	0.02490	0.03112	0.02317
2012	0.00575	0.0256	0.03134	0.02463	0.03208	0.02349	0.02875	0.02496	0.03222	0.02526
2013	0.00544	0.0163	0.02832	0.02630	0.02890	0.02951	0.02559	0.02532	0.02903	0.01598
2014	0.00494	0.0142	0.02617	0.01837	0.02688	0.02069	0.02516	0.01591	0.02659	0.01397
2015	0.00419	0.0255	0.02166	0.01442	0.02208	0.01457	0.02120	0.01400	0.02182	0.02525
2016	0.00132	0.0168	0.01921	0.02555	0.01957	0.02650	0.01884	0.02871	0.01940	0.01617
2017	0.00399	0.0144	0.01981	0.01618	0.02032	0.01620	0.01958	0.01611	0.01988	0.01376
2018	0.00347	0.0247	0.01758	0.01398	0.01803	0.01387	0.0174	0.01390	0.01781	0.03661

〈각 모델의 훈련데이터와 검증데이터에서의 mse〉

따라서 6개의 모델 중 Lasso와 Random Forest를 제외한 4개의 모델 linear regression, linear median regression, Xgboost, Ridge를 모델 후보로 선택하여 각 시기별로 모델을 선택하여 월별 수익률을 예측해보고자 하였다.

2. 각 시기별로 모델을 선택 후 적합 (test-set기간: 1년, 모델 후보 4개)

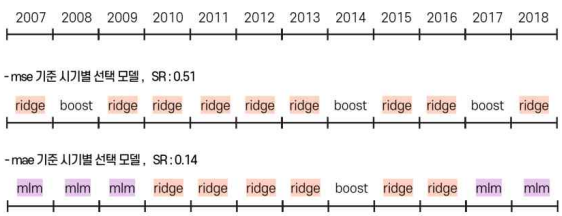
- 모델 후보: linear regression, median linear regression, ridge, xgboost

<결과값>

	Avg	Std	SR
모델선택_mse 기준	1	6.77	0.51
모델선택_mae 기준	0.32	7.78	0.14
단일모델_Xgboost	1.52	7.66	0.69

아래의 그림처럼 훈련데이터 4년, 검증데이터 1년, 테스트데이터 1년으로 데이터를 분할하고, time window rolling 과정을 거치며, 각 시기별로 선택된 후보 4개 중 가장 작은 mse와 mae를 가지는 모델로 해당 시기의 수익률을 예측하였다.

시기별로 다른 모델을 선택하여 적합하면 성능이 좋아질 것이라고 예상했던 바와는 달리 Xgboost 단일 모델만으로 전체 시기를 예측했을 때보다 시기별로 다른 모델을 선택하여 예측하였을 때 sharpe ratio가 더 낮아졌다.



어떤 지점에서 성능이 낮아졌는지를 확인하기 위해 각 시기별로 어떤 모델이 선택되었는지 살펴보았고, 결과는 오른쪽과 같았다. mse를 기준으로 모델을 선택하였을 때는 Ridge 모델이 많이 선택되었고, mae를 기준으로 모델을 선택하였을 때는 linear median regression 모델이 많이 선택되었다.

Ridge 모델의 경우 검증데이터에서 가장 낮은 mse값을 가진 lambda를 이용하여 모델을 훈련시켰다. Ridge 모델의 경우 편향이 크고 분산이 작은 모델로 이러한 성질 때문에 검증데이터에서는 좋은 성능을 보였지만, 테스트데이터에서는 낮은 성능을 보인 것이 아닌가 판단이 들어 최종 모델 후보에서 제외하였다.

linear median regression의 경우 성능지표로 mae를 사용했을 때만 선택된 것을 확인할 수 있다. Sharpe ratio의 값을 높이기 위해 잘 예측해야 하는 데이터는 수익률이 중간값을 가지는 구간이 아니라 수익률이 높거나, 낮은 값을 가지는 구간이다. linear median regression의 경우 매우 높거나 낮은 값의 오차에 덜 민감한 모형이기 때문에 수익률이 높거나, 낮은 구간을 잘 예

측하지 못해서 이런 결과가 발생했다는 생각이 들어 최종 모델 후보에서 제외하였다.

따라서 최종 모델 후보를 Ridge와 linear median regression을 제외한 linear regression과 Xgboost 모델 2개로 축소시켰다.

3. 각 시기별로 모델을 선택 후 적합 (test-set기간: 1년, 모델 후보 2개)

- **모델 후보: linear regression, xgboost**

linear regression과 Xgboost를 모델 후보로 각 시기별로 모델을 선택하여 수익률을 예측한 뒤 계산한 sharpe ratio값은 오른쪽 표와 같다. 성능지표로 mae를 사용하여 모델을 선택했을 때, Xgboost 단일 모델을 이용했을 때보다 더 높은 Sharpe ratio 값을 가지는 것을 알 수 있다.

	Avg	Std	SR
모델선택(후보4개)_mse 기준	1	6.77	0.51
모델선택(후보4개)_mae 기준	0.32	7.78	0.14
모델선택(후보2개)_mse 기준	1.02	7.47	0.47
모델선택(후보2개)_mae 기준	1.64	7.7	0.74
단일모델_Xgboost	1.52	7.66	0.69

4. 각 시기별로 모델을 선택 후 적합 (test-set기간: 6개월, 모델 후보 2개)

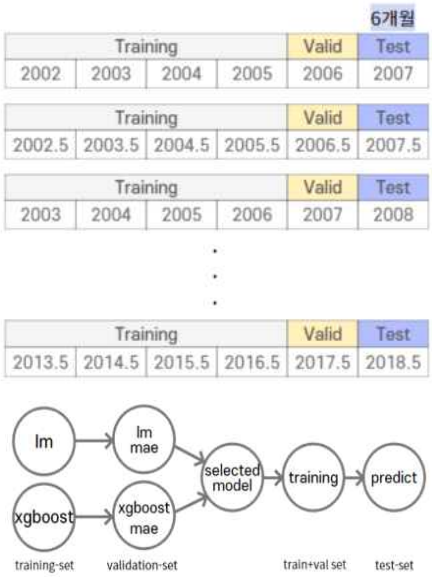
- **모델 후보: linear regression, xgboost**

3의 상황을 그대로 예측 시기를 6개월로 축소하여 같은 과정을 반복하였다. 그 결과 mae를 기준으로 모델을 선택했을 때 1.05로 가장 높은 sharpe ratio값을 얻을 수 있었고, 예측기간 6개월, 훈련기간 5년, 모델 후보 linear regression, xgboost, 그룹 수 10개일 때의 조건을 최종 모델로 선정하였다.

	Avg	Std	SR
모델선택(1년)_mse 기준	1.02	7.47	0.47
모델선택(1년)_mae 기준	1.64	7.7	0.74
모델선택(6개월)_mse 기준	1.9	7.37	0.89
모델선택(6개월)_mae 기준	2.29	7.57	1.05

5. 최종 모델

직전 5년치 데이터를 각각 4년, 1년씩 훈련데이터와 검증데이터로 분할하고, 이후 6개월을 테스트데이터로 수익률을 예측한다. 그리고 time window rolling과정을 거치며 2007년부터 2018년까지의 월별 수익률을 예측할 수 있도록 한다.



이렇게 분할된 예측 시기별로 훈련데이터에서 linear regression과 xgboost 모델을 훈련시킨다. xgboost의 경우 5-fold cross validation을 이용하여 하이퍼파라미터 튜닝 과정을 거치게 된다. 이렇게 훈련된 두 모델로 검증데이터에서 mae값을 계산하고 더 낮은 mae값을 가진 모델을 해당 시기의 최종 모형으로 선정한다. 최종 선정된 모형을 전체 훈련세트, 즉 훈련데이터와 검증데이터를 합친 데이터에 대해 다시 학습시킨다. 여기서 최종 선정된 모형이 xgboost라면, xgboost는 다시 한 번 하이퍼파라미터 튜닝 과정을 거치게 된다. 이렇게 최종적으로 학습된 모델을 이용하여 테스트 데이터를 예측하고, 이를 2007년에서 2018년까지 반복하였을 때 최종 sharpe ratio 1.05를 얻을 수 있었다.

Ⅲ. 데이터분석 프로젝트 자체 평가

해당 절에서는 본 프로젝트를 자체적으로 평가하고, 추후에 같은 분석을 하게 된다면 추가로 시도해보거나, 수정해보면 좋을 사항들을 작성하였다.

1. 설명변수 – 특성변환 추가

본 프로젝트에서는 주어진 변수를 변환없이 그대로 사용하였다. 분석 초반 데이터에 대한 이해가 부족하여 모델선택에만 초점을 맞춰 진행을 하다보니 설명변수에 담긴 정보를 많이 사용하지 못했다고 생각한다. 특히 ridge와 lasso같은 선형모델을 사용할 때 설명변수 X들의 제곱, 세제곱, 제곱근 등의 특성 변환한 변수들을 설명변수에 추가하고, 변수들간의 교호작용항을 추가

하여 ridge와 lasso를 적합해보면 단순 선형 모형에서는 찾아내지 못한 비선형관계나 교호작용이 모델에 반영되어 성능 향상을 기대해볼 수 있을 것 같다.

2. 모델을 선택할 때 사용하는 성능지표

시기별로 모델을 선택할 때 사용하는 성능지표로 일반적으로 모델 선택에 많이 사용되는 MSE와 MAE를 사용하였다. 하지만 해당 프로젝트의 목표는 수익률의 Sharpe ratio를 가장 높게 하는 모델을 찾아내는 것이다. 따라서 모델을 선택할 때 사용하는 성능지표로 validation-set의 Sharpe ratio를 사용했다면 시기별로 프로젝트의 목적에 더 적합한 모델을 선택할 수 있어 결과를 더 좋게 만들 수 있다고 생각한다.

3. Xgboost – 검증된 하이퍼파라미터 사용

최종 모델에서 특정 시기의 선택모델이 Xgboost일 때, 전체 훈련데이터(훈련데이터 + 검증데이터)에 대해 다시 한 번 5-fold CV과정을 거쳐 하이퍼파라미터 튜닝을 다시 진행하였다. 하지만 이미 validation-set을 이용하여 특정 하이퍼파라미터 조합이 linear model보다 좋은 성능을 낸다는 것이 검증된 상태에서 다시 한 번 하이퍼파라미터 튜닝과정을 거치게 된다면, 이전에 검증된 모델과는 다른 모델을 사용하는 것이 된다. 이 부분에서 "검증데이터를 이용하여 더 좋은 모델을 선택하자"는 분석 목적에 부합하지 않았다고 생각한다. 따라서 다시 프로젝트를 진행한다면 시기별 모델 비교에서 사용된 하이퍼파라미터 조합을 그대로 전체훈련데이터에 대해 훈련시킨 모델을 해당 시기의 최종 모델로 사용해야 한다.

4. 하이퍼파라미터 튜닝

프로젝트 진행 과정에서 모든 하이퍼파라미터 튜닝과정을 R패키지에 의존하여 진행하였다. 이 지점에서 데이터에 적절한 하이퍼파라미터 조합을 찾지 못해 성능 향상을 하지 못했다고 생각한다. 또한, 패키지에 의존하다보니 불필요한 모델 훈련 과정이 생겼고 그만큼 훈련시간이 오래 걸리는 문제가 발생했다. 데이터와 분석 목적에 적절한 파라미터 조합을 가지고 grid search를 진행한다면 성능 향상과 시간 단축의 이점을 얻을 수 있을 것이다.

5. 모델 후보의 부족, linear regression vs Xgboost

최종 모델에 사용하는 분석 모델의 후보는 linear regression과 Xgboost 단 2개라는 점에서 모델의 선택폭이 좁았다고 생각한다. 분석 초반 Random Forest를 과적합 문제로 모델 후보에서 제외하였는데, 하이퍼파라미터 조정을 통해 과적합 문제를 해결해보고, 만약 해결이 된다면 모델 선택 후보에 포함시켜도 좋을 것 같다.

	random forest		linear regression		median linear regression		xgboost		ridge	
year	train	val	train	val	train	val	train	val	train	val
2007	0.00672	0.0355	0.03410	0.02217	0.03593	0.02097	0.03456	0.02101	0.03563	0.02049
2008	0.00686	0.0391	0.03208	0.03543	0.03304	0.03656	0.03160	0.03506	0.03230	0.035149
2009	0.00708	0.0359	0.0352	0.03665	0.03638	0.03696	0.03463	0.03663	0.03519	0.03409
2010	0.00708	0.0249	0.03598	0.03529	0.03680	0.03698	0.03486	0.03526	0.03701	0.02365
2011	0.00571	0.0242	0.03056	0.02451	0.03130	0.02386	0.02795	0.02490	0.03112	0.02317
2012	0.00575	0.0256	0.03134	0.02463	0.03208	0.02349	0.02875	0.02496	0.03222	0.02526
2013	0.00544	0.0163	0.02832	0.02630	0.02890	0.02951	0.02559	0.02532	0.02903	0.01598
2014	0.00494	0.0142	0.02617	0.01837	0.02688	0.02069	0.02516	0.01591	0.02659	0.01397
2015	0.00419	0.0255	0.02166	0.01442	0.02208	0.01457	0.02120	0.01400	0.02182	0.02525
2016	0.00132	0.0168	0.01921	0.02555	0.01957	0.02650	0.01884	0.02871	0.01940	0.01617
2017	0.00399	0.0144	0.01981	0.01618	0.02032	0.01620	0.01958	0.01611	0.01988	0.01376
2018	0.00347	0.0247	0.01758	0.01398	0.01803	0.01387	0.0174	0.01390	0.01781	0.03661

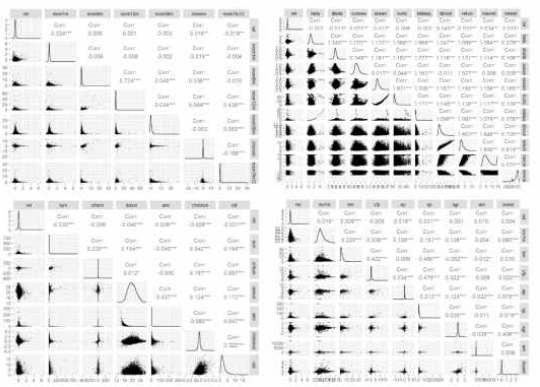
〈각 모델의 훈련데이터와 검증데이터에서의 mse〉

3페이지에서 살펴본 각 모델의 훈련데이터와 검증데이터에서의 mse값을 정리한 표를 보면, Xgboost와 linear regression이 비슷한 mse값을 가지거나 훈련데이터보다 검증데이터의 mse가 더 작은 시기를 찾아볼 수 있다. 이를 토대로 다른 시각으로 해당 표를 해석해보자면, Random Forest가 과적합된 것이 아니라 Xgboost와 linear regression이 과소적합된 것이라고 생각해볼 수 있었다. 따라서 **4**에서 언급한 것처럼 적절한 하이퍼파라미터 튜닝 과정을 거친 Xgboost와 Random Forest의 mse값을 비교하여 모델 과적합 여부를 판단했어야 한다고 생각한다.

Ⅳ. 그 외에 시도해 본 방법들

해당 절에서는 위에서 언급하지 않은 시도해봤지만 좋은 성능을 내지 못한 시도들을 정리해놓았다.

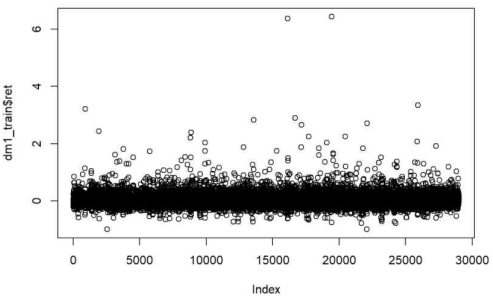
1. 주성분 분석



데이터 탐색단계에서 산점도와 상관계수를 살펴보았을 때 몇몇 설명변수들간에 선형관계가 존재하는 것을 확인할 수 있었다. 따라서 다중공선성 문제가 존재할 수 있다고 생각했고, 주성분분석을 이용하여 이를 해결해보고자 했다.

PCA를 진행한 뒤 누적설명력이 80~90%인 구간에서 변수를 선택하여 새로운 설명변수로 사용해보았지만 성능이 좋지않았다. 특정 변수가 큰 설명력을 가지지 않았고, 차원을 축소하는 과정에서 발생한 정보 손실이 발생하여 이런 결과가 나왔을 것으로 예상된다.

2. 이상치 제거



분석 초반 반응변수인 ret의 산점도를 살펴보았을 때, 주식데이터의 특성상 유독 높은 수익률을 나타내는 데이터가 존재했다. 따라서 이를 이상치로 간주하고 제거한 뒤에 모델 훈련을 진행했지만, 좋은 성능을 내지 못했다.

sharpe ratio를 올리기 위해서는 높은 수익률을 잘 예측하는 것이 중요한데, 높은 수익률을 가진 자료를 삭제함으로써 중요한 정보가 빠진 점에서 성능이 낮아졌다고 생각한다. 만약 이상치를 제거하고자 한다면 일반적인 X와 y의 관계와 다른 양상을 띄는 자료를 제거하는 것이 더 올바른 방향이였을 것이라고 생각한다.