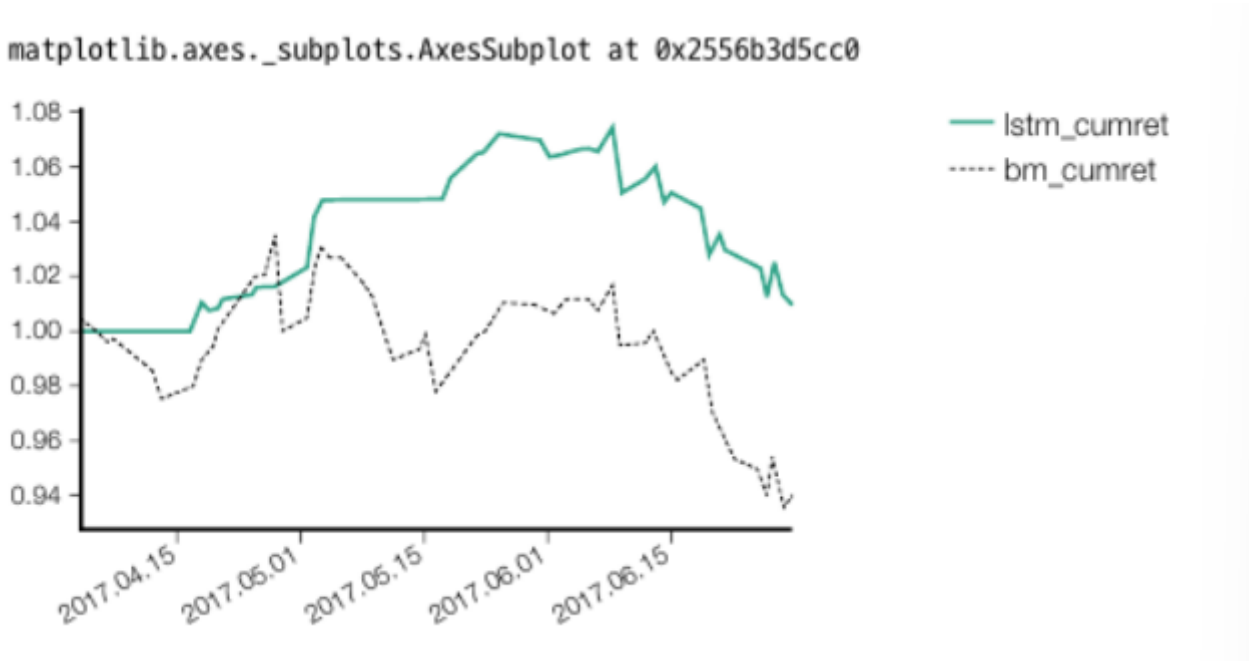
RNN을 활용한 주가 방향성 분류 예측 - 분석

작성일시	@2022년 8월 15일 오후 1:51
최종 편집일시	@2022년 8월 19일 오전 12:06
문서 유형	hai stock
레퍼런스	

결과 해석



전체적으로 하락장에서 단순 보유했다면 손실을 봤겠지만, 이 전략에서 나타난 시그널을 적용했다면 손실을 면할 수 있었다.

그럼 백테스팅 결과를 살펴보자. MDD 는 테스트 기간 주가에 대한 MDD 를 계산했고, CAGR, 변동성, 샤프 지수는 해당 기간의 성과에 연율화를 적용했으므로 다소 과장된 수치가 나올 수 있다.

```
"Buy and Hold"
CAGR :  -21.92 %
Sharpe :  -1.62
VOL :  14.82 %
MDD :  9.73 %
```

```
"LSTM st"
CAGR :  3.9 %
Sharpe :  0.46
VOL :  9.43 %
MDD :  9.73 %
```

기간이 너무 짧아서 연율화를 하게 되면 오히려 수익률을 부풀리는 효과가 있을 수 있다. CAGR, 샤프지수, 변동성은 모두 연율화했고 MDD는 종가에 대해 MDD를 구한다. LSTM 전략을 사용했을 때 연율화 시 3% 이상의 수익률이 계산된다. 전체적으로 단순 보유한 성과보다는 더 좋은 성과를 보인다.

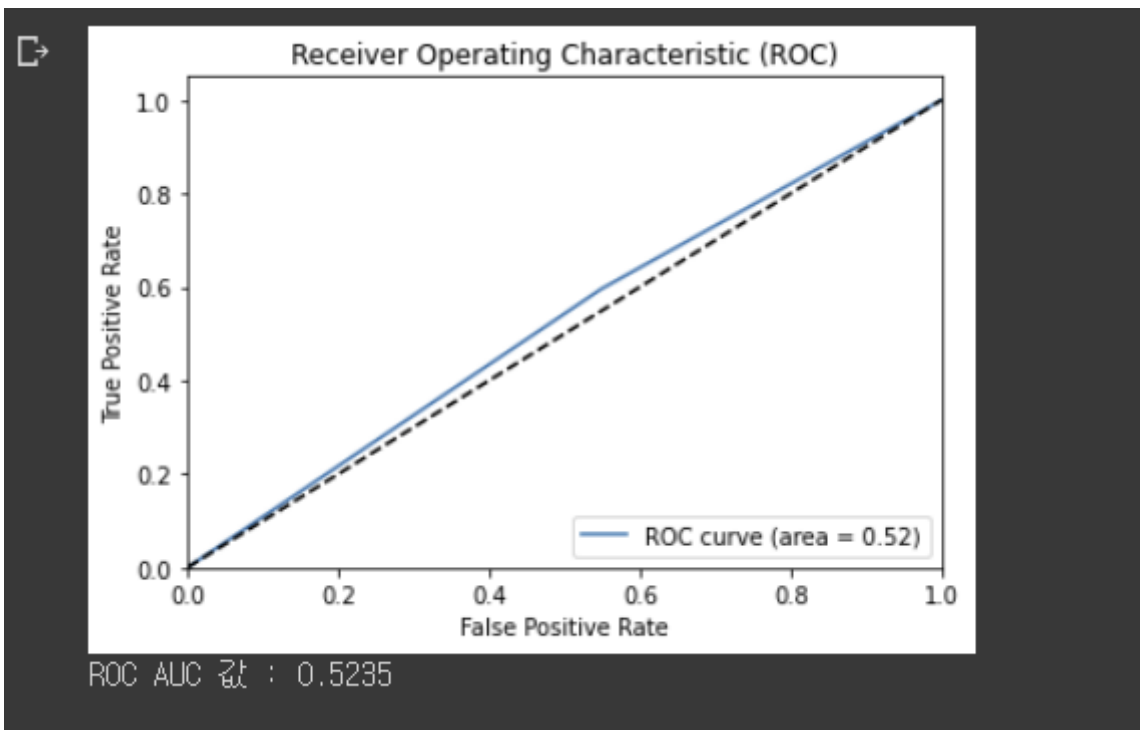
TQQQ 에 대해서

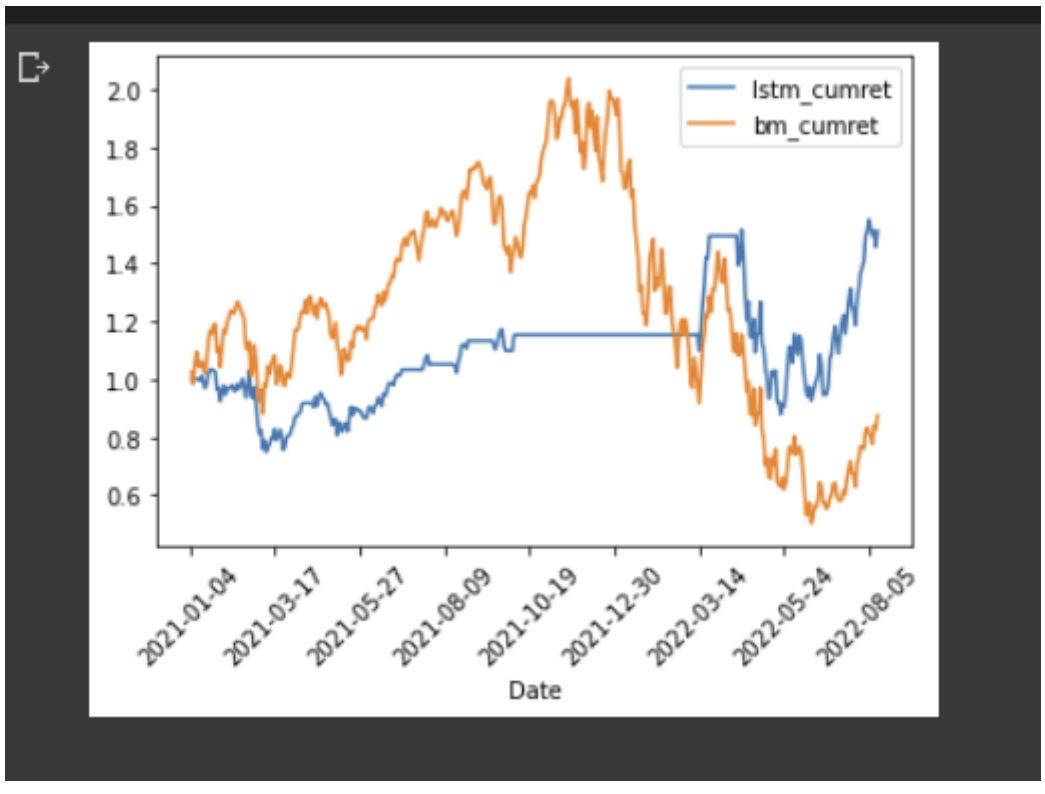
epochs = 50

```
=======
20epochs_10batch
TN: 84
FN: 87
TP: 128
FP: 102
TPR: 0.5953488372093023
FPR: 0.5483870967741935
accuracy: 0.529
specitivity: 0.452
sensitivity : 0.595
mcc : 0.047
              precision    recall  f1-score   support

           0       0.49      0.45      0.47       186
           1       0.56      0.60      0.58       215

    accuracy                           0.53       401
   macro avg       0.52      0.52      0.52       401
weighted avg       0.53      0.53      0.53       401
=======
```





바이앤홀드 전략 BM

```
[64]  1  CAGR = lstm_book_df.loc[lstm_book_df.index[-1],'bm_cumret'] ** (252./len(lstm_book_df.index)) -1
      2  Sharpe = np.mean(lstm_book_df['ret']) / np.std(lstm_book_df['ret']) * np.sqrt(252.)
      3  VOL = np.std(lstm_book_df['ret']) * np.sqrt(252.)
      4  MDD = historical_dd.min()
      5  print('CAGR : ',round(CAGR*100,2),'%')
      6  print('Sharpe : ',round(Sharpe,2))
      7  print('VOL : ',round(VOL*100,2),'%')
      8  print('MDD : ',round(-1*MDD*100,2),'%')
```

```
CAGR :  -8.03 %
Sharpe :  0.26
VOL :  74.23 %
MDD :  75.32 %
```

LSTM 전략

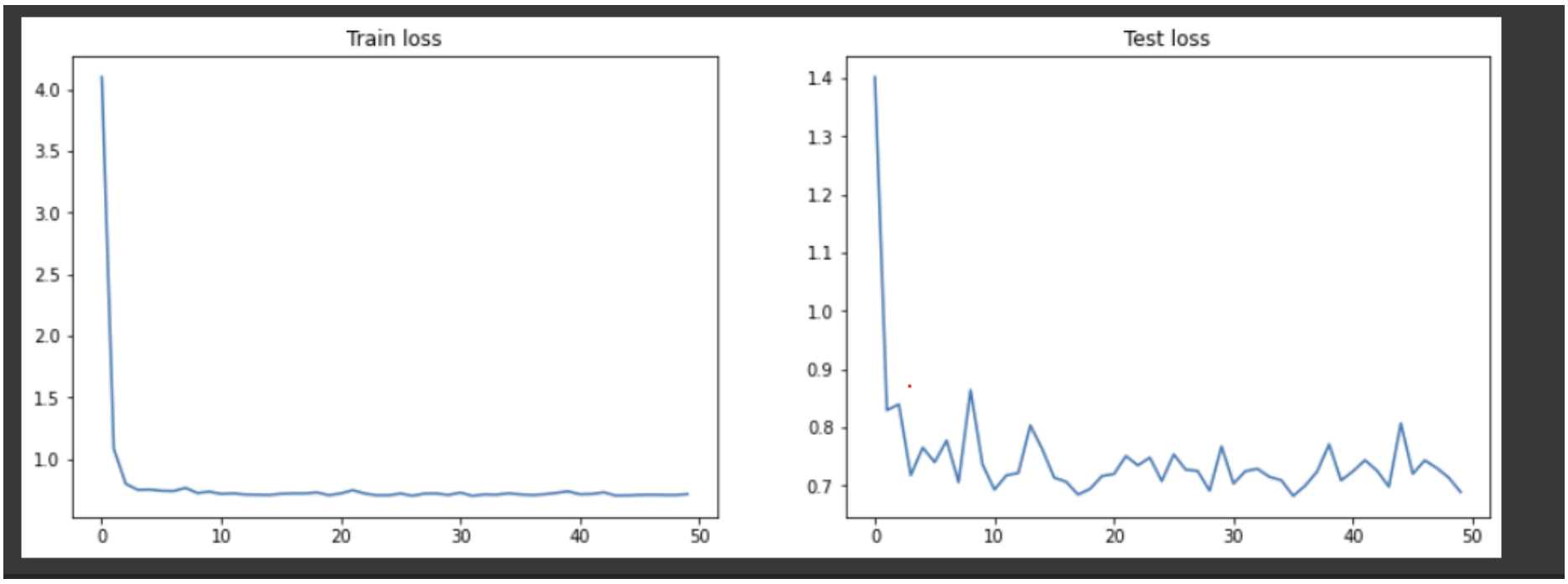
```
1  CAGR = lstm_book_df.loc[lstm_book_df.index[-1],'lstm_cumret'] ** (252./len(lstm_book_df.index))
2  Sharpe = np.mean(lstm_book_df['lstm_ret']) / np.std(lstm_book_df['lstm_ret']) * np.sqrt(252.)
3  VOL = np.std(lstm_book_df['lstm_ret']) * np.sqrt(252.)
4  MDD = historical_dd.min()
5  print('CAGR : ',round(CAGR*100,2),'%')
6  print('Sharpe : ',round(Sharpe,2))
7  print('VOL : ',round(VOL*100,2),'%')
8  print('MDD : ',round(-1*MDD*100,2),'%')
```

```
CAGR :  29.19 %
Sharpe :  0.78
VOL :  47.15 %
MDD :  75.32 %
```

애플, 마이크로소프트, 아마존 데이터 방출

나스닥 종합지수, 나스닥-100, S&P500, VIX 지수 만으로 데이터 구성.

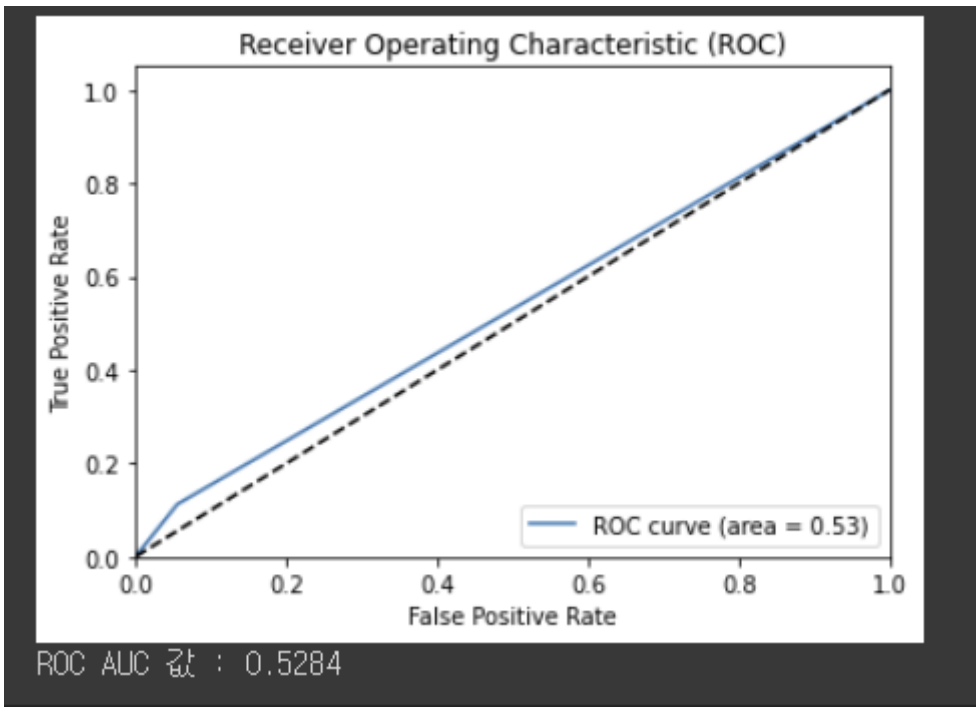
결과

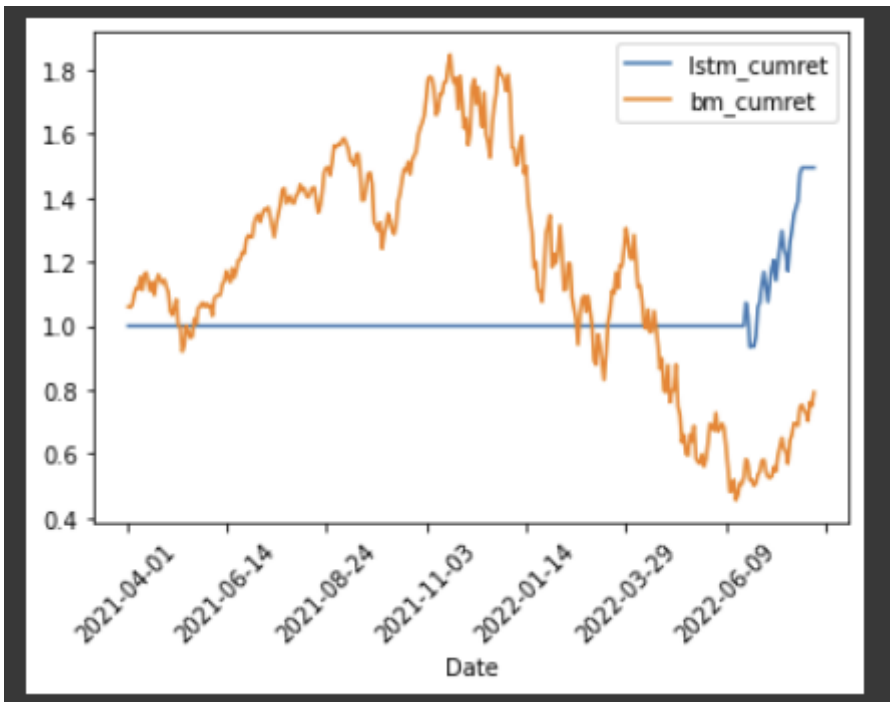



```
=======
50epochs_10batch
TN: 153
FN: 158
TP: 20
FP: 9
TPR: 0.11235955056179775
FPR: 0.05555555555555555
accuracy: 0.509
specitivity: 0.944
sensitivity : 0.112
mcc : 0.102
              precision    recall  f1-score   support

           0       0.49      0.94      0.65       162
           1       0.69      0.11      0.19       178

    accuracy                           0.51       340
   macro avg       0.59      0.53      0.42       340
weighted avg       0.60      0.51      0.41       340
=======
```


Receiver Operating Characteristic (ROC)
True Positive Rate
False Positive Rate
ROC curve (area = 0.53)
0.0
0.2
0.4
0.6
0.8
1.0
ROC AUC 값 : 0.5284


lstm_cumret
bm_cumret
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
2021-04-01
2021-06-14
2021-08-24
2021-11-03
2022-01-14
2022-03-29
2022-06-09
Date

바이앤홀드 전략 BM

```
[47]  1  CAGR = lstm_book_df.loc[lstm_book_df.index[-1],'bm_cumret'] ** (252./len(lstm_book_df.index)) -1
      2  Sharpe = np.mean(lstm_book_df['ret']) / np.std(lstm_book_df['ret']) * np.sqrt(252.)
      3  VOL = np.std(lstm_book_df['ret']) * np.sqrt(252.)
      4  MDD = historical_dd.min()
      5  print('CAGR : ',round(CAGR*100,2),'%')
      6  print('Sharpe : ',round(Sharpe,2))
      7  print('VOL : ',round(VOL*100,2),'%')
      8  print('MDD : ',round(-1*MDD*100,2),'%')
```

```
CAGR :  -15.66 %
Sharpe :  0.15
VOL :  74.38 %
MDD :  75.32 %
```

LSTM 전략

```
1  CAGR = lstm_book_df.loc[lstm_book_df.index[-1],'lstm_cumret'] ** (252./len(lstm_book_df.index)) -1
2  Sharpe = np.mean(lstm_book_df['lstm_ret']) / np.std(lstm_book_df['lstm_ret']) * np.sqrt(252.)
3  VOL = np.std(lstm_book_df['lstm_ret']) * np.sqrt(252.)
4  MDD = historical_dd.min()
5  print('CAGR : ',round(CAGR*100,2),'%')
6  print('Sharpe : ',round(Sharpe,2))
7  print('VOL : ',round(VOL*100,2),'%')
8  print('MDD : ',round(-1*MDD*100,2),'%')
```

```
CAGR :  34.08 %
Sharpe :  1.48
VOL :  21.41 %
MDD :  75.32 %
```

심층 신경망을 이용한 변동성 돌파 전략 주식 매매 방법에 관한 연구

매매 전략

당일 변동성 돌파 전략에 기반한 목표 매수가에 주식을 매수하고 다음 날 시가에 판매하였을 때의 수익 달성 여부를 예측한다.

당일 고가가 목표 매수가 이상인 경우와 미만인 경우를 분류하였다. 그리고 목표 매수가에 해당 주식을 매수 후 다음날 시가에 매도했을 경우의 수익률 계산을 위해 당일 목표 매수가 대비 다음날 시가의 증감율을 계산하였다.

매매 전략의 단점

변동성 돌파 전략에 기반한 매매 전략은 주식 종목의 가격이 목표 매수가 이상을 넘어서는 경우에만 매수가 일어나기 때문에 거래가 발생하는 경우가 제한적이며, 거래가 발생하더라도 다음날 시가에 매도했을 경우 수익이나 손실이 나는 경우의 수가 주식 종목마다 상이하다.

데이터 : 학습에 사용되는 지표

실제 거래 활성화 정도를 정확히 파악하기 위하여 단순한 거래량 대신 거래량과 종가를 곱한 거래액을 사용하였다.

표 2. 데이터 학습에 사용되는 지표

구분	항목	내용
가격	시가	당일 시가
	고가	당일 고가
	저가	당일 저가
	종가	당일 종가
	거래액	당일 거래량 X 종가
	기관 거래액	당일 기관 거래량 X 종가
	외국인 거래액	당일 외국인 거래량 X 종가
	외국인 보유액	당일 외국인 보유액
	외국인 보유율	당일 시가 총액 대비 외국인 보유액 비율
	매수 여부	당일 고가 >= 매수 목표가 여부
	수익률	(다음날 시가 / 목표 매수가 - 1) - 0.05
기술 지표	이동 평균선	5일, 10일 이동 평균선
	이동 표준 편차	5일, 10일 이동 표준편차
	전일 변동폭	전일 고가 - 저가
	볼린저 밴드	20일 이동 평균선, 상하위 2표준 편차선
	ATR	평균 변동성값
	모멘텀	1개월, 3개월
	ROC	3개월 모멘텀 변동 비율
	CCI	14일 CCI
	MACD	단기 기준일수 12일, 장기 기준일수 26일
	Williams %R	14일 Williams' %R
시장 지수	KOSPI 지수	당일 KOSPI 지수
	KOSPI 거래량	당일 KOSPI 거래량
	KOSDAQ 지수	당일 KOSDAQ 지수
	KOSDAQ 거래량	당일 KOSDAQ 거래량

> 아이디어 : Nasdaq 거래량, S&P500 거래량 데이터 추가

연구에서는 2021.08.31 을 기준으로 과거 100 거래일 동안의 데이터를 성능 평가를 위한 테스트 데이터로 나누고, 그 이전의 100 거래일을 검증 데이터, 그리고 그 이전 500 거래일 동안의 데이터를 훈련 데이터로 분류한다.

모델 설계 및 학습

- 손실 함수 BCE(Binary Cross Entropy) 사용
- optimizer : Adam
- 미니배치 학습법
- 각 은닉층마다 배치 정규화 적용
- (2, 3, 4, 5) 개의 은닉층 개수와 (200, 400, 600, 800, 1000) 개의 은닉층 유닛 수, (10, 16, 32, 64, 128, 256) 개의 배치 사이즈를 교차 검증하여 실험적으로 최적인 값을 찾았다.
- epochs : 5000
- dropout 비율 0.2로 적용
- ModelCheckPoint 콜백 사용해서 검증 데이터의 예측 정확도 값이 가장 낮은 값을 기록한 모델 저장.
- EarlyStopping 콜백 사용해서 100 epoch 동안 검증 데이터의 예측 정확도 값이 증가하지 않는 경우에는 학습을 중단하도록 했다.

⇒ 찾아낸 최적의 값은, 유닛 수 800, 은닉층 개수 4, 배치 사이즈 32 이며, 학습 정확도는 0.971, 검증 정확도 0.722를 기록했다.

성능 평가

성능 평가 방법으로는 예측 정확도, ROC 곡선, 수익률, CAGR, MDD, 변동성, 샤프 지수를 사용했다.

- 제안 모델 예측 결과 혼동 행렬

  다음은 해당 종목의 당일 고가가 목표 매수가 이상일 경우에, 다음날 수익 실현 여부 예측 결과를 혼동 행렬로 나타낸 결과이다.

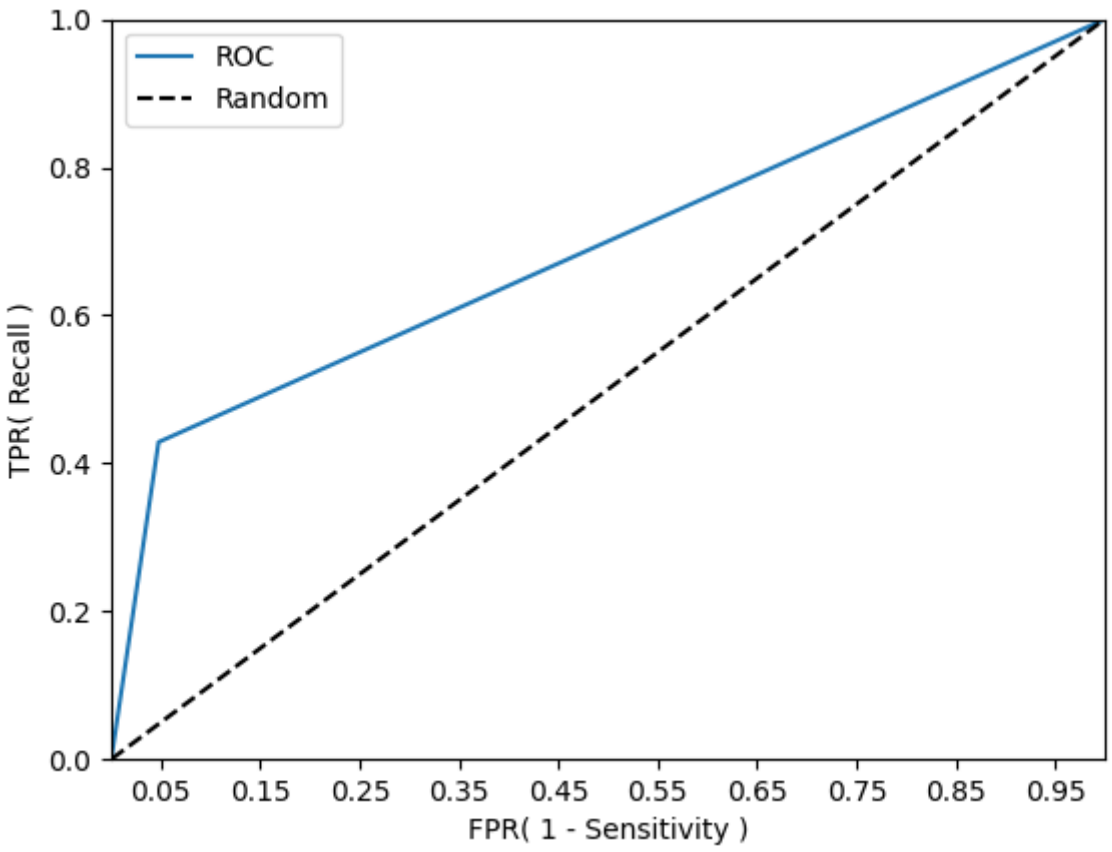
표 5. 제안 모델 예측 결과 혼동 행렬

		예측값	
		FALSE	TRUE
실제값	FALSE	20	1
	TRUE	8	6

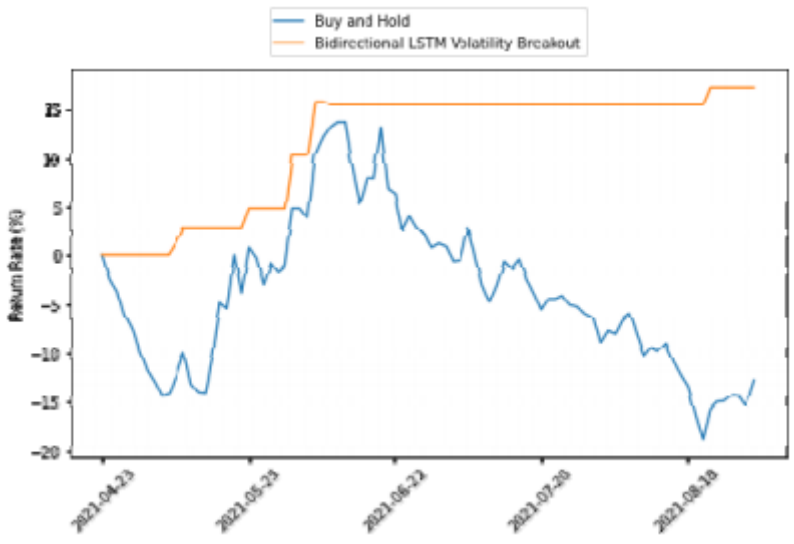
  - 정확도 : 74%
  - 정밀도 : 86%

- ROC 곡선과 AUC

  AUC = 0.691 측정



- 바이앤홀드와의 비교

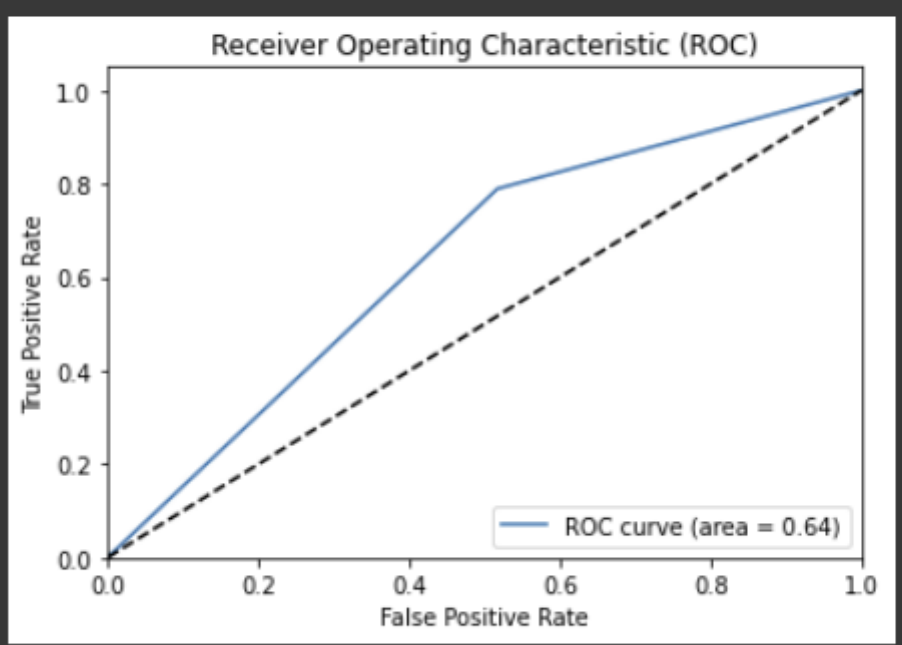


- CAGR : 56.0%
- 샤프 지수 : 3.541
- 변동성 : 12.9%
- MDD : 0.2%

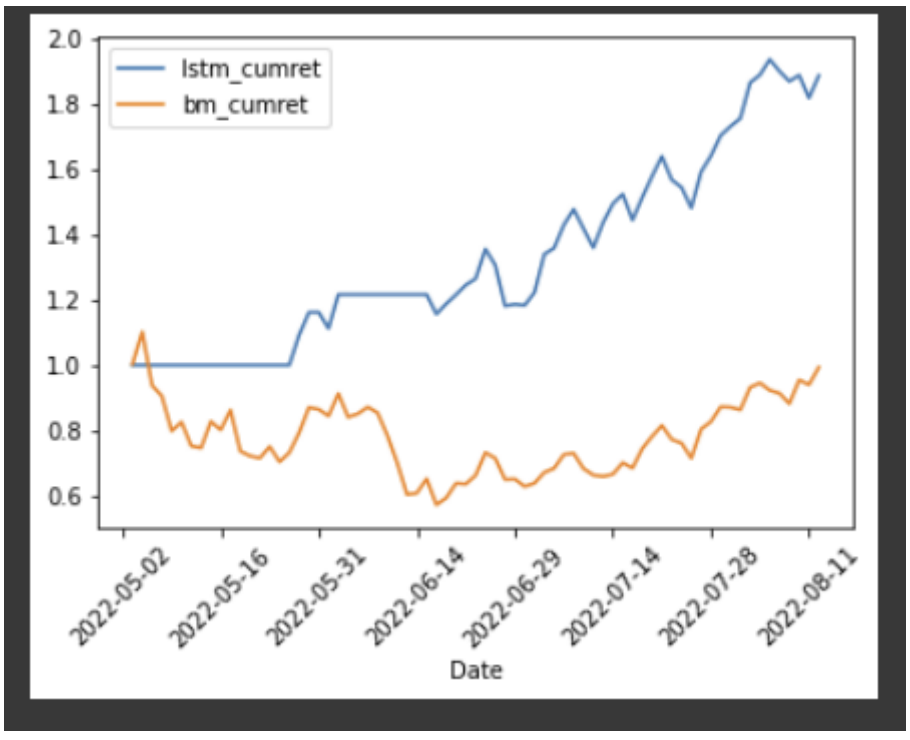
LSTM_Stock_v3

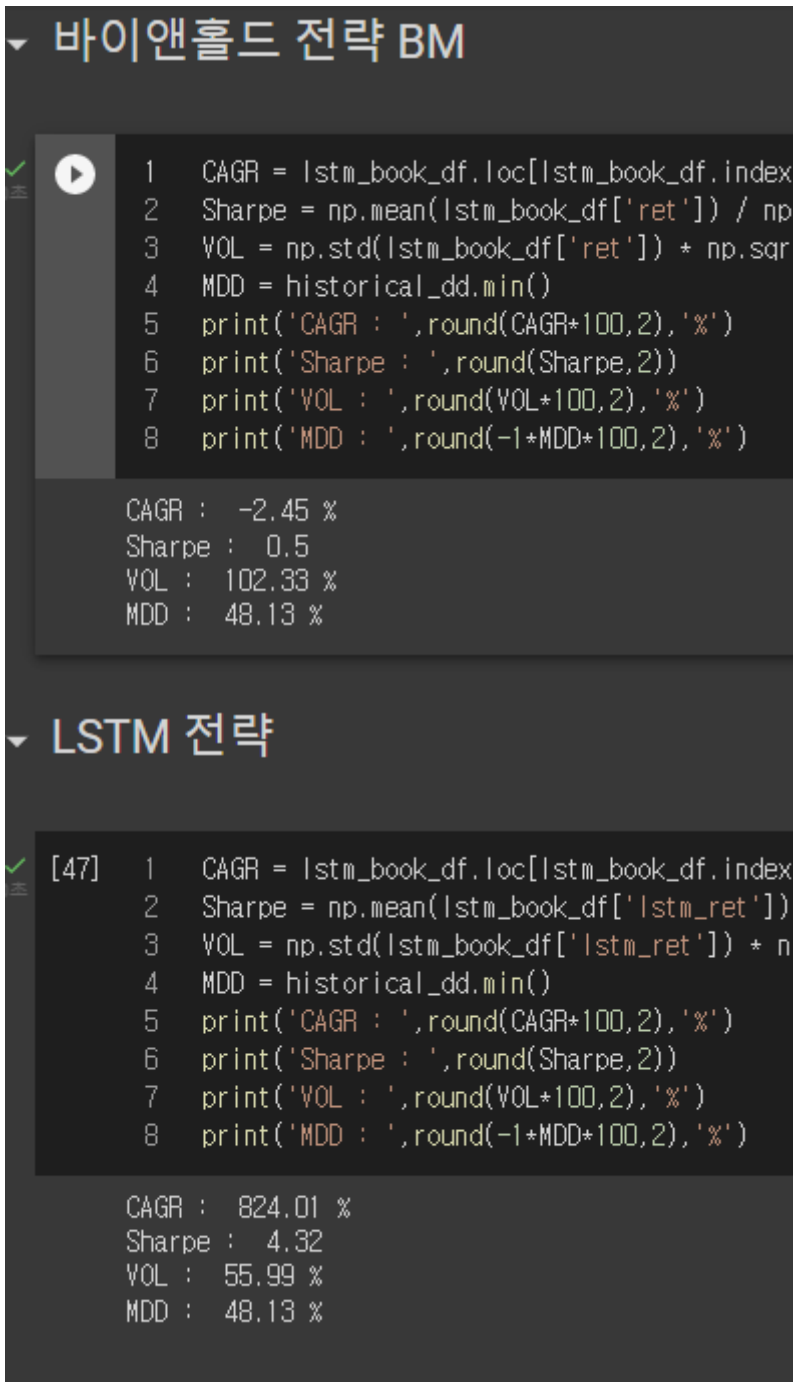
epochs = 50

```
plot_roc(y_pred,Y_test)
from sklearn.metrics import roc_auc_score
roc_score = roc_auc_score(Y_test,y_pred)
print('ROC AUC 값 : {0:.4f}'.format(roc_score))
```



ROC AUC 값 : 0.6361





mdd 가 꽤 큼.

여러 가지 실험들

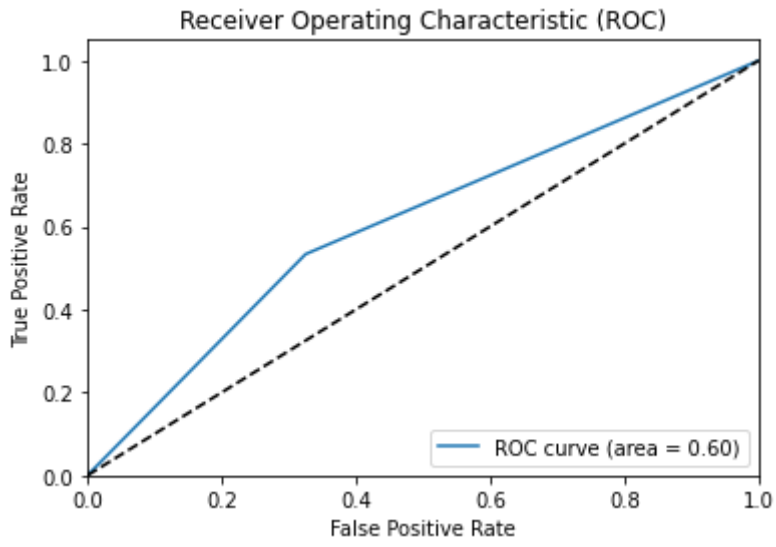
실험 1

실험 조건

- 은닉층 수 : 5
- 유닛 : 800
- 배치 사이즈 : 32
- 에포크 : 100 지정

결과 분석

- ROC 커브와 AUC



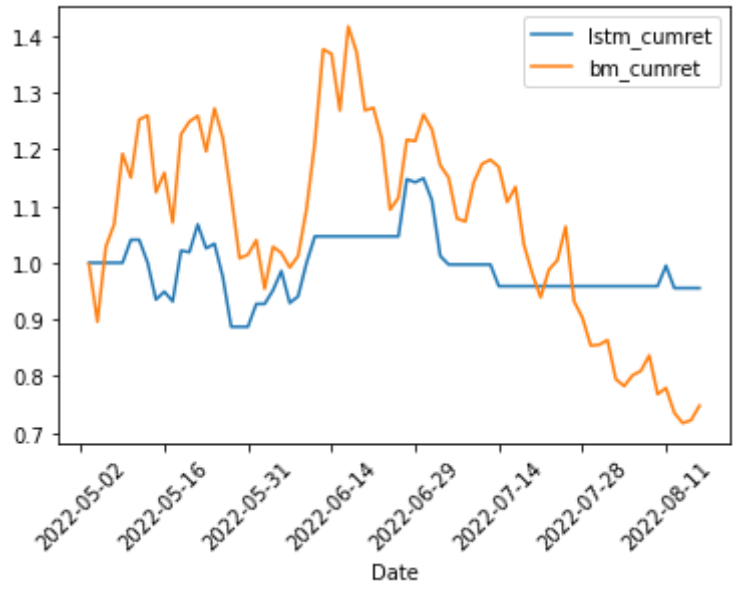
- 여러 가지 score

```
=======
100epochs_10batch
TN: 27
FN: 14
TP: 16
FP: 13
TPR: 0.5333333333333333
FPR: 0.325
accuracy: 0.614
specitivity: 0.675
sensitivity : 0.533
mcc : 0.209
              precision    recall  f1-score   support

           0       0.66      0.68      0.67        40
           1       0.55      0.53      0.54        30

    accuracy                           0.61        70
   macro avg       0.61      0.60      0.60        70
weighted avg       0.61      0.61      0.61        70
=======
```

- 바이앤홀드 전략과의 비교



	bm	lstm
CAGR	-62.33%	-14.22%
Sharpe	-0.49	-0.08%

VOL	100.22%	48.48%
MDD	49.39%	49.39%

- 모델명 : lstm_stock_v3_sqqq_2239_100.h5

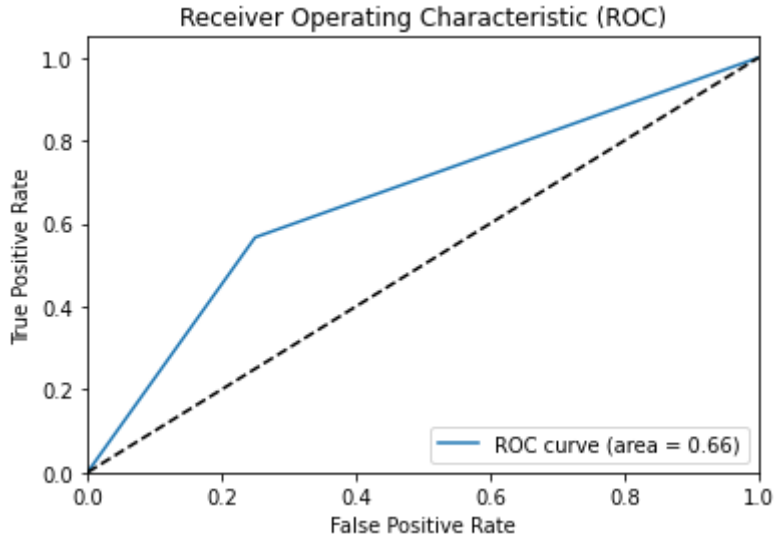
실험 2-1

실험 조건

- 은닉층 수 : 4
- 유닛 : 800
- 배치 사이즈 : 32
- 에포크 : 100 지정
-

결과 분석

- ROC 커브와 AUC



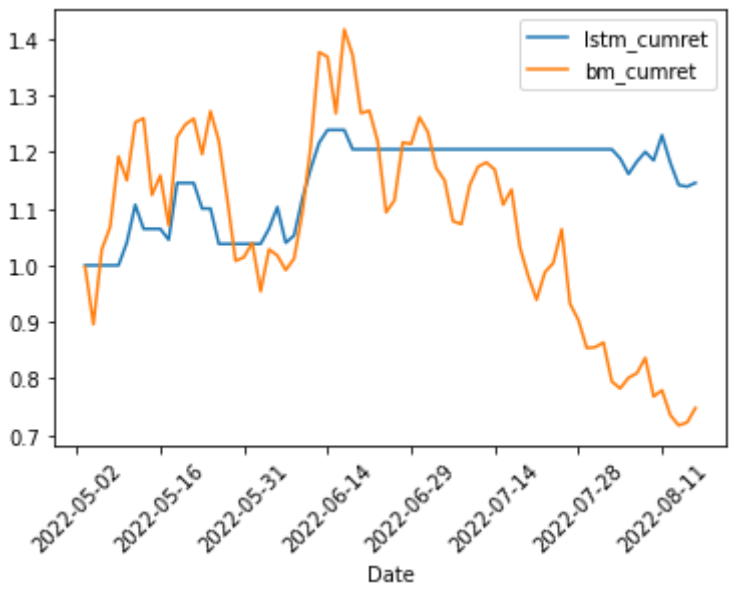
- 여러 가지 score

```
=======
100epochs_10batch
TN: 30
FN: 13
TP: 17
FP: 10
TPR: 0.5666666666666667
FPR: 0.25
accuracy: 0.671
specitivity: 0.75
sensitivity : 0.567
mcc : 0.322
              precision    recall  f1-score   support

           0       0.70      0.75      0.72        40
           1       0.63      0.57      0.60        30

    accuracy                           0.67        70
   macro avg       0.66      0.66      0.66        70
weighted avg       0.67      0.67      0.67        70
=======
```

- 바이앤홀드 전략과의 비교



	bm	lstm
CAGR	-62.33%	57.97%
Sharpe	-0.49	1.43%
VOL	100.22%	37.32%
MDD	49.39%	49.39%

- 모델명 : lstm_stock_v3_sqqq_2320_100.h5

실험 2-2

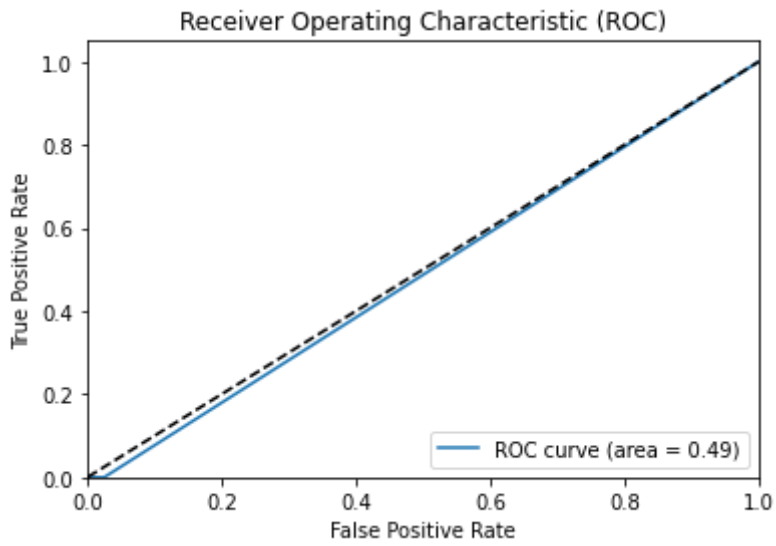
실험 조건

- 은닉층 수 : 4
- 유닛 : 800

- 배치 사이즈 : 16
- 에포크 : 100 지정
-

결과 분석

- ROC 커브와 AUC



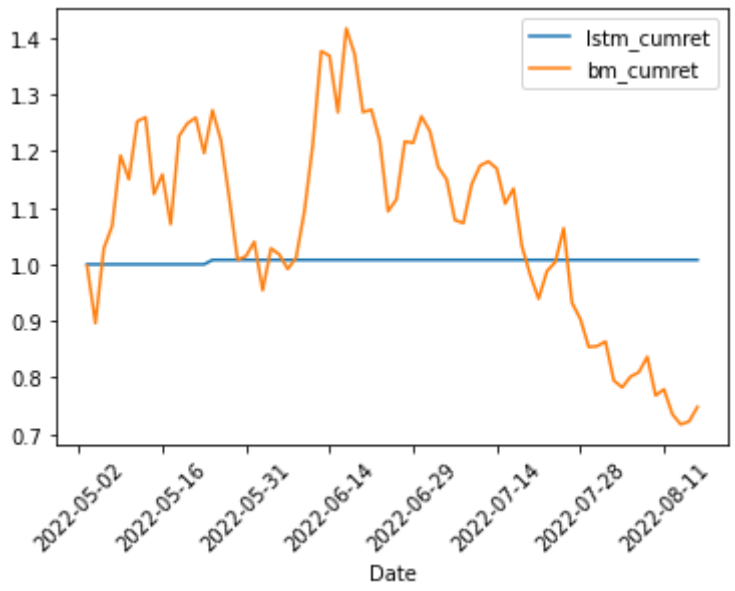
- 여러 가지 score

```
TN: 39
FN: 30
TP: 1
FP: 1
TPR: 0.03225806451612903
FPR: 0.025
accuracy: 0.563
specitivity: 0.975
sensitivity : 0.032
mcc : 0.022
              precision    recall  f1-score   support

           0       0.57      0.97      0.72        40
           1       0.00      0.00      0.00        30

    accuracy                           0.56        70
   macro avg       0.28      0.49      0.36        70
weighted avg       0.32      0.56      0.41        70
=======
```

- 바이앤홀드 전략과의 비교



	bm	lstm
CAGR	-62.33%	2.65%
Sharpe	-0.49	1.86%
VOL	100.22%	1.43%
MDD	49.39%	49.39%

- 모델명 : lstm_stock_v3_sqqq_2351_100.h5

실험 3

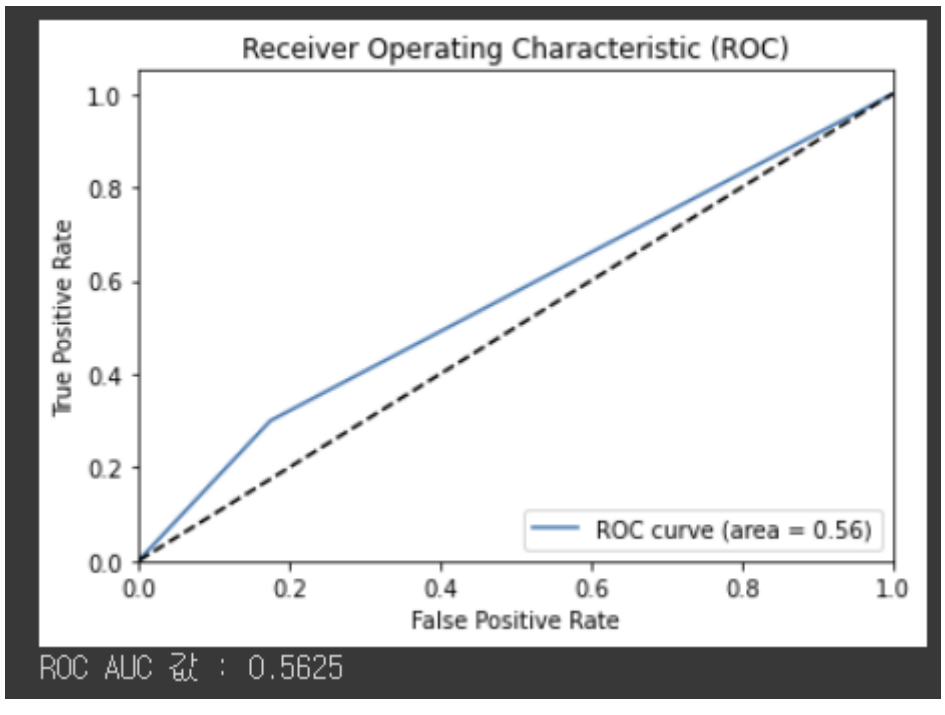
실험 조건

- 은닉층 수 : 4
- 유닛 : 800
- 배치 사이즈 : 32
- 에포크 : 100
- EarlyStopping 콜백 추가

```
early_stopping_cb = keras.callbacks.EarlyStopping(patience=20,
                                                  restore_best_weights=True)
```

결과 분석

- ROC 커브와 AUC



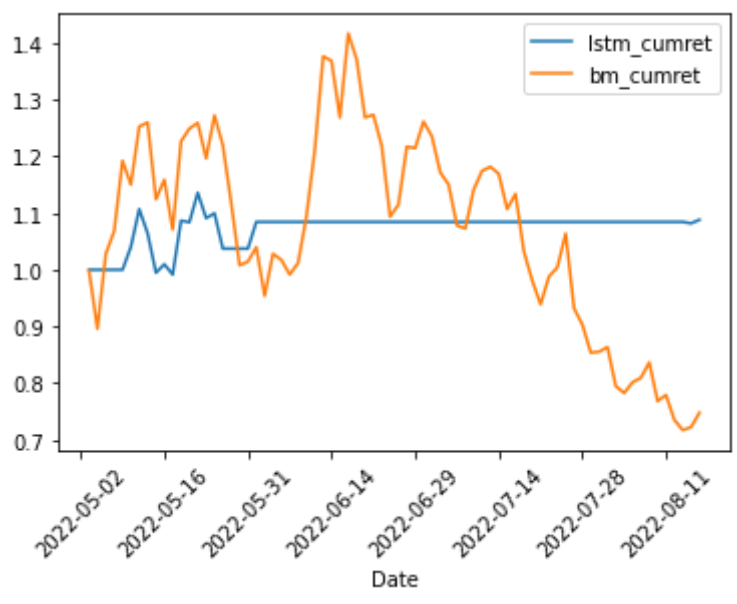
- 여러 가지 score

```
TN: 33
FN: 21
TP: 9
FP: 7
TPR: 0.3
FPR: 0.175
accuracy: 0.6
specitivity: 0.825
sensitivity : 0.3
mcc : 0.147
              precision    recall  f1-score   support

           0       0.61      0.82      0.70        40
           1       0.56      0.30      0.39        30

    accuracy                           0.60        70
   macro avg       0.59      0.56      0.55        70
weighted avg       0.59      0.60      0.57        70
=======
```

- 바이앤홀드 전략과의 비교



	bm	lstm
CAGR	-62.33%	32.84%
Sharpe	-0.49	1.05%

VOL	100.22%	32.16%
MDD	49.39%	49.39%

- 모델명 : lstm_stock_v3_sqqq_2405_earlystop49.h5