LSTM으로 구현해본 Fama·French 3요인 모형

-구현 실패로부터 파악한 금융데이터의 특징을 중심으로

Fama·French 3요인 모형 소개

Fama·French 3요인 모형 소개

- Eugene Fama교수와 Kenneth French교수의 『The Cross-Section of Expected Stock Returns(92년)』, 『Common risk factors in the returns on stocks and bonds(93년)』의 분석결과를 토대로 제시된 모형
- 92년, 93년 논문의 핵심은 '베타로 측정한 주식의 위험은 수익률을 예측할 수 있는 지표가 아니다'라는 것임.
- 3요인은 시가총액, BE/ME(=PBR의 역수), 시장을 의미
- 시가총액이 작고, PBR이 낮은 종목(가치주, 1이하)일수록 초과수익을 올리기에 유리하다는 것을 증명

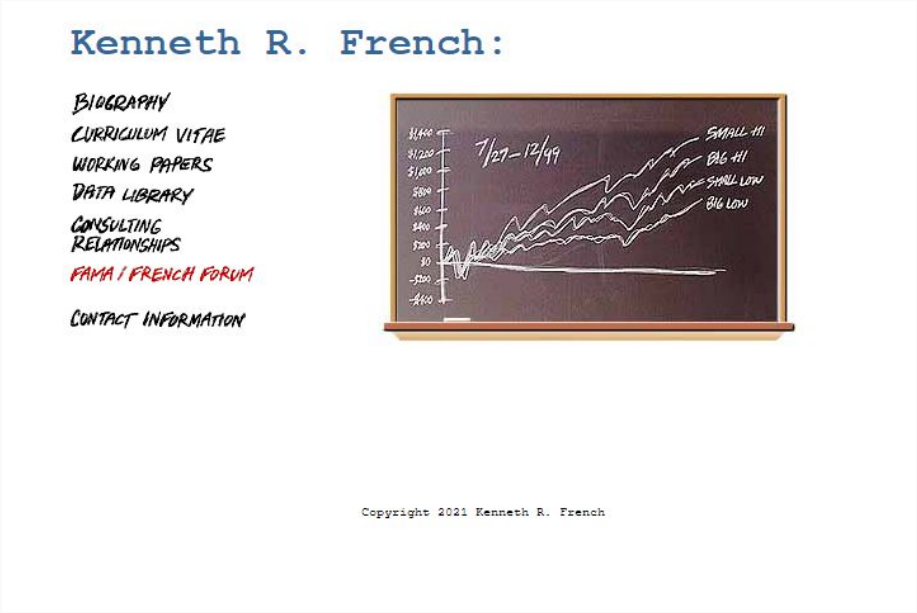
3요인 모델은 CAPM을 기반으로 한 모델로 **'주식의 체계적 위험'** 뿐만 아니라 **'소형주의 체계적 위험'**, **'가치주의 체계적위험'**이라는 리스크 요소를 추가하여 구성된 모델이다.

- $E(R) = r_f + \beta_{MKT}(rm - r_f) + \beta_{SMB} \cdot E_{SMB} + \beta_{HML} \cdot E_{HML}$
- E(R): 자산의 기대수익률
- r_f: 무위험이자율 또는 무위험수익률
- rm: 시장포트폴리오수익률
- $(rm - r_f)$: 초과 시장수익률
- E_{SMB}: 소형주의 대형주에 대한 시장의 평균적인 위험프리미엄(소형주)
- E_{HML}: 높은 BE/ME의 낮은 BE/ME에 대한 위험프리미엄(가치주)
- β_{MKT}: 주식의 체계적 위험과 그 민감도에 대한 보상
- β_{SMB}: 소형주 체계적 위험과 그 민감도에 대한 보상
- β_{HML}: 가치주 체계적 위험과 그 민감도에 대한 보상

3요인 모형 구현

Data set 소개

- French교수의 개인 홈페이지에서 미국의 최신 3요인과 5요인 포트폴리오 수익률 데이터를 구할수 있다.
- https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/
- 그러나 이왕이면 한국 주식시장을 분석해보는 것이 좀더 보람있다고 생각하였다.



Data set 소개

- 한국 주식시장의 3요인모형 관련 Data set은 투자회사인 한다 파트너스(Handa partners)가 제공해준 자료를 바탕으로 구성하였다.

  (https://gitlab.com/pr_handa/research)
- 다만 자료에서 무위험이자율과 자산의 기대수익율 자료를 구할수 없었으므로 무위험이자율의 대용치로 10년장기 국공채 수익률을 사용하였으며, 자산의 기대수익률의 대용치로 인덱스펀드 (미래에셋 코스피200인덱스 증권투자신탁 1호)의 수익률을 사용하였다.

QR코드를 통해 Data set과 모델을 구성한 코드를 보실수 있습니다.

(사용 언어: Python)

Parameters

- epoch = 300
- scaler = StandardScaler
- X_train: (1529, 3) Y_train: (1529, 1)
- X_test: (528, 3) Y_test: (528, 1)

- 활성함수: Relu
- 손실함수: mean Squared error
- optimizer: adam

Data set & Model

DateTime	mkt_excess	SMB	HML	종가	국고채10년(평균)	port_excess
2011-10-21	0.02	0.00	-0.00	0.02	0.04	-0.03
2011-10-24	0.03	-0.01	-0.00	0.04	0.04	-0.00
2011-10-25	-0.01	0.00	0.01	-0.01	0.04	-0.05
2011-10-26	0.00	0.00	0.00	0.00	0.04	-0.04
2011-10-27	0.01	-0.00	-0.01	0.02	0.04	-0.03
...	...	...	...	...	...	...
2020-02-24	-0.04	0.00	0.00	-0.04	0.02	-0.06
2020-02-25	0.01	0.01	-0.02	0.01	0.02	-0.01
2020-02-26	-0.01	0.00	0.00	-0.01	0.02	-0.03
2020-02-27	-0.01	-0.01	0.00	-0.01	0.02	-0.03
2020-02-28	-0.03	-0.00	-0.00	-0.03	0.02	-0.05

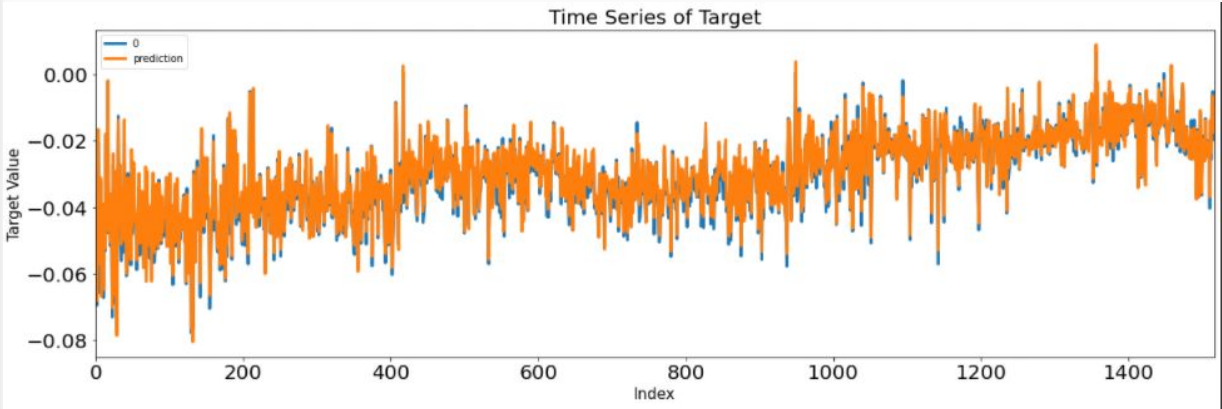
2057 rows × 6 columns

```python
# LSTM
model = Sequential()
model.add(LSTM(128, input_shape=(X_train.shape[1], X_train.shape[2]), return_sequences=True, activation='relu'))
model.add(Dropout(dropout_ratio))
model.add(LSTM(256, return_sequences=True, activation="relu"))
model.add(Dropout(dropout_ratio))
model.add(LSTM(128, return_sequences=True, activation="relu"))
model.add(Dropout(dropout_ratio))
model.add(LSTM(64, return_sequences=False, activation="relu"))
model.add(Dropout(dropout_ratio))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.summary()
model_fit = model.fit(X_train, Y_train,
                      batch_size=batch_size, epochs=epoch,
                      verbose=verbose)

plt.plot(pd.DataFrame(model_fit.history))
plt.grid(True)
plt.show()

```

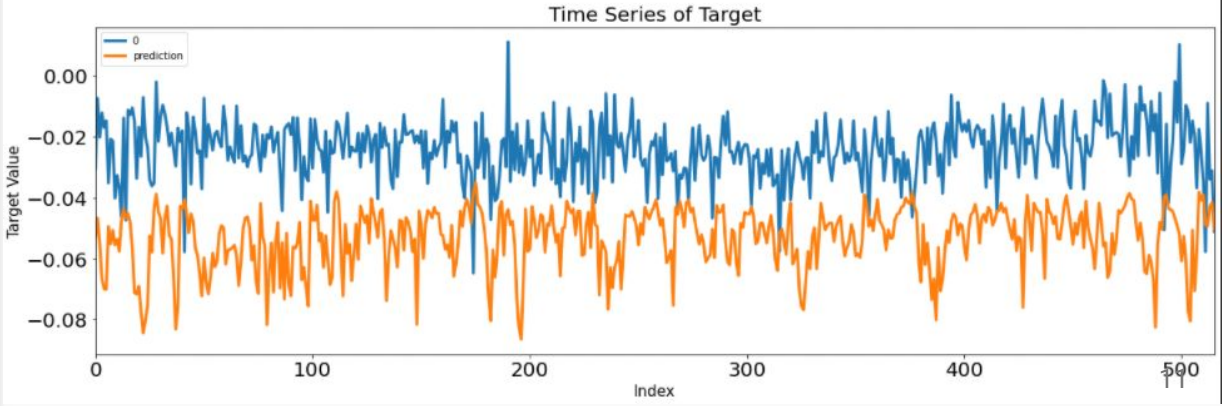
Drop out = 0%

	MAE	MSE	MAPE
Train	0.00	0.00	-4.59
Test	0.03	0.00	-176.87

Train data set



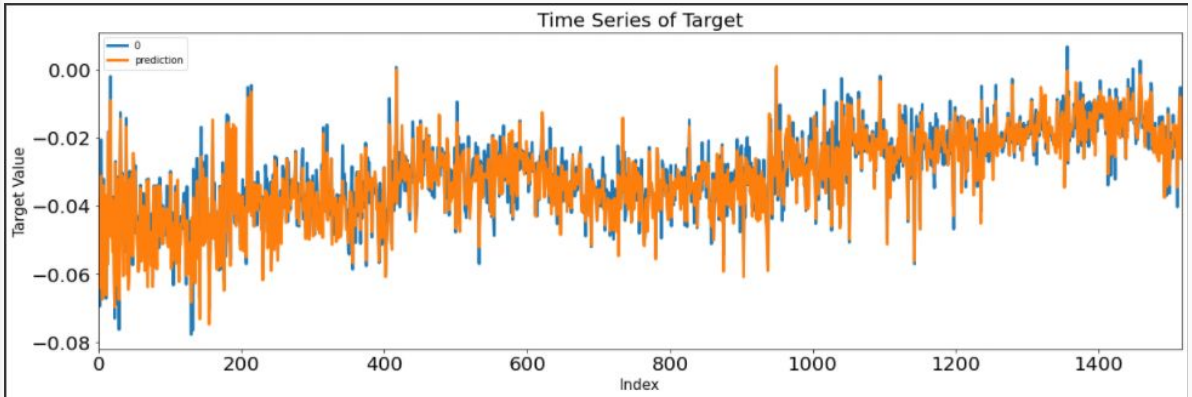
Test data set



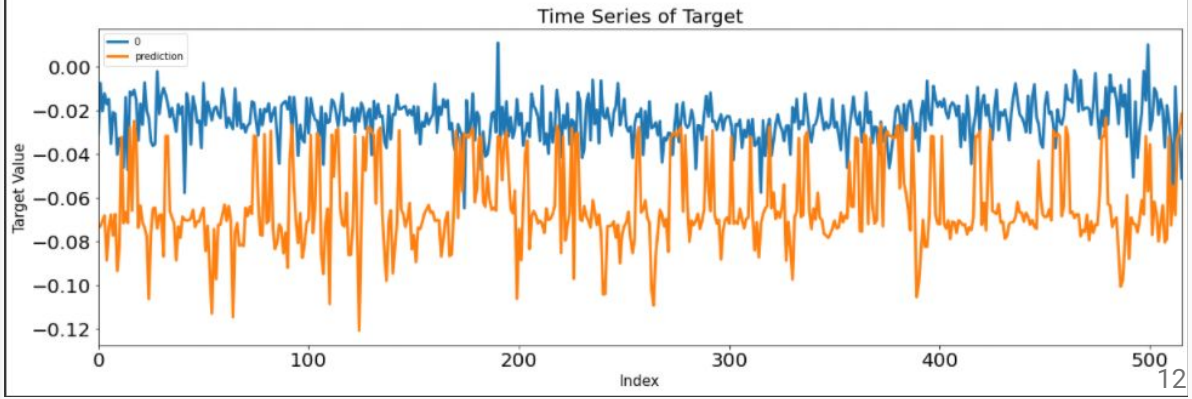
Drop out = 20%

	MAE	MSE	MAPE
Train	0.00	0.00	-8.97
Test	0.04	0.00	-238.85

Train data set



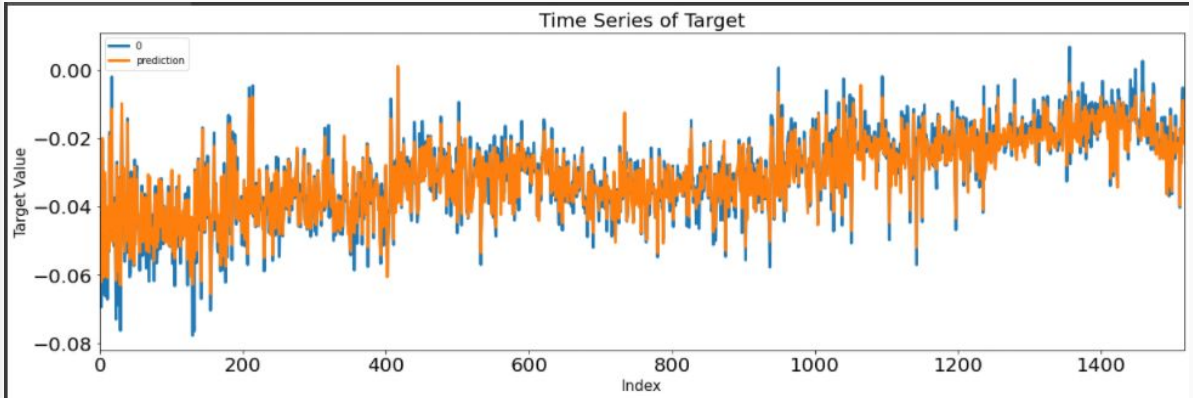
Test data set



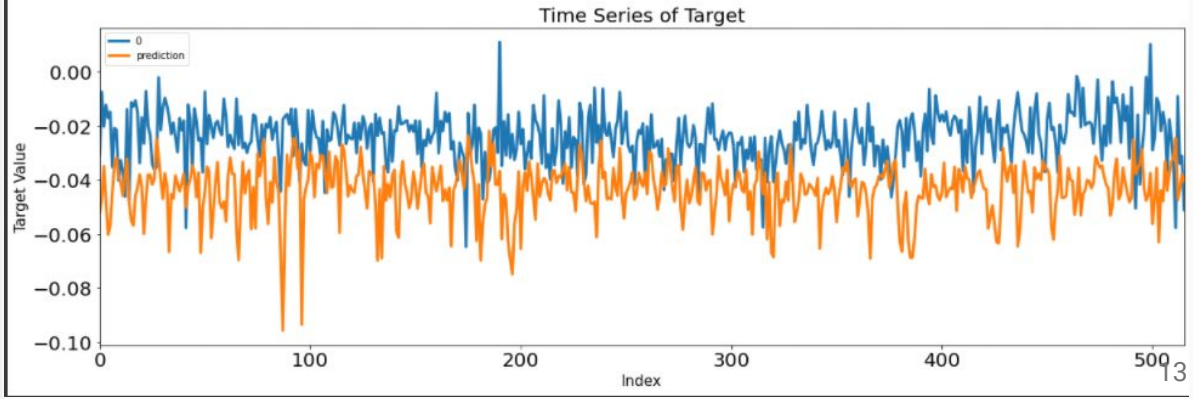
Drop out = 50%

	MAE	MSE	MAPE
Train	0.00	0.00	-14.36
Test	0.01	0.00	-59.40

Train data set



Test data set

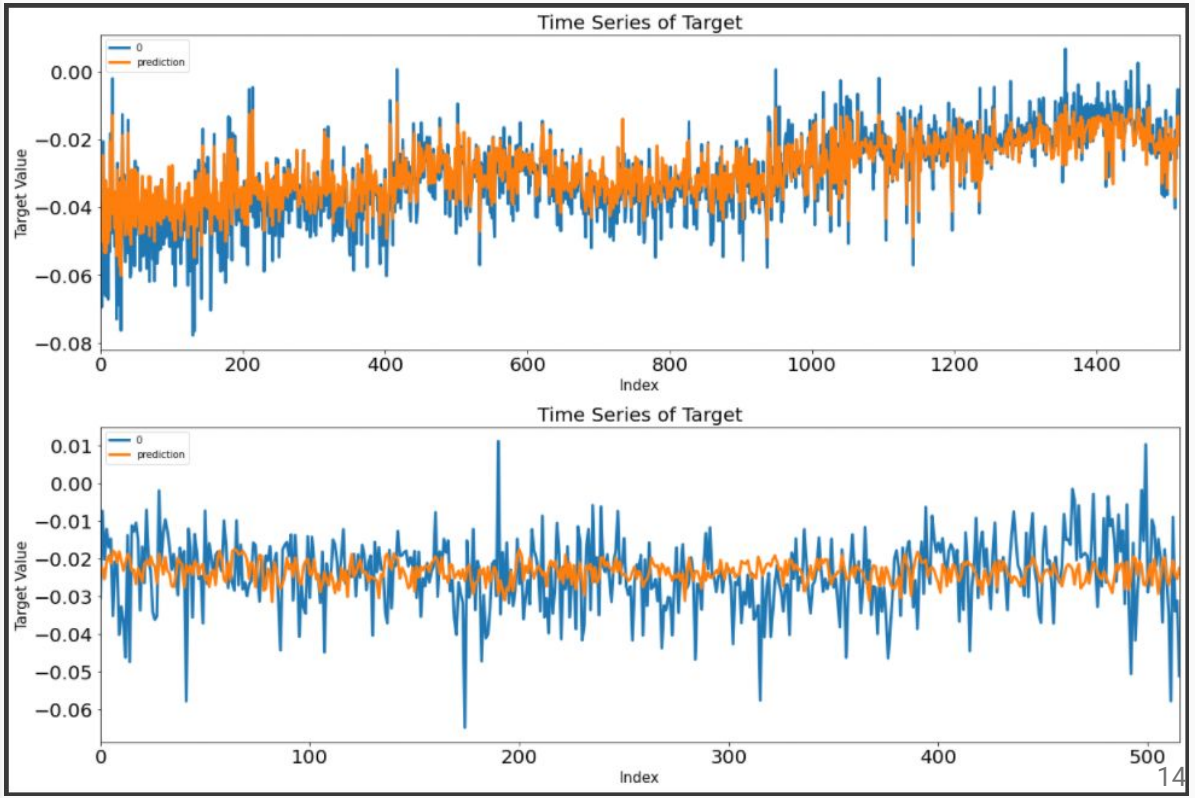


Drop out = 70%

Train data set

Test data set

	MAE	MSE	MAPE
Train	0.00	0.00	-9.60
Test	0.01	0.00	-45.24



모델링 결과 평가

- Training set의 학습효과는 잘 fit되었으나, Test결과 예측이 완전히 빗나감

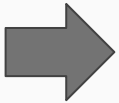
  → 과도한 학습으로 인한 Overfitting이 일어남

- Drop out을 시도하여 오버피팅을 줄였지만 만족할만한 예측을 하지 못함

도출한 금융 데이터의 문제점

문제점 1. 시계열 Feature자체의 노이즈

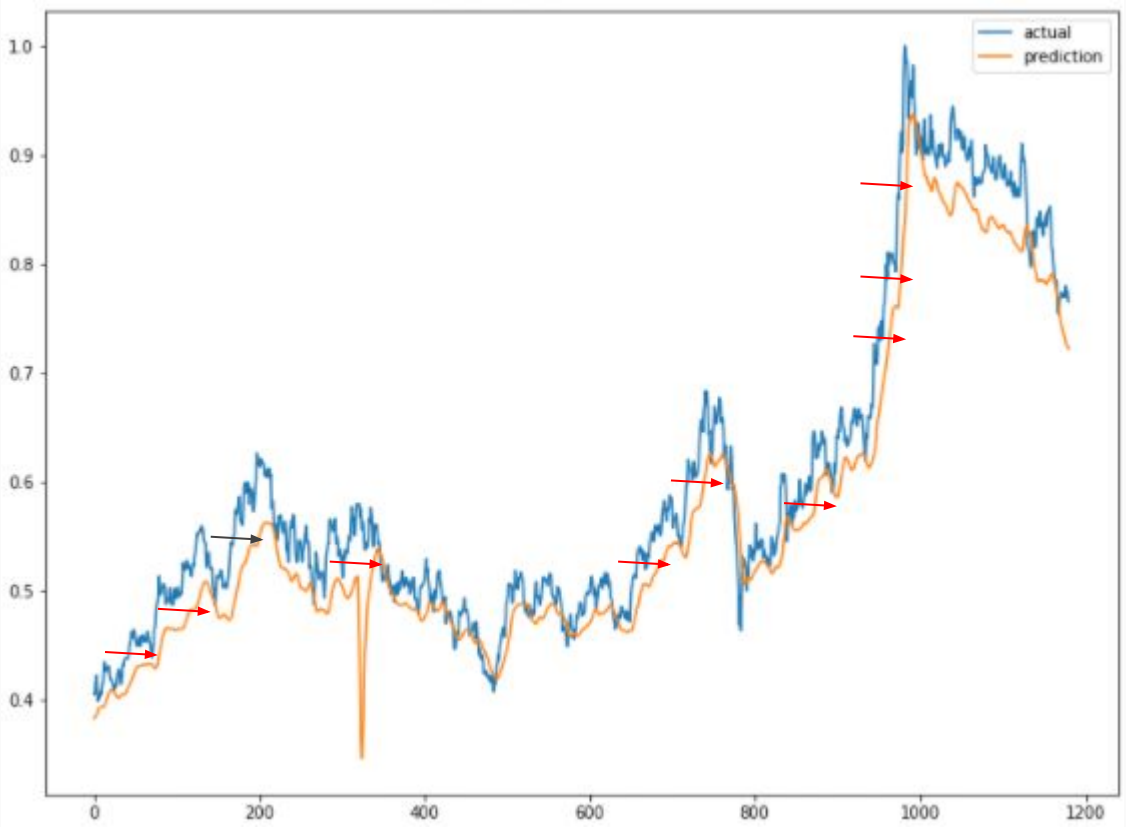
- **차기 주가 = 당기 주가+ 정보 + noise**
- 그러나 주가시계열 데이터는 패턴을 알수 있는 '정보'보다 noise양이 많다.
- 때문에 딥러닝 모형이 정보를 포착하여 학습하기가 어렵다.



결과적으로 차기 주가에 대한 최선의 예측값은 현재값이 되버림

1. 시계열 Feature자체의 노이즈

- 차기 주가에 대한 최선의 예측값이 현재의 주가가 되버리므로 모델학습결과 **그래프가 전체적으로 오른쪽으로 Lagging된 형태**로 나타남



문제점 2. 시계열 feature수 대비 짧은 시계열 길이

- 자산배분을 위해 고려해야할 요소들

-->자산군 모멘텀효과, 자산군 평균회귀 효과, 확장적 통화정책/긴축적 통화정책 분류, 장단기 부채 사이클 등의 macro데이터

→ 많은 요소들이 Feature로 사용되어야함

문제점 2. 시계열 feature수 대비 짧은 시계열 길이

- Feature로 사용되는 요소들이 늘어날수록 이를 위한 시계열 데이터의 길이도 늘어나야 한다.
- 하지만 금융데이터는 길어봐야 40년 수준.

→ Monthly데이터로 환산할 경우 40년 x 12월 = 480 row

→ 데이터 길이가 짧으므로 데이터를 설명하기 위해 많은 Feature를 집어넣게됨.

→ '차원의 저주' 가 생기고 Overfeatting이 매우 쉽게 일어남

문제점 극복을 위한 향후 연구방향

1. 시계열 Feature자체의 노이즈 → Denoising 방안 구상

- 시계열 자료의 denoising방법에는 Moving average나 Bilateral Filter등이 있음

- 하지만 가급적 연구자의 자의적 판단을 제거하고 학습과정에서 스스로 파라미터를 찾아내서 노이즈를 제거하는 방식으로 denoising을 구성하고자함

  → CNN을 기반으로한 Stacked AutoEncoder 등의 방식이 있다고 함

문제점 2. 시계열 feature수 대비 짧은 시계열 길이
→ 상황에 따른 Feature Rotation 구성을 고려

- 자료 길이가 짧으므로 무작정 많은 feature를 사용할 수 없음.
- 다만 적합한 피쳐들을 선별해두고 현재의 경제적 상황에 따라 모델이 피쳐들을 스스로 선별하여 Data set을 구성하게 한 후 학습하는 방식을 고려해 볼수 있다고 생각함.

  → 상황에 따라서 Model이 Feature를 스스로 선택하여 학습→ Feature Rotation

→ 경제 상황에 맞는 최적의 Macro data를 찾아내는 일이 중요

END

감사합니다.