머신러닝과 시계열 분석 모형을 활용한 서울 아파트 실거래가지수 예측

UNICON

(4강의장 1조 [UNICON조] 결과보고서)

데이터 전처리 방법

데이터 수집은

- 한국감정원
- 국가통계포털
- 한국은행
- 온나라 부동산포털

4개 포털에서

(2007.01 ~ 2019.12)

데이터수집

데이터 전처리는

- 30개 변수 중 상관분석을 통한 높은 상관관계에 해당하는 22개 변수 선정.

- 분석 기간별(data_1, data_2)로 나누어 정제하였으며, datetime column은 머신러닝/딥기법 데이터 처리 방식에 따라 Drop처리or유지

사용된 변수 설명

투자/공급/유동성/소비 대분류를 가지는 30가지의 변수를 토대로,
상관관계상 유의미한 변수 22개 선정.

- 양의 상관관계 (0.5 이상)
- 음의 상관관계 (-0.5 이하)

선정 변수

- kosave
- kosave_quant
- kosave_amt
- bc
- Ar
- unsoldapt
- tr_bond
- loan_apt
- cd
- houseloan_county
- houseloan_seoul

선정 변수

- cash_cur
- deli_ratio
- termdepo
- interrate_call
- interrate_kb
- cpi_jeon
- cpi_total
- ppi_realestate
- ppi_buildlease
- cci
- cli

변수명	변수의미	대분류
kosave	KOSPI 평균	투자
kosave_quant	KOSPI 거래량.일평균	
kosave_amt	KOSPI 거래대금.일평균	
mmc	MMF	
CMA	CMA	
bc	수익증권	
ar	매출어음	
wti	WTI.현물유가등락률	
CBD	회사채수익률	
unsoldapt apt	미분양주택현황.서울	공급
apt	서울.아파트.매매량.동호별	
tr_bond	국고채.3년.수익률	유동성
loan_apt	대출잔액.주택담보대출	
cd	CD(금리).수익률	
houseloan_county	가계대출.지역별.서울	
houseloan_seoul	가계대출.주택담보대출.서울	
cash_cur	현금통화	
deli_ratio	연체률.가계대출	
termdepo	만기2년미만정기예적금	
interrate_call	무담보콜금리.전체	
interrate_kb	한국은행 기준금리	
M2	통화량	
MMI	광공업지수	소비
bsi	경제심리지수	
cpi_month	소비자물가지수_월세.서울	
cpi_jeon	소비자물가지수_전세.서울	
cpi_total	소비자물가지수_전체.서울	
ppi_realestate	생산자물가지수_비주거용.부동산관리.전국	
ppi_buildlease	생산자물가지수_비주거용.건물임대.전국	
ppi_total	생산자물가지수_총지수	
unemprate	실업률	
cci	경기종합지수	
cli	선행종합지수	

BACKGROUND

“국가, 기업, 가계가 보유한 자산 중에서 부동산으로 편중된 자산 구조로 인해 부동산 가격 변동은 경제 상황에 큰 영향을 미치게 된다. 이로 인해 부동산 가격의 상승 또는 하락 여부는 주요 관심사항이며, 가격변화에 대비하기 위해 다양한 방법을 이용하여 부동산 가격지수예측을 연구하고 있다.”

6.8%

국민 전체의 순자산 작년대비 증가

55%

비금융 순자산의 토지 자산 비중

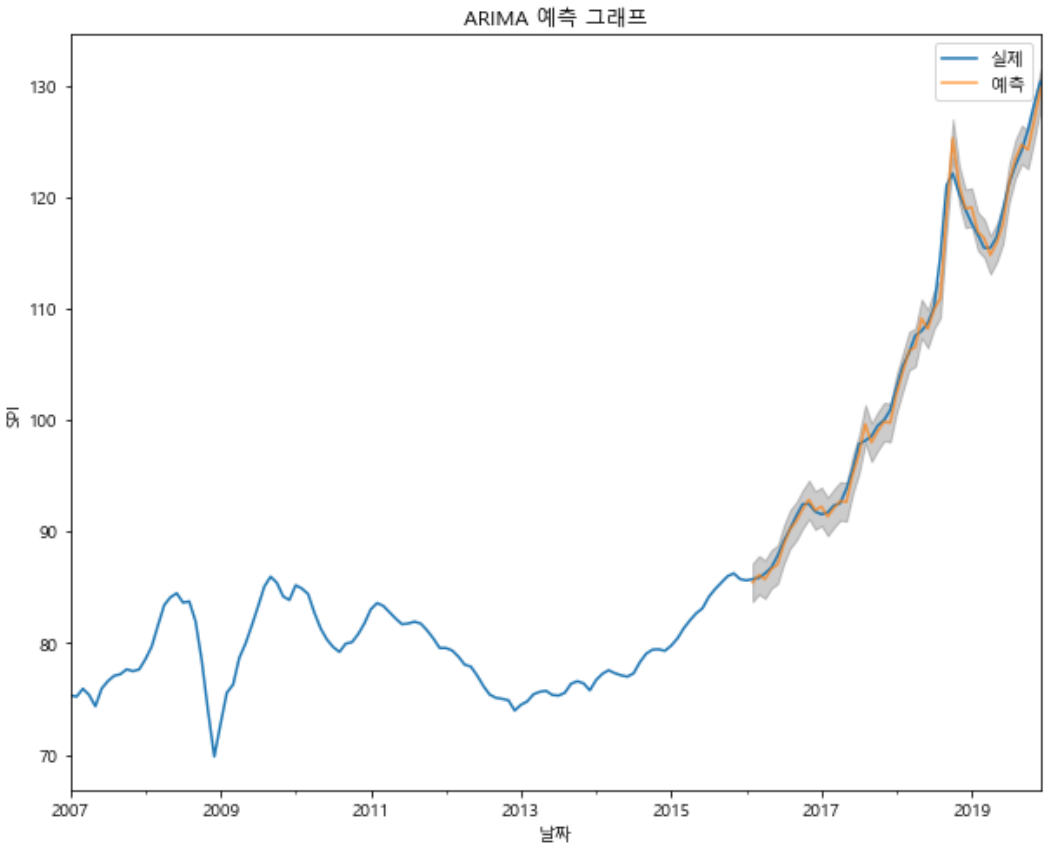
76%

일반정부, 비금융, 금융기업의 순자산 중 부동산 비중

선행 예측 연구 기법 – ARIMA

- 비정상적 시계열 자료에 대해 분석하는 방법
- 시계열의 변동형태를 파악, 예측이 가능하다는 장점으로 증권시장 등 경제분야에 응용
- 특히, 시간의 흐름에 따라 변동이 빠를 때 이를 민감하게 반영 가능함

MAE	RMSE
0.7659	1.0774



INDEX

선행 연구

1. **전통적 시계열 분석**
   - ARIMA 모형
   - VAR 모형
   - VECM 모형
   - 베이지언 VAR 모형

2. **머신러닝 모형**
   - 다층퍼셉트론
   - DNN
   - LSTM
   - SVR
   - RF

- 김근용(1998) ARIMA 모형
- 손정식 외(2002) ARIMA 모형과 VAR모형
- 임정식(2014) 자기회귀오차모형, ARIMA모형, 개입분석모형
- 김성환 외(2016) 베이지언 개념을 도입하여 기존 VAR모형의 한계 극복
- 함종영 · 손재영(2016) VAR모형과 베이지언 VAR모형
- 정원구 · 이상엽(2007) 2개의 은닉층으로 구성된 인공신경망
- 이형욱 · 이호병(2009) ARIMA모형과 2개의 은닉층으로 구성된 인공신경망
- 배성완 · 유정석(2017) ARIMA모형, DNN, LSTM모형
- 민성욱(2017)선형회귀 모형, SVM, 랜덤포레스트, 인공신경망 모형

목적

기존의 선행연구를 바탕으로 서울 아파트 실거래가지수 예측에 사용될 수 있는 머신러닝 기법을 적용하며 예측력을 비교 분석

1

독립변수 선정을 어떻게 했으며 선행 연구들과 차이점은 무엇인가?

2

선정된 변수로 머신러닝과 딥러닝 기법을 어떻게 실행 하였는가?

3

선정된 5개의 머신러닝 기법 중 예측 정확도가 높은 기법은 어떤것인가?

목적

기존의 선행연구를 바탕으로 서울 아파트 실거래가지수 예측에 사용될 수 있는 머신 러닝 기법을 적용하며 예측력을 비교 분석

선행연구와의 차별성

1 선행연구에서 사용한 독립변수를 상관분석으로 차원 축소

2 기존 연구들에서 사용되었던 머신러닝 모형외에 Attention Mechansim을 적용

데이터 설명

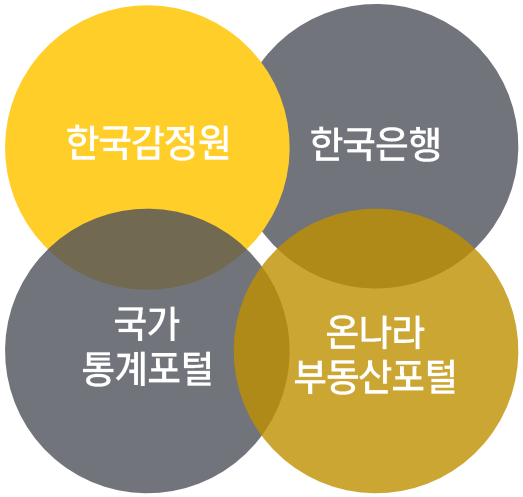
공간적 범위

2007–2019년

시간적 범위

22개

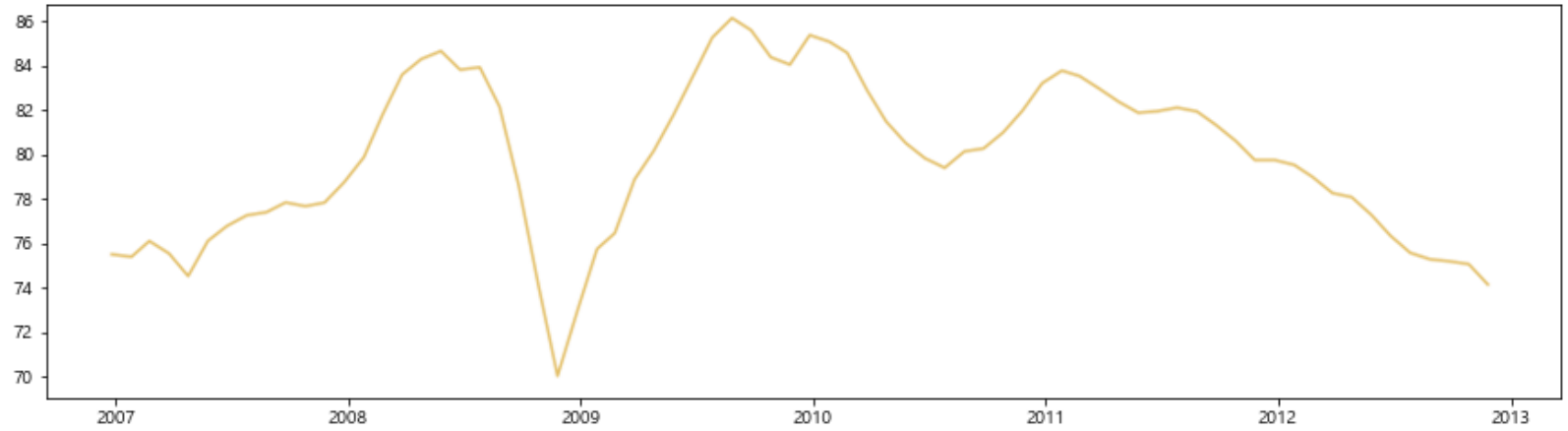
독립변수



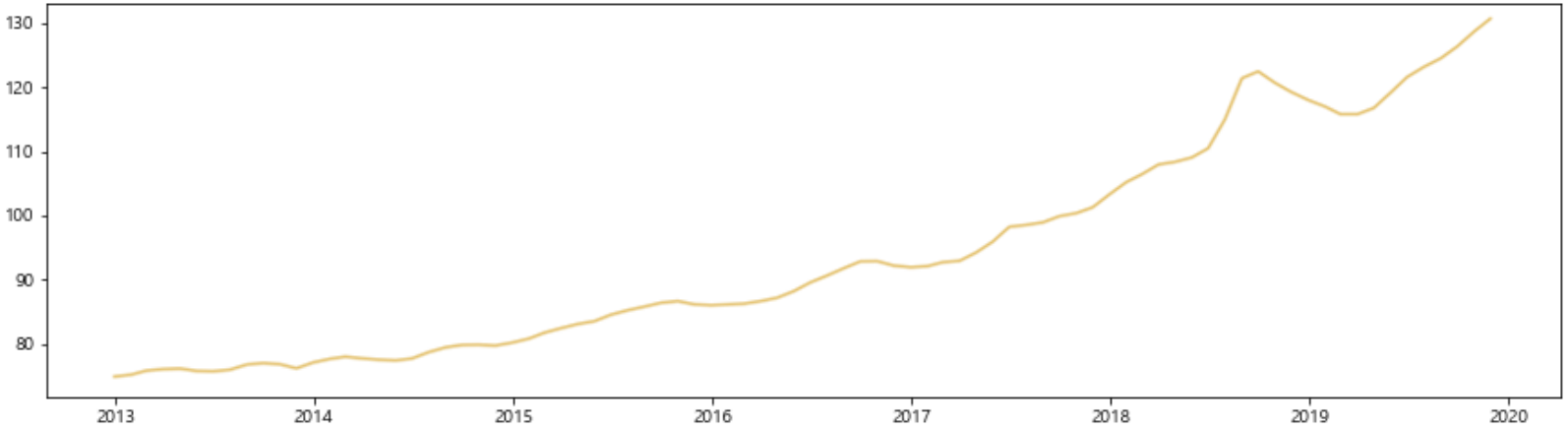
데이터 출처

데이터 설명

기간 I
2007년 1월 – 2012년 12월



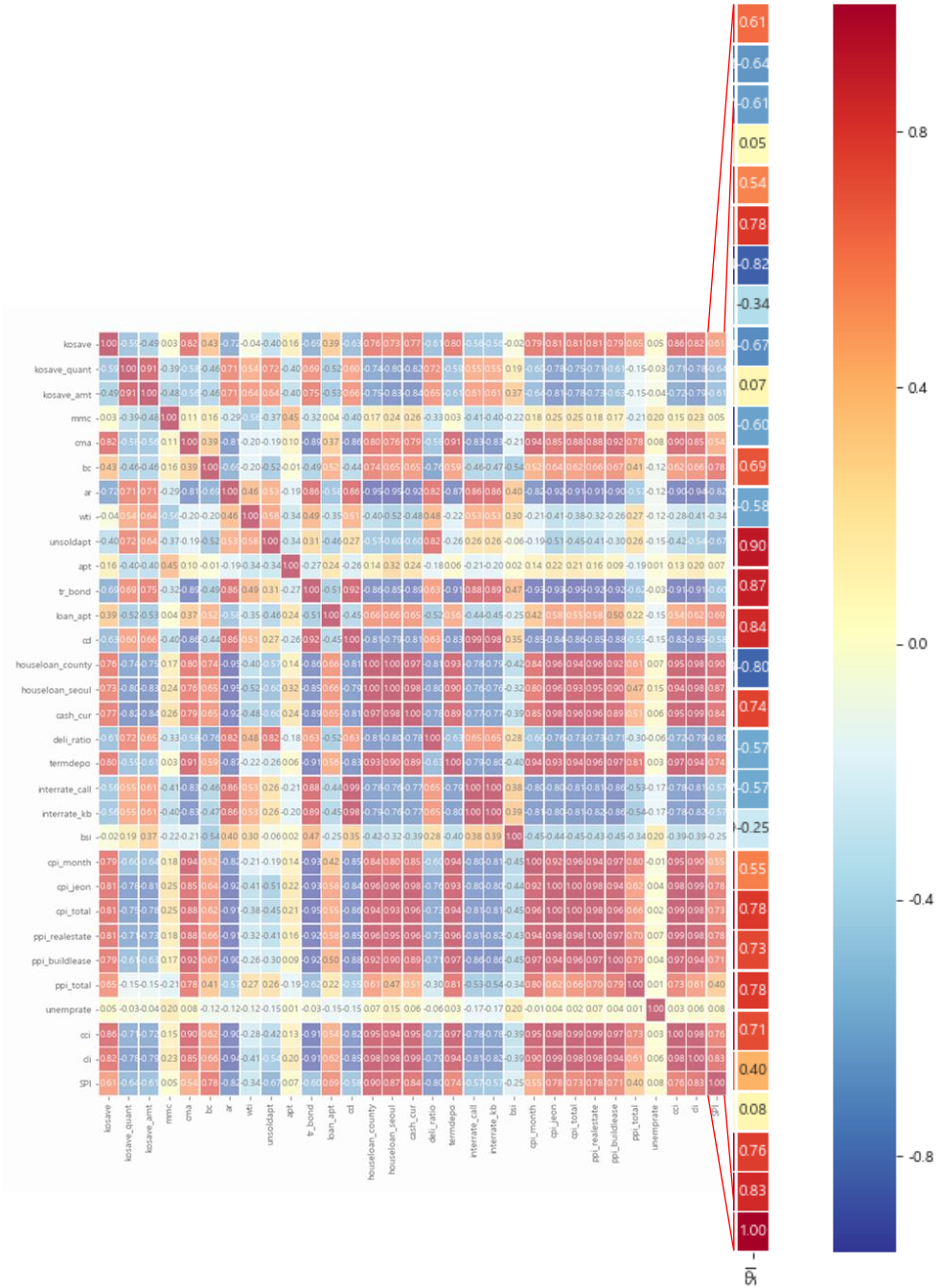
기간 II
2013년 1월 – 2019년 12월



데이터 설명

종속변수인 “SPI”와 상관분석을 통해
22개의 독립변수를 선택

상관계수가 0.5이상인 양적관계와
-0.5이하인 음적 상관관계를 가진 변수들을 채택

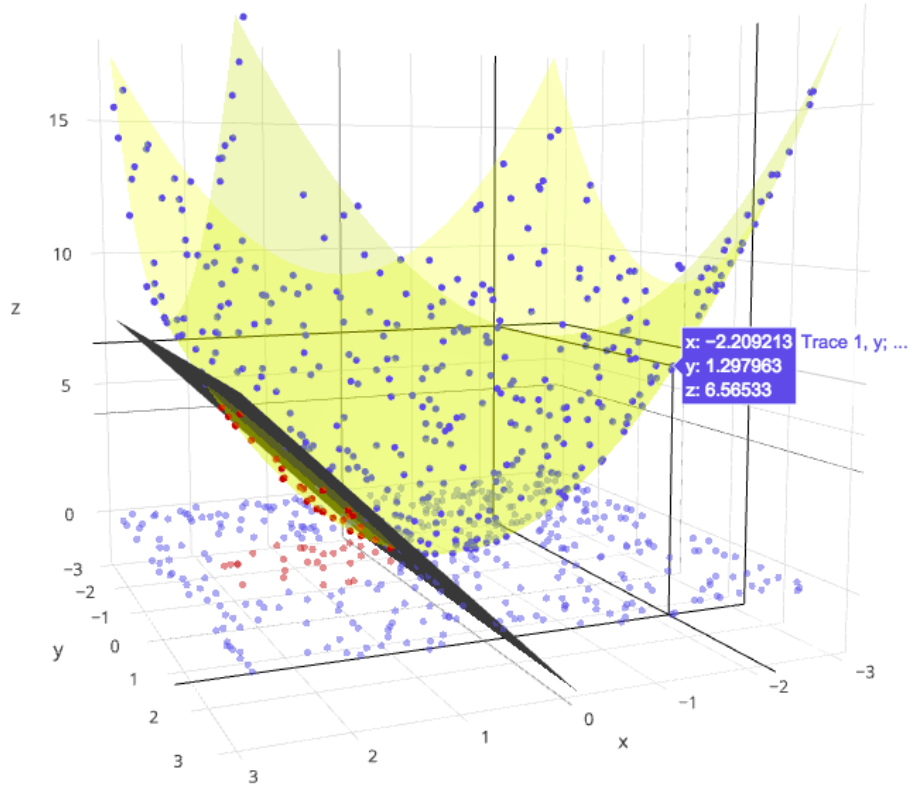


SUPPORT VECTOR REGRESSION

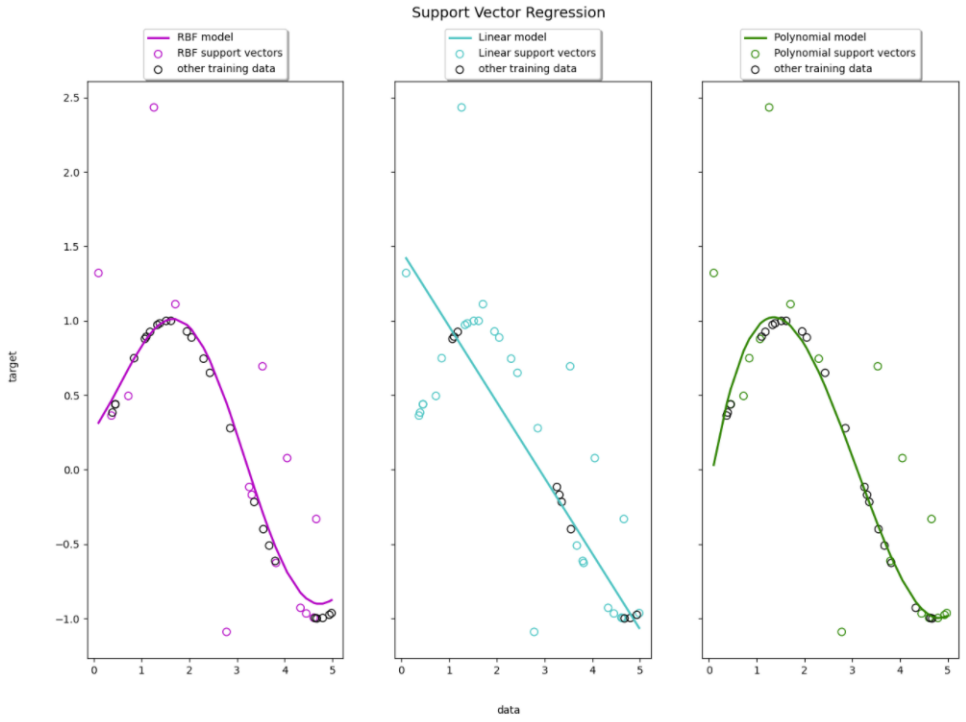
배경

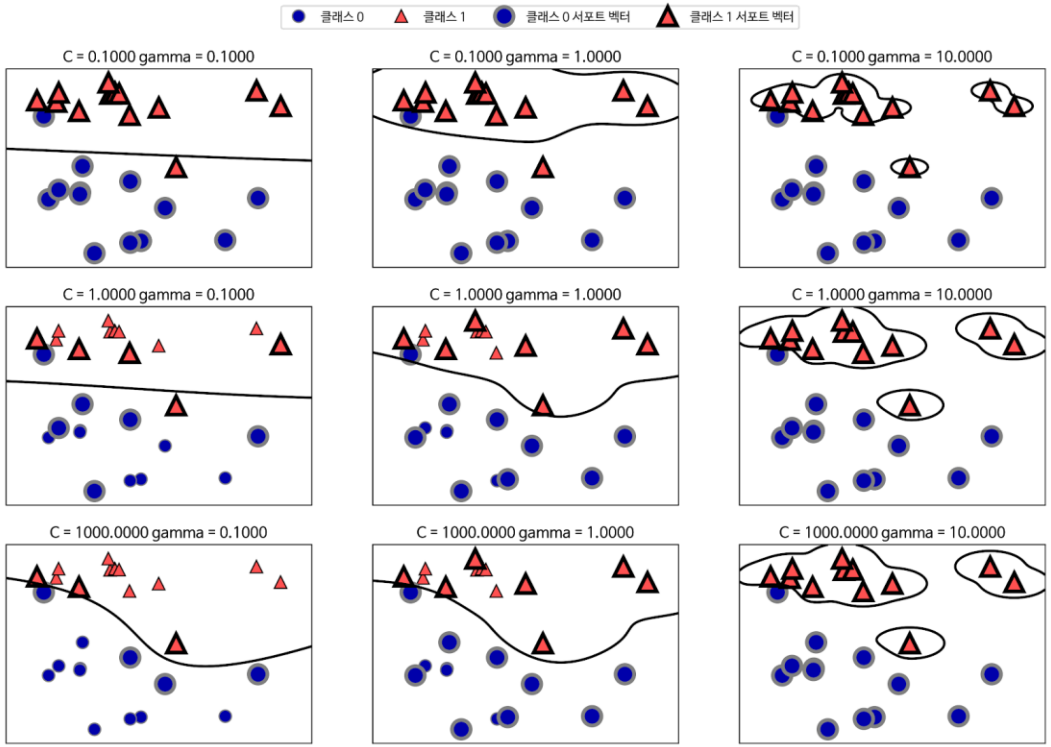
시계열로부터 단순한 초평면으로 정의되지 않는 더 복잡한 모델로 만들 수 있도록 확장

데이터셋에 비선형 특성 추가. 고차원에서 분류기 학습(커널 기법)



실전 적용 및 파라미터 이해





Kernel : Linear, RBF, Polynomial

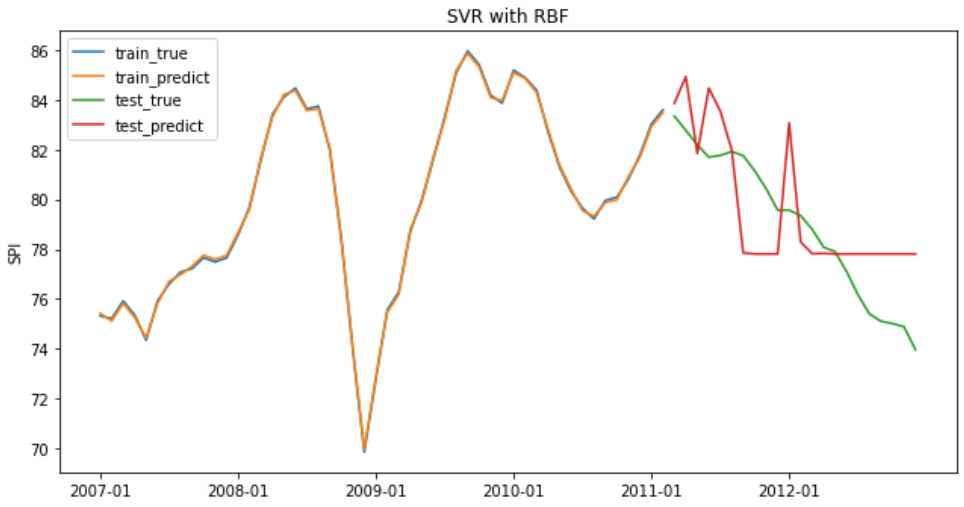
C(cost) : 규제 매개변수. 클수록 error를 적게 허용

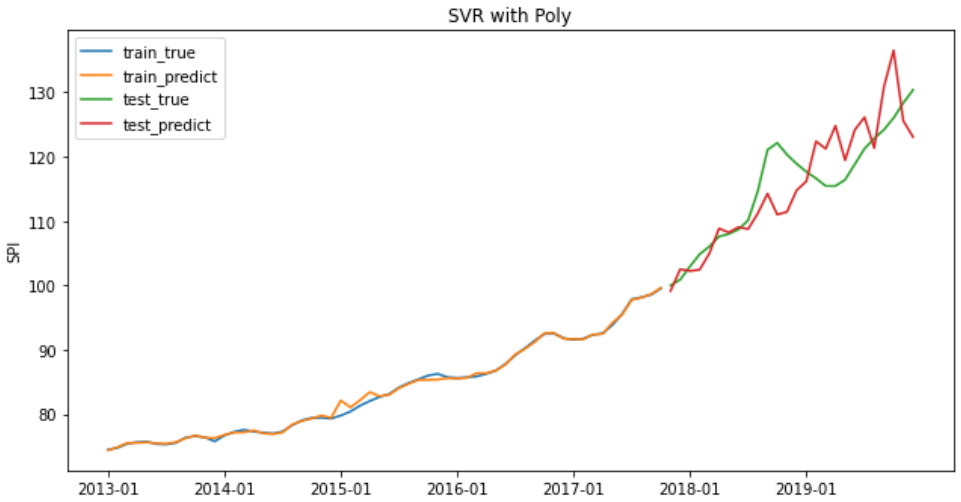
gamma : 하나의 데이터 샘플이 영향력을 미치는 거리 결정. 클수록 포인터들이 영향력 행사하는 거리 짧아짐

SVR

Scaler = {MinMax, Standard, Robust}
Kernel = {Linear, Poly, RBF}
C = {100, 200, 300, 500, 1000}
degree = {1, 3, 5, 7, 10}
gamma = {0.1, 0.3, 0.5, 0.7, 1}

	기간 I	기간 II
파라미터	Scaler = Robust Kernel = RBF C = 500 gamma = 0.038	Scaler = Robust Kernel = Poly C = 270 degree = 2 gamma = 0.1
MAE	1.9193	2.2809
RMSE	4.1796	5.2946





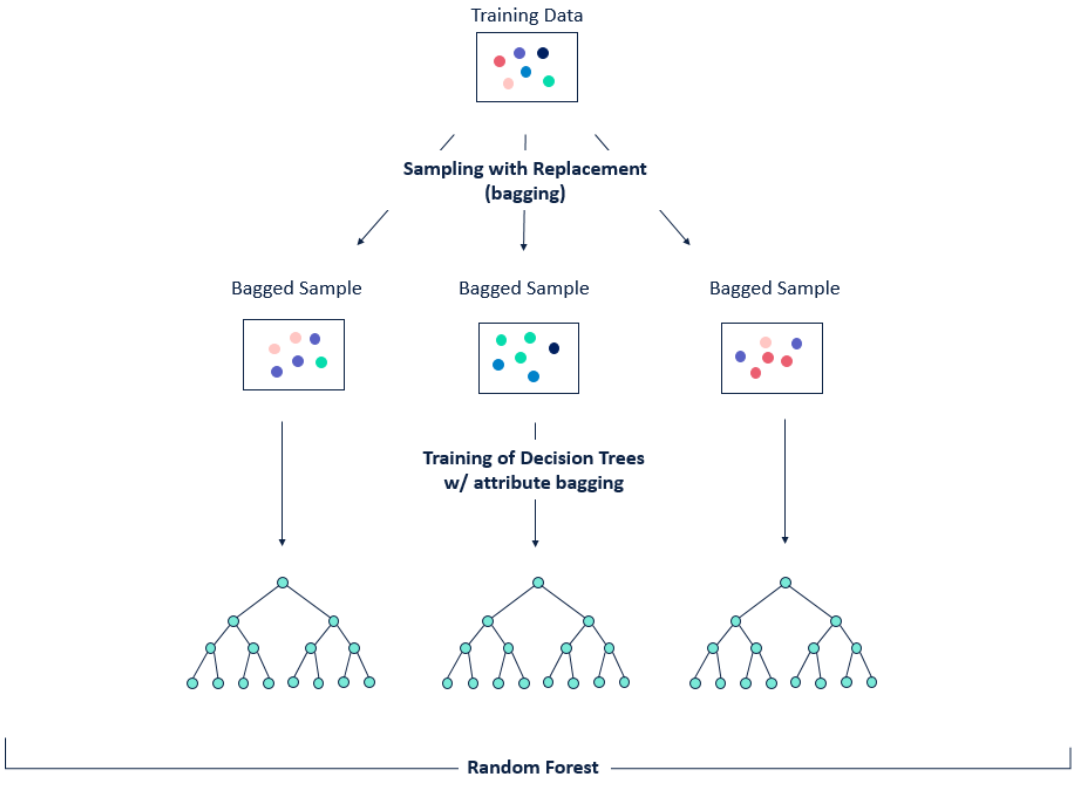
RANDOM FOREST

Random Forest (RF)

Random Forest:
회귀와 분류가 가능한 앙상블 모델 중 하나
(Bagging 계열에 속하는 알고리즘)

장점
- 단일 트리의 과대적합 생성의 단점 해소
- 특별히 매개변수 튜닝 없어도 예측 가능

단점
- 단일 트리에 비해 예측과정 시각화 힘듦
- 대량의 데이터셋에서 훈련&예측이 느림
- 차원이 높은 데이터 셋에서 작동이 힘듦

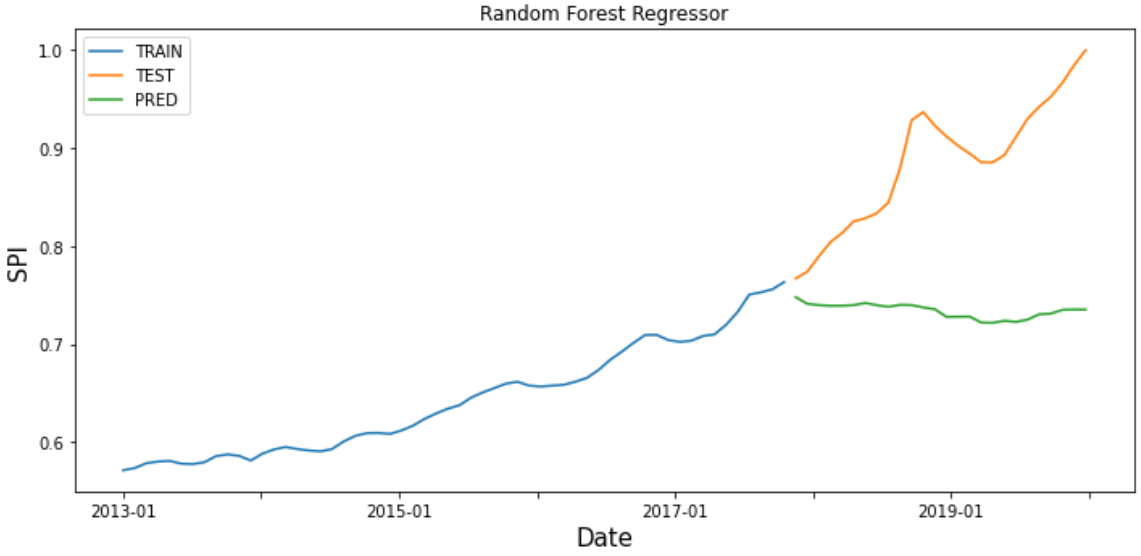


Random Forest (RF)

n_estimators
criterion
max_depth
min_samples_split
min_samples_leaf
min_weight_fraction_leaf
max_features
max_leaf_nodes
min_impurity_decrease
min_impurity_split
bootstrap
oob_score
n_jobs
random_state
verbose
warm_start
ccp_alpha
max_samples

파라미터명	설명
n_estimators	트리의 개수 지정 기본값은 10
criterion	데이터 분할의 퀄리티 측정 (mse, mae)
n_jobs	병렬로 실행할 작업의 개수, -1 은 모든 프로세서 사용
random_state	샘플의 무작위성 통제

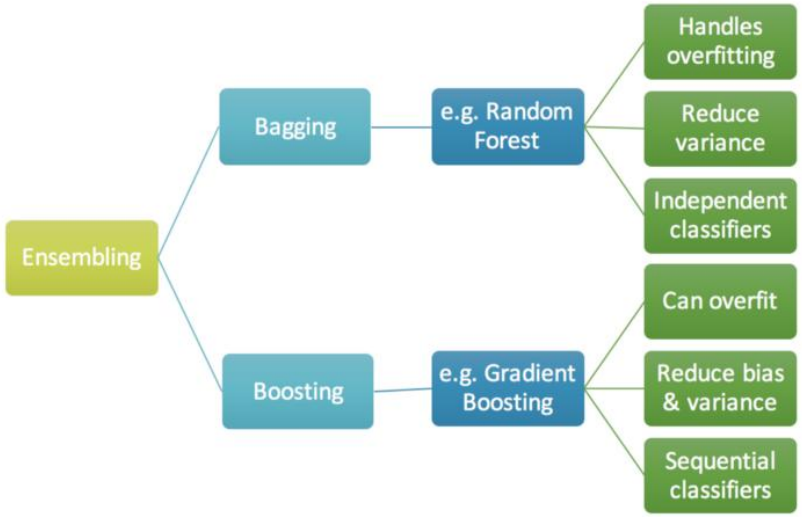
Random Forest (RF)

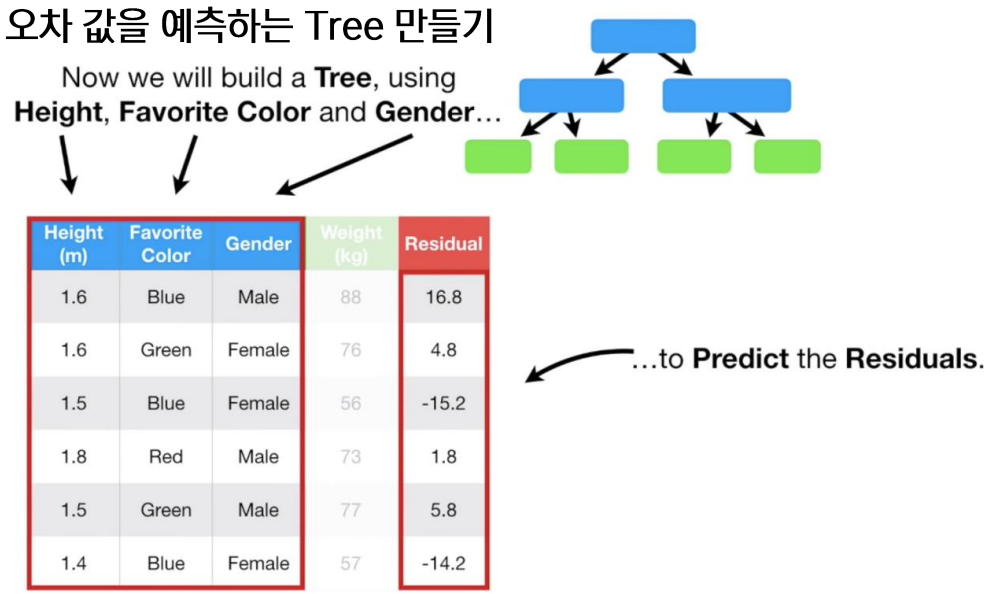


	MAE	RMSE
기간 I	3.6915	4.4457
기간 II	19.8820	21.7705

GRADIENT BOOSTING

GB 알고리즘 이해





Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Gradient Boosting Machine은 회귀 분석 또는 분류 분석을 수행할 수 있는 예측모형으로, Ensemble 방법론 중 **Boosting** 계열에 속한다.

Random Forest와 같이 기본적으로 Decision Tree를 기반으로 Ensemble하는데 차이점은 Gradient Boosting은 무작위성(Random)이 없이 사전에 강력한 가지치기를 통해 **이전 트리의 오차를 보완하는 방식**으로 순차적으로 트리를 생성한다는 것이다.

GB 알고리즘 이해

Boosting

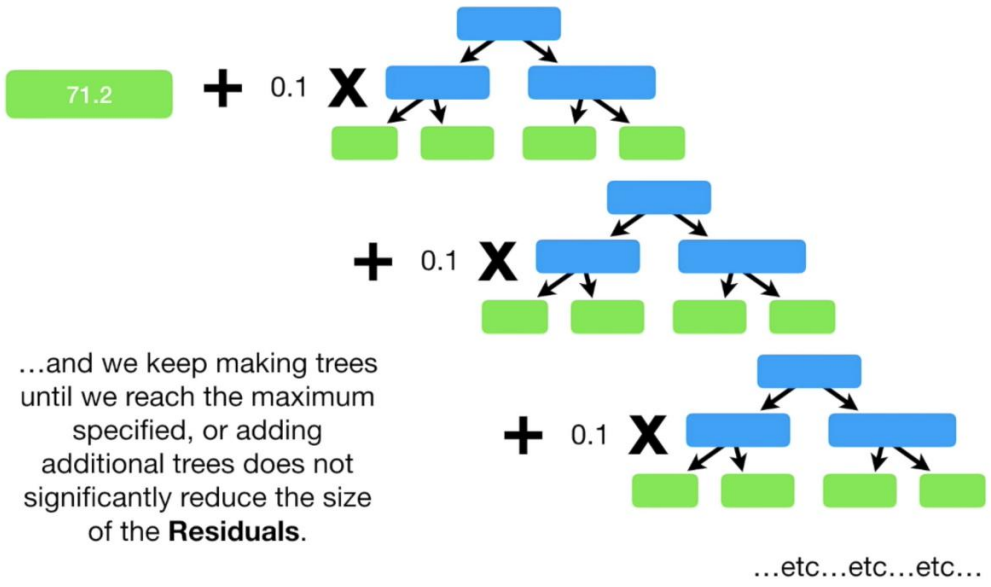
“Residual fitting” 이전 round의 잔차를 예측하는 함수

Gradient Boosting

Gradient descent 과정을 현재까지 진행된 모델 함수에 적용하여 오차를 줄여나가는 방법

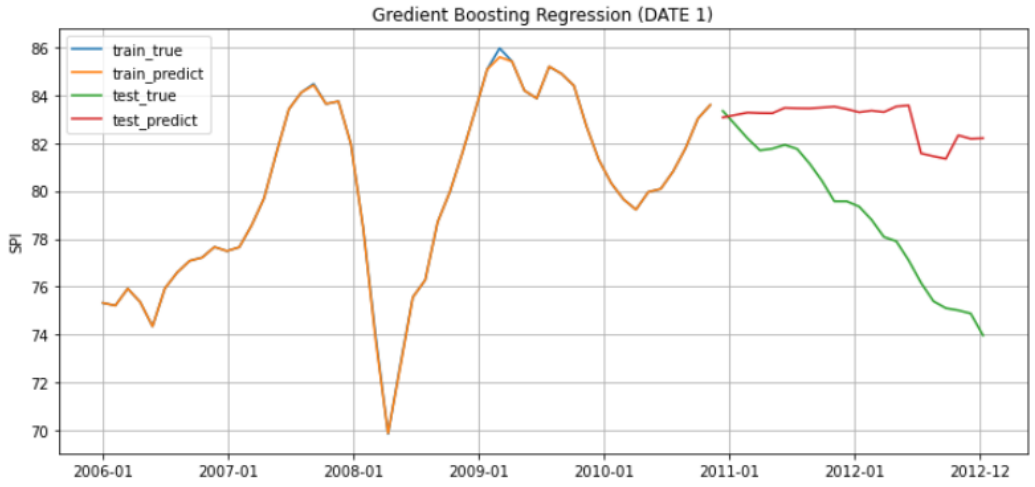
GBR 모델 주요 파라미터 설명

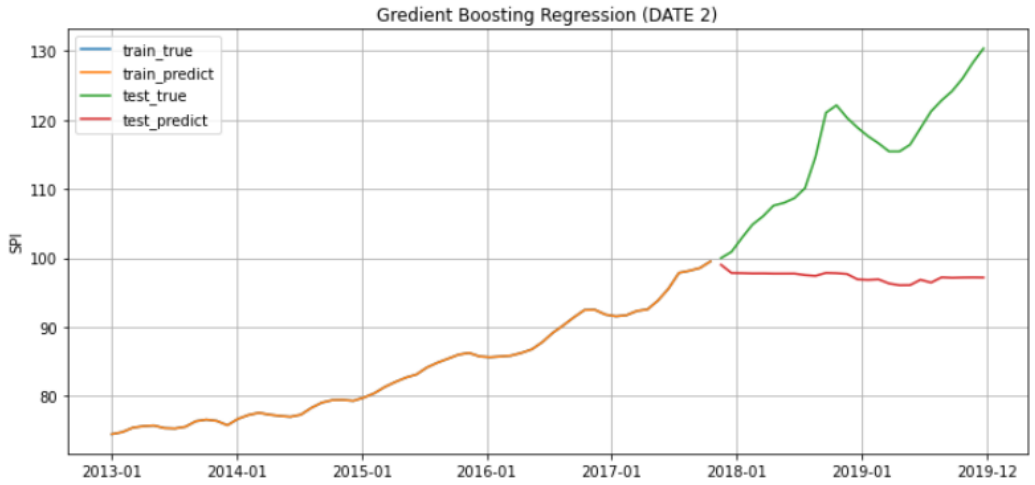
1. loss : 손실함수
2. learning_rate : Weak learner가 순차적으로 오류 값을 보정해 나가는 데 적용하는 계수
3. n_estimators : weak learner의 개수. 개수가 많을수록 예측 성능이 일정 수준까지 좋아지나 시간이 오래 걸림
4. min_samples_leaf : 리프노드(leaf node)가 되기 위한 최소한의 샘플 데이터 수



GB Regression

	기간 I	기간 II
	GridSearchCV를 통한 Best 파라미터 출력 ** 공통 조건 : KFold(n_splits=10)	
파라미터	'learning_rate': 0.1 'loss': 'lad' 'max_depth': 5 'min_samples_leaf': 2 'n_estimators': 500	'learning_rate': 0.1 'loss': 'ls' 'max_depth': 2 'min_samples_leaf': 3 'n_estimators': 500
MAE	3.9837	18.0640
RMSE	4.6407	20.0299



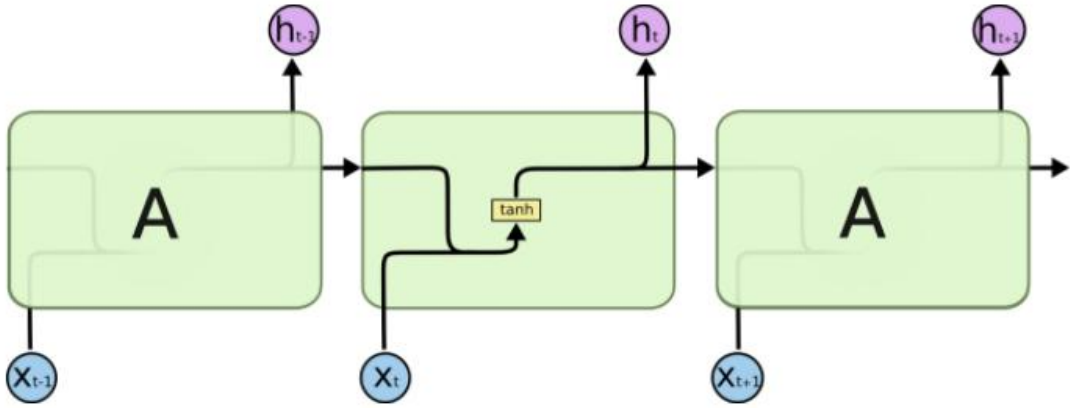


LSTM

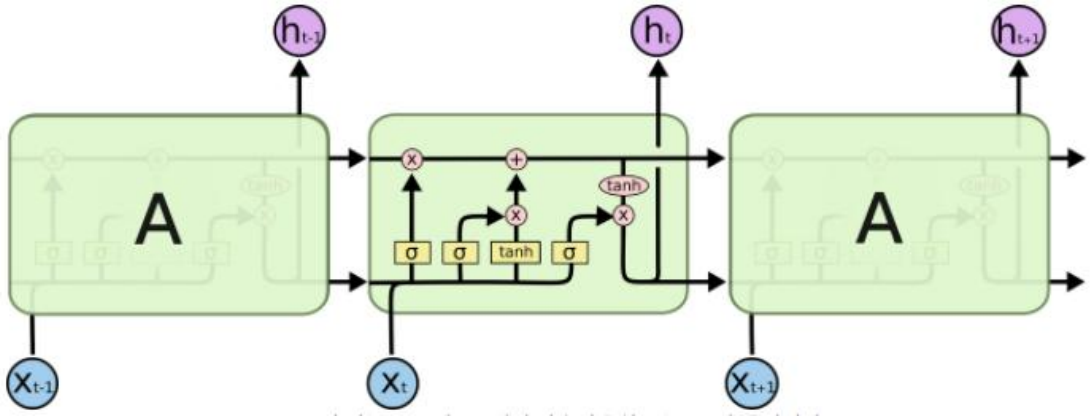
배경

LSTM은 무엇인가 ?

- LSTM은 기존 RNN의 변형 버전으로서, RNN의 **아주 긴 시퀀스 데이터**에서 역전파시 발생하는 **그레디언트 소실 문제(vanishing gradient problem)을 보완**하기 위해 등장
- RNN은 **순환 신경망**으로서 스스로를 반복하며 **이전 단계에서 얻은 정보를 지속하며 짧은 기간**에서는 **좋은 성능**을 보이지만 **기간이 길어지면 정보를 이어가지 못해** 성능이 좋지 않음
- LSTM은 RNN의 hidden state에 **cell state(장기 기억상태)**를 추가한 구조로서 먼 기간의 정보를 기억하며 긴 의존 기간을 필요로 하는 학습 수행 능력을 가짐



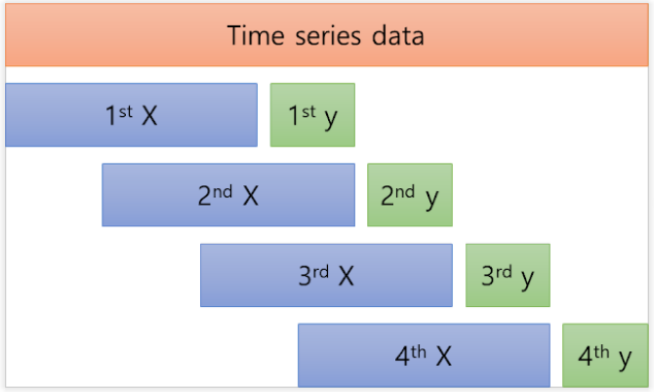
RNN

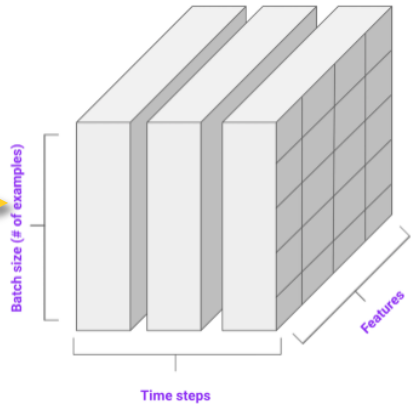


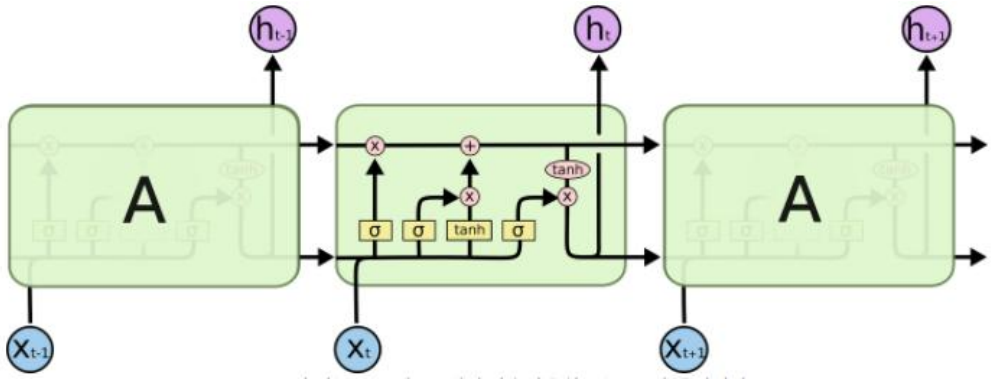
LSTM

배경

다변량 총 22개 시계열 데이터에서의 LSTM 적용







시계열 데이터를 LSTM에 적용할 경우
Window를 생성하여 timestep을 만들어야함
1. 다변량 t-12~t-1(X), t(y) 시점 생성
2. 다변량 t-12~t-1(X), t~t11(y) 시점 생성

LSTM은 3D 텐서를 input으로 넣어줘야 하므로 (배치 크기, 타임스텝수, 변수량) 순서로 input값 reshape

LSTM 모델 생성하여 하이퍼 파라미터 조정

LSTM

다변량 (t−12~t−1)시점으로 t시점 예측

다변량 시계열 LSTM 기법을 활용하여 t-12~t-1시점 으로 t시점 예측

```
reframed = series_to_supervised(scaled, 12,1)
#t-12~t데이터를 한 행으로 두며 윈도우를 생성한다. (각 변수의 시점을 t-12부터 t까지)
```

```
reframed.head()
```

	var1(t-12)	var2(t-12)	var3(t-12)	var4(t-12)	var5(t-12)	var6(t-12)	var7(t-12)	var8(t-12)	var9(t-12)	var10(t-12)	...	var14(t)
12	0.328068	0.000000	0.000000	0.000000	0.308060	0.118133	0.607143	0.185287	0.693370	0.000000	...	0.186230
13	0.387857	0.069949	0.136390	0.019048	0.512210	0.066116	0.583333	0.274416	0.701657	0.011721	...	0.266655
14	0.384343	0.155257	0.100221	0.059364	0.915984	0.113272	0.535714	0.179262	0.698895	0.005795	...	0.296410
15	0.476220	0.334567	0.334246	0.045527	1.000000	0.112299	0.595238	0.223199	0.701657	0.012697	...	0.317764
16	0.581335	0.413174	0.438320	0.073445	0.901895	0.121536	0.658730	0.187798	0.726519	0.008181	...	0.347260

5 rows × 299 columns

```
#각 데이터셋의 차원 확인
print(train_X.shape,valid_X.shape,test_X.shape)
print(train_y.shape,valid_y.shape,test_y.shape)
```

```
(23, 1, 298) (15, 1, 298) (22, 1, 298)
(23, 1) (15, 1) (22, 1)
```

기간 I

```
#각 데이터셋의 차원 확인
print(train_X.shape,valid_X.shape,test_X.shape)
print(train_y.shape,valid_y.shape,test_y.shape)
```

```
(29, 1, 298) (18, 1, 298) (25, 1, 298)
(29, 1) (18, 1) (25, 1)
```

기간 II

1. Data import (data1, data2)

2. Data split (train, valid, test)

3. Data Scaling (MinMaxScaler)

4. 데이터 병합 후 window 생성 (timestep을 t−12부터 t까지)

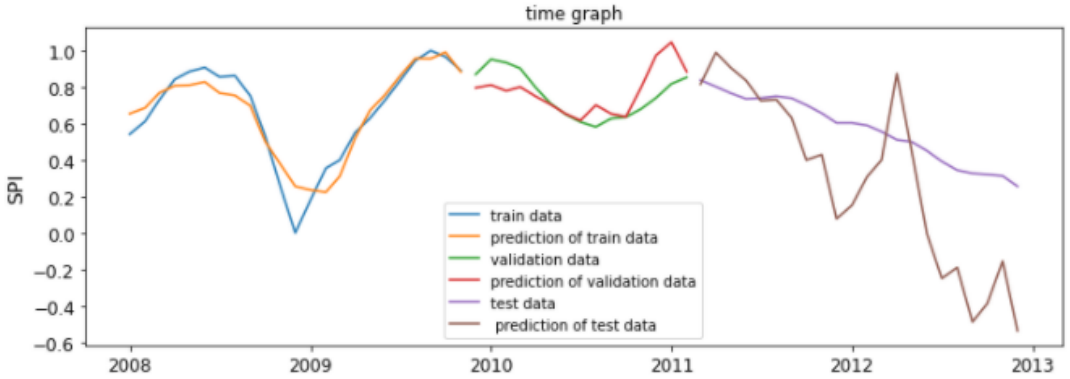
5. 독립변수(t−12~t−1시점의 22개의 변수)를 X로 종속변수 (t시점의 실거래가 지수)를 y로 split

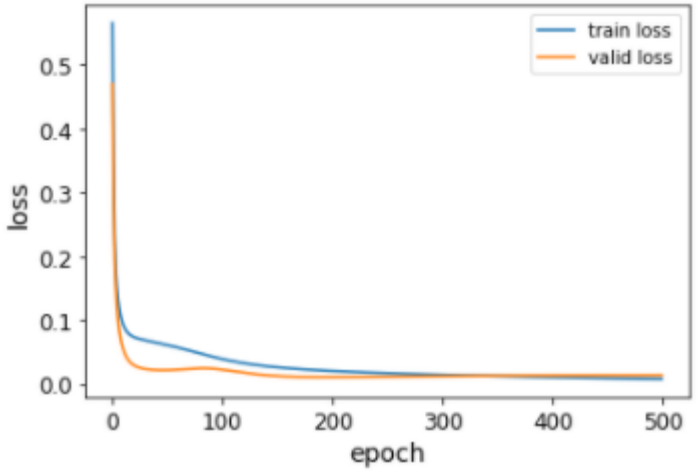
6. Input값을 3차원 텐서로 변환

LSTM

출력 결과 – 기간 I

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Layer	3	2	1	1	1	1	5
Units	20	20	20	10	20	8	5
Optimizer	Adam	Adam	Adam	Sgd	Sgd	Sgd	Adam
Epochs	1000	1000	1000	300	800	500	2000
Val_MSE	0.0206	0.0357	0.0168	0.0080	0.0163	0.0133	0.0147
Val_MAE	0.1281	0.1486	0.0974	0.0763	0.1023	0.0865	0.0937
Val_RMSE	0.1435	0.1889	0.1296	0.0894	0.1276	0.1153	0.1212

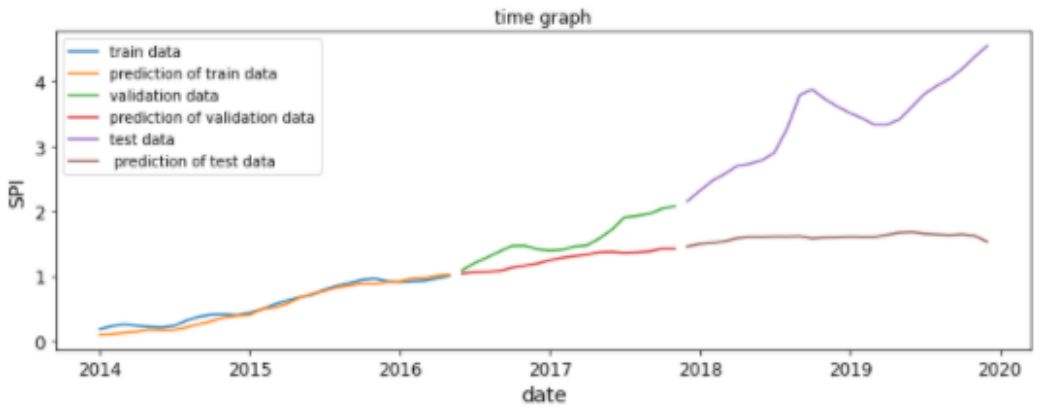


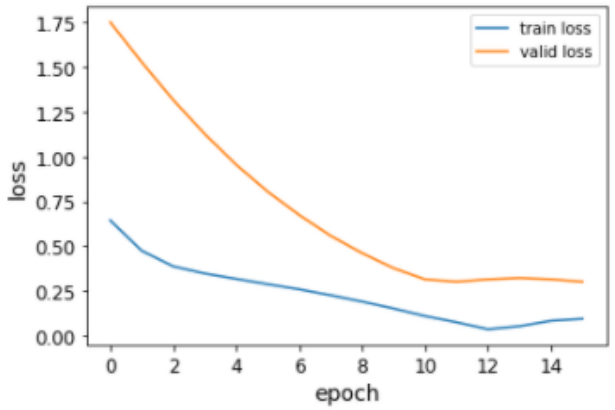


LSTM

출력 결과 – 기간 II

	Model 1	Model 2	Model 3	Model 4
Layer	1	1	1	1
Units	150	150	100	15
Optimizer	Adam	Adam	Adam	Adam
Epochs	1000	1000	300	1000
Val_MSE	0.0208	0.0158	0.0160	0.1273
Val_MAE	0.1271	0.1131	0.1274	0.3011
Val_RMSE	0.1442	0.1256	0.1256	0.3567





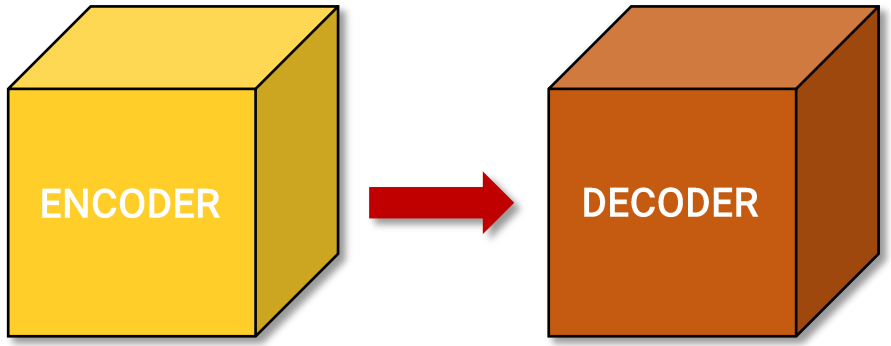
DUAL ATTENTION MECHANISM

배경

Attention Mechanism이란

- 어텐션 메커니즘을 사용한 모델은 출력을 만들기 위해 해당 시점에서 예측해야할 단어와 연관이 있는 입력 단어 부분을 집중(attention)해서 가중치를 부여

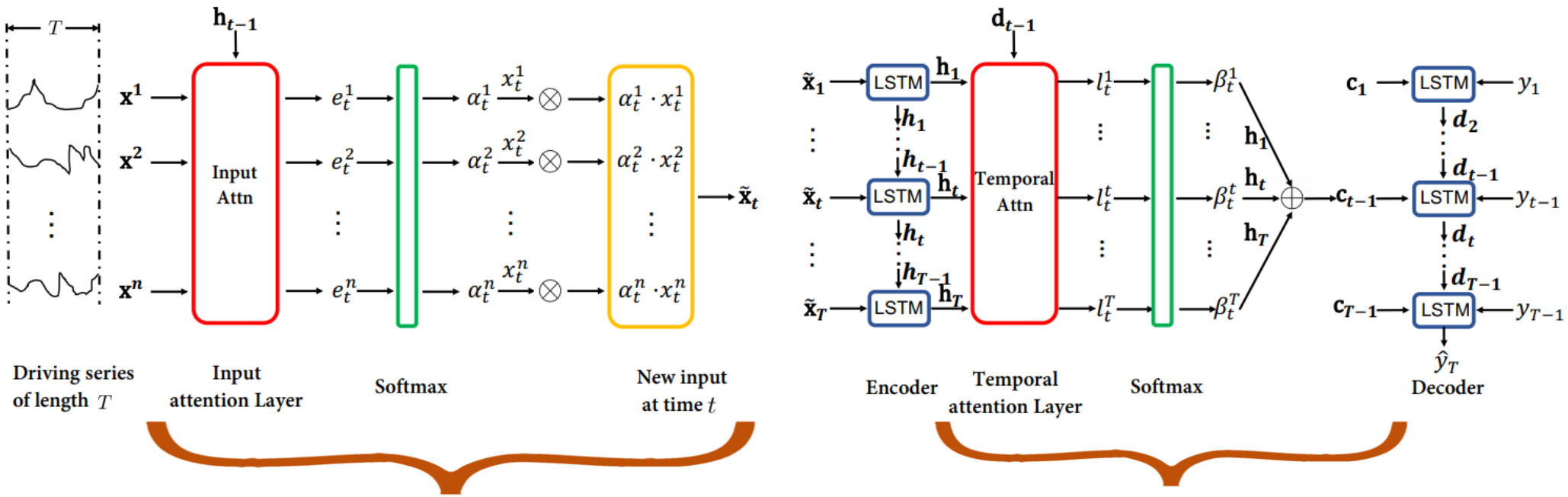
- 최근 LSTM 기법만을 사용한 연구에서, 낮은 상관관계를 가지고 있는 인자들로 생기는 성능저하의 문제 발견



다음 예측값에 관련 있는 외생변수들을 판단

모든 시간간격에 대해서 관련성이 있는 변수들에 대한 encoder의 hidden states를 선택

DUAL ATTENTION MODEL



다음 예측값 관련 외생변수 판단

모든 시간간격에 대해서
관련성 있는 변수들의 가중합을
참고해서 context vector 출력

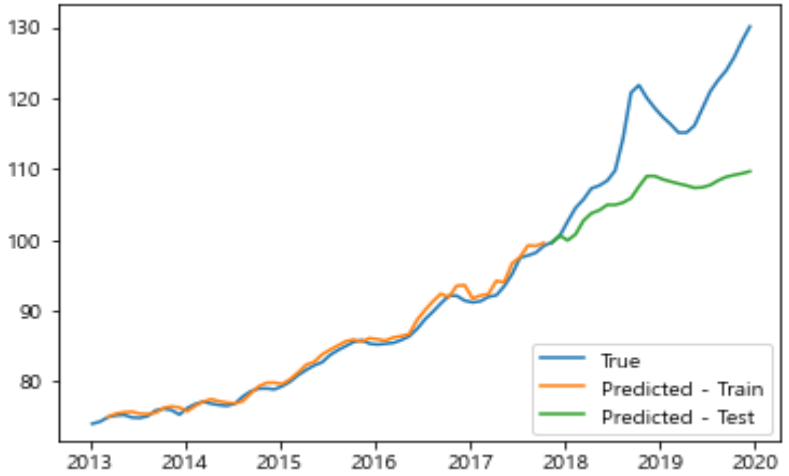
DUAL ATTENTION MODEL

Batch = {64,128}
Encoder = {64,128,256}
Decoder = {64,128,256}
Ntime = {3,5,10}
LR = {0.1,0.001,0.003,0.005,0.0001}
Epochs = {100,500,1000}

최종선택

Batch = 64
Encoder = 256
Decoder = 256
Ntime = 3
LR = 0.003
Epochs = 1000

	MAE	RMSE
기간 I	0.5946	0.5972
기간 II	0.5448	0.6674



결론

	기간 I		기간 II	
	RMSE	MAE	RMSE	MAE
SVR	1.9193	2.2809	4.1796	5.2946
Random Forest	3.6915	4.4457	19.8820	21.7705
Gradient Boosting	3.9837	4.6407	18.0640	20.0299
LSTM	0.0865	0.1153	0.3011	0.3567
Dual Attention	0.5972	0.5946	0.6674	0.5448

결론

- **시장이 급변하는 시기인 기간 1**의 경우 모든 모형들이 **비교적 유사하게 시장 추세를 예측**하는 반면, **안정적인 시장인 기간 2**의 경우 **앙상블 기법들을 제외한 모든 모형들이 시장 추세를 예측**할 수 있었다

- 일부 모형의 경우 예측력이 우수한 것으로 나타나고 있는데 그래프를 보면 **시계열분석 모형의 예측값이 실제 시장 추세와는 전혀 다른 양상을 보이고 있어 MAE 및 RMSE를 통한 예측력 비교에 큰 의미를 둘 수 없다**

- 최근 시계열 분석에서 사용할 수 있다는 **Attention Mechanism**은 RF, GRBT, SVR에 비해서는 비교적 유사하게 시장 추세를 예측하지만 **LSTM보단 우수하지 못했다**

- 일부 머신 러닝 방법의 경우 **다변량 변수를 적용한 모형보다 단변량 변수를 적용한 모형의 예측력이 더 우수**한것으로 나왔다

시사점 및 한계

- 본 프로젝트는 시계열분석 방법론을 비교한 연구로서 **분석자료, 변수 설정에 따라 분석 결과**가 달라질 수 있기 때문에 특정 방법이 우수하다고 단정하기에는 무리가 있으며 이에 대해서 **추가적인 연구가 필요하다**

- 딥러닝 **hyperparameter 설정은 연구자의 경험과 데이터 이해도의 강한 영향**을 받기 때문에 연구자들의 높은 역량을 요구한다

- 또한 **머신 러닝 방법은 모형을 최적화하기 위한 명확한 기준**이 없다는 점에서 적용 변수에 따라 결과가 달라진다

- 기존의 모형은 과거의 자료에 기반하기 때문에 **현재의 가격만을 예상하는 것이 최선이고** 시계열 분석의 경우 **자료의 기간이 짧기 때문에 분석의 정확도가 떨어지게 된다**

감사합니다

UNICON