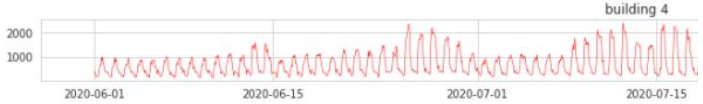
ML팀
시계열 발표

목차

1. 시계열이란 무엇인지?
2. 시계열 데이터의 특징
3. 시계열을 나타내는 특성들
4. 시계열 모델에 대해
    a. 통계적 모델
    b. 딥러닝 모델

1. 시계열 데이터란?



건물의 전력사용량

시계열 데이터란 시간 순서대로 정렬된 데이터를 뜻함.

연속된 데이터는 서로 상관성이 존재.

사용되는 영역

많은 도메인에서 시계열 데이터를 볼 수 있음.

시계열 분석은 복잡한 기록 관리 시스템을 요구함. 현대 정부와 기업, 과학 기반 시설이 장 시간 고품질의 데이터를 계속 수집 가능하게 되면서 시계열 분석이 발전.

예) 산업 (Amazon Forecast의 데이터 세트 도메인- 소매 수요 예측, 공급망 및 재고 계획 수립), 의료 (애플 건강앱-운동성 데이터, 심박수 데이터, 건강 변화 추세 '자신의 건강에 대해 더 많이 알수록 적절히 대처할 수 있는 힘도 더 커지는 법'), 자원(신재생에너지 발전량 예측 및 분석) , 건축, 일기예보, 경제성장 예측(경제 호황과 불황 주기 등 방지, 미래 시장 예측) , 천문학 등

애플 건강앱

2. 시계열 데이터의 특징

Date	High	Low	Open	Close	Volume	Adj Close
2009-12-31	30.478571	30.080000	30.447144	30.104286	88102700.0	20.159719
2010-01-04	30.642857	30.340000	30.490000	30.572857	123432400.0	20.473503
2010-01-05	30.798571	30.464285	30.657143	30.625713	150476200.0	20.508902
2010-01-06	30.747143	30.107143	30.625713	30.138571	138040000.0	20.182680
2010-01-07	30.285715	29.864286	30.250000	30.082857	119282800.0	20.145369

시계열 데이터란? 일정한 시간동안 수집 된 일련의 순차적으로 정해진 데이터 셋의 집합

시계열 데이터의 특징? 시간에 관해 순서가 매겨져 있다는 점, 연속한 관측치는 서로 상관관계를 갖고 있다는 점

어떤게 시계열 데이터? Time stamp 즉, Datetime 이 있는 데이터

표 2-4 체중 감량 앱의 식사 일기 예

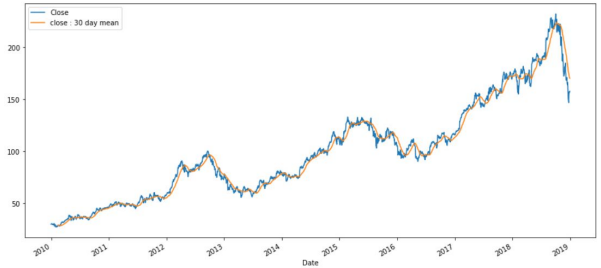
시간	섭취 음식
Mon, April 7, 11:14:32	팬케이크
Mon, April 7, 11:14:32	샌드위치
Mon, April 7, 11:14:32	피자

Time stamp만 있다면 무조건 시계열 데이터? NO!

- 연속적인 패턴(상관관계)의 존재 유무
- time stamp만의 또 다른 문제점

그럼 Time stamp가 없어도 시계열 데이터로 접근 가능? Hmm..

- 도메인 지식을 활용
- 연속적인 패턴(상관관계) 확인

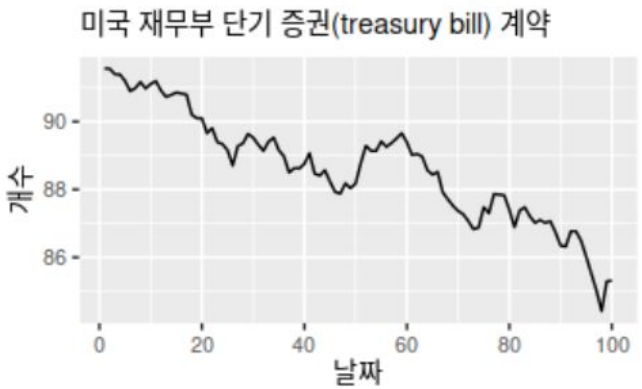


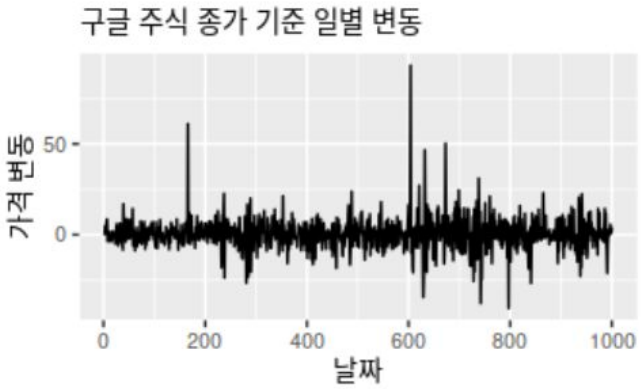
시계열 패턴

추세

계절성

주기성

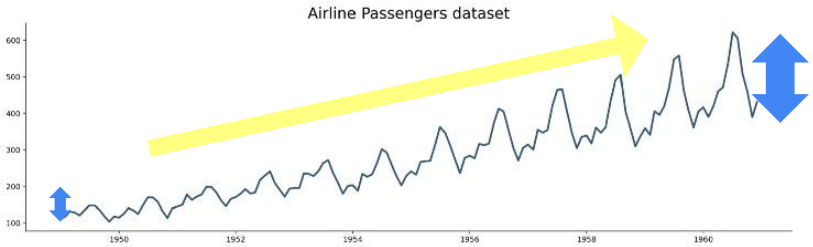




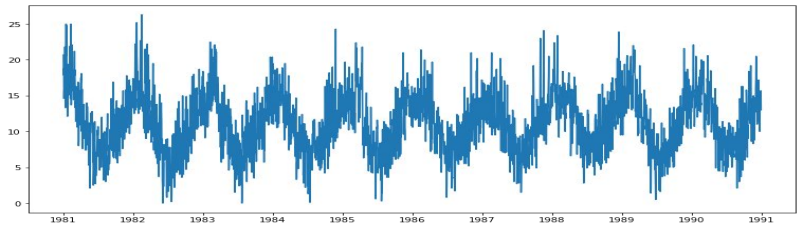
3. 시계열 데이터의 특성 - 정상성

정상성
Stationarity

시간에 따라 통계적 특성(평균, 분산 등)이 변하지 않음

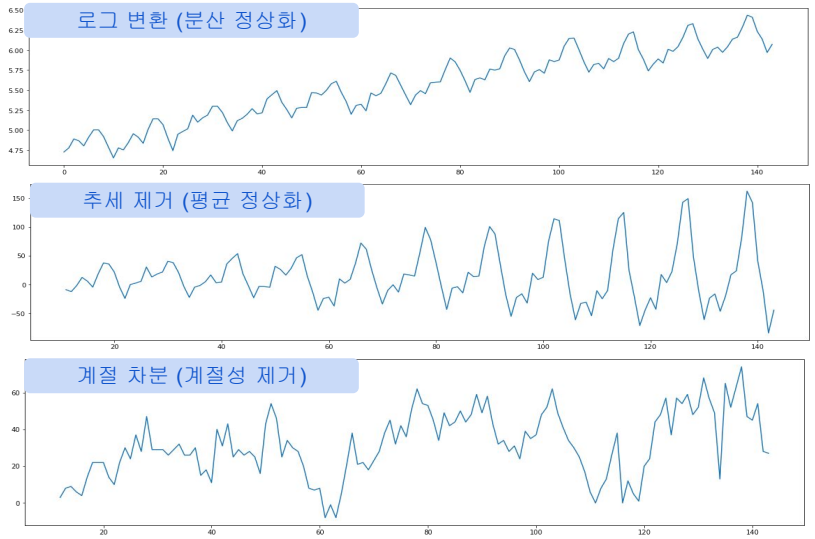


비정상 시계열의 예 - 평균이 증가, 변동폭 증가, 계절성



비정상 시계열의 예 - 계절성

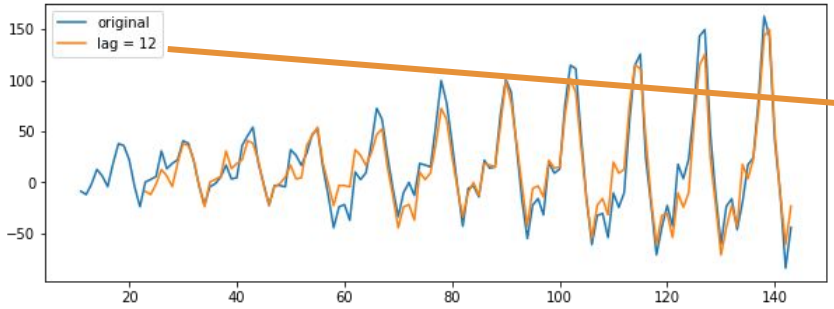
비정상 시계열을 정상 시계열로 변환해
분석과 모델 적용이 잘 되도록 만든다!

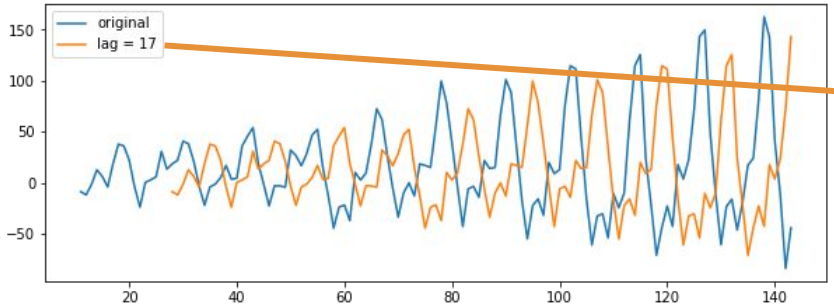


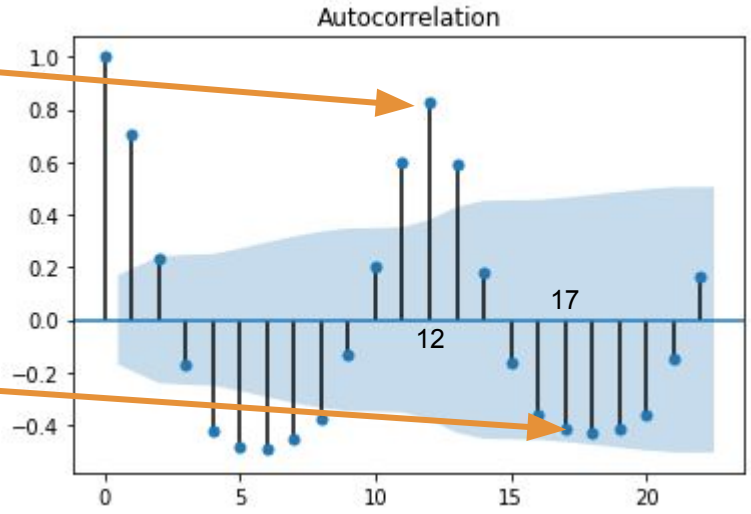
3. 시계열 데이터의 특성 - 자기상관

자기상관
Autocorrelation

(다른 시간대의) 자기 자신과의 상관관계







* 가로축: 데이터를 shift한 칸 수

시계열 모델

통계적 모델과 딥러닝 모델이 있습니다

통계 모델:

- AR, MA, ARMA, ARIMA, VAR

딥러닝 모델:

- RNN, GRU, LSTM

통계 모델은 데이터 수가 적을 때 좀 유리합니다.

데이터 수 많아지면 딥러닝 모델의 압승!

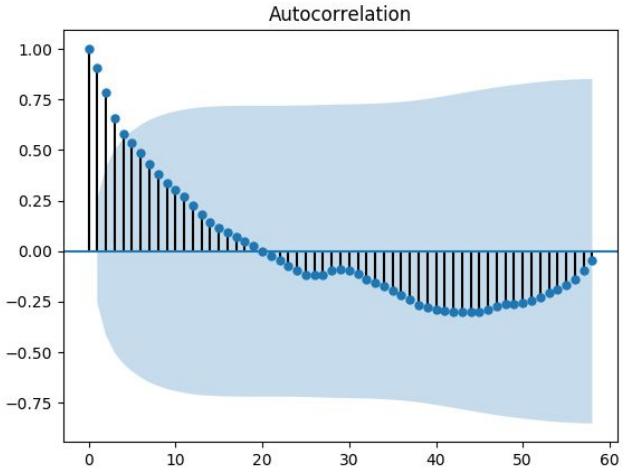
↑ (TMI) 이 대회에선 통계 모델이 1등을 차지하기도 했어요!

1. ACF & AR

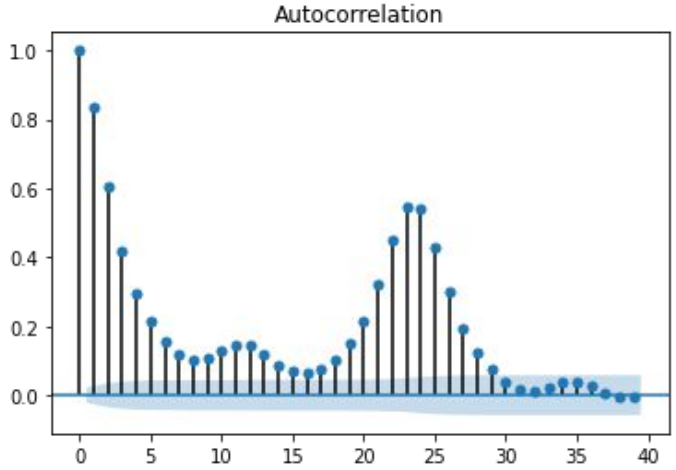
ACF 그래프는 시계열 데이터간 상관관계를 보여주는 그래프. (이때 0 시점과 t시점간 ACF는 1, 2, 3, …. , t-1 시점에 영향을 받음)

그리고 이 그래프를 통해 AR모델의 파라미터를 추정(큰 폭으로 감소하는 구간)

꾸준한 감소세 -> p의 값을 0으로 추정



3~5 지점에서 큰 감소세를 띔 -> p = 3~5중 선택

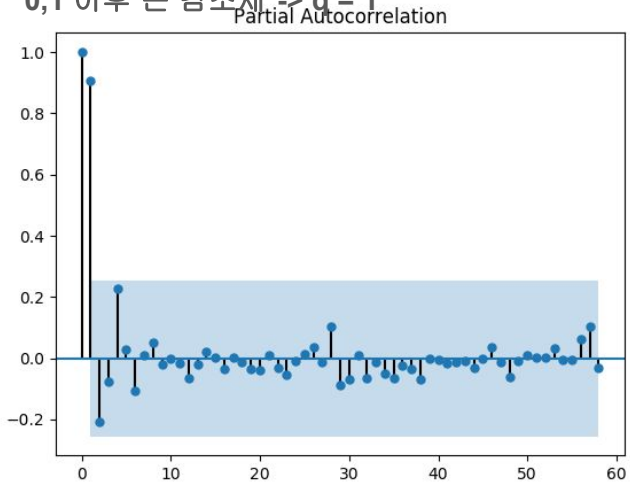


2. PACF & MA

PACF 그래프는 시계열 데이터간 순수상관관계를 보여주는 그래프. **(ACF**와는 달리 특정 두 지점이 다른 지점들에 영향을 받지 않음**)**

그리고 이 그래프를 통해 **MA**모델의 파라미터를 추정**(**큰 폭으로 감소하는 구간**)**

0,1 이후 큰 감소세 **-> q = 1**



3. ARIMA

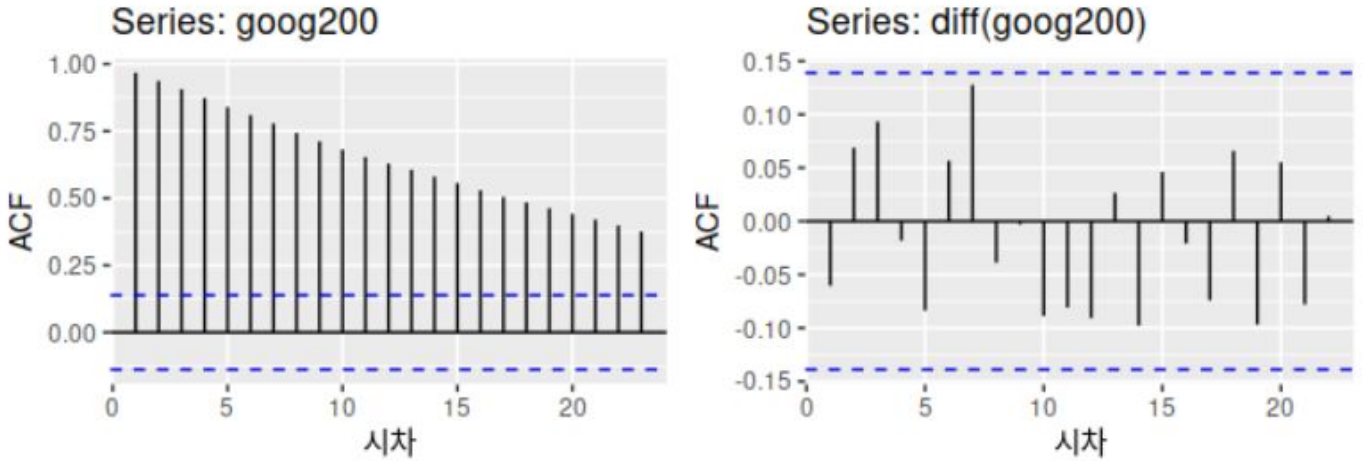
ARIMA(p,d,q) 모형은 d차 차분한 데이터에 위 AR(p) 모형과 MA(q) 모형을 합친 모형으로, 식은 다음과 같다.

$$y_t' = c + \phi_1 y_{t-1}' + \phi_2 y_{t-2}' + \ldots + \phi_p y_{t-p}' + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

단, y'은 d차 차분을 구한 시계열, p는 자기회귀 부분의 차수, d는 차분 회수, q는 이동평균 부분의 차수이다.

AR(p)모형과 ARIMA(p,0,0)모형은 같은 모형이며, MA(q)모형과 ARIMA(0,0,q) 모형은 같은 모형이다.

3. 차분



왼쪽의 시계열 데이터의 ACF는 비정상적인 시계열 데이터임을 보여주는데,
차분을 한 오른쪽 ACF그래프는 비교적 정상적인 형태를 보여주고 있음

4. Deep learning

LSTM

LSTM (Long Short Term Memory)는 기존의 RNN이 출력과 먼 위치에 있는 정보를 기억할 수 없다는 단점을 보완하여 장/단기 기억을 가능하게 설계한 신경망의 구조를 말합니다. 주로 시계열 처리나, 자연어 처리에 사용됩니다.

LSTM vs ARIMA 모델

=>전통 시계열 예측 모델인 ARIMA와 딥러닝 기반 LSTM의 비교

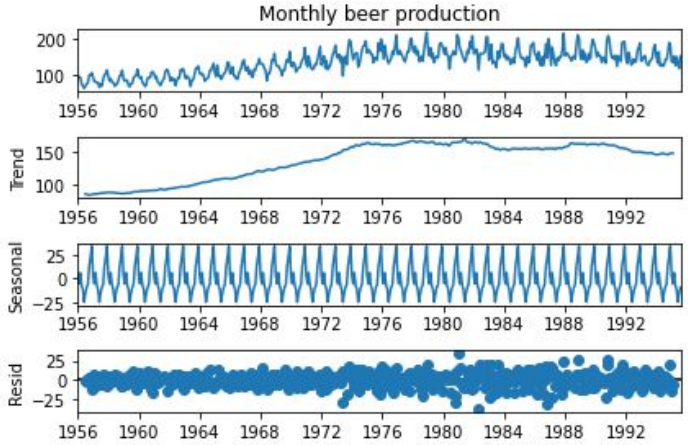
=>당연히, 딥러닝이 가지는 장점(풍부한 파라미터수와 높은 학습력)으로 인해 생산성이 떨어지는 비교이긴함 (당연히 데이터가 풍부할때는 LSTM이 훨씬 높은 성능을 보임

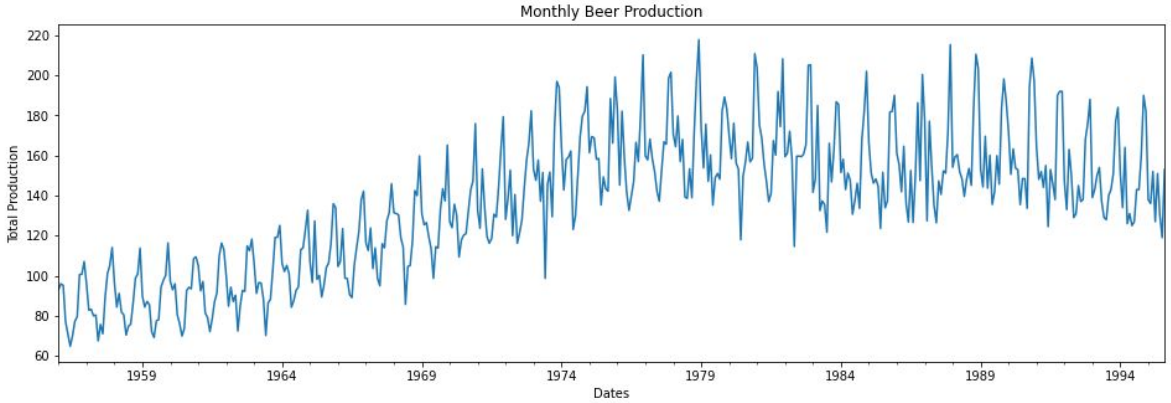
=>그럼에도, 간단한 데이터(정상성이 잘 유지되는 단순한 feature의 데이터)의 경우는 어떠한 차이를 보일까를 분석해봄

4. Deep learning

LSTM 쓴다!

Month	Monthly beer production
1956-01-01	93.2
1956-02-01	96.0
1956-03-01	95.2
1956-04-01	77.1
1956-05-01	70.9





4. Deep learning

LSTM 쓴다!

```
from keras.preprocessing.sequence import TimeseriesGenerator

n_input = 12
n_features= 1
generator = TimeseriesGenerator(scaled_train_data, scaled_train_data, length=n_input, batch_size=1)
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM

lstm_model = Sequential()
lstm_model.add(LSTM(200, activation='relu', input_shape=(n_input, n_features)))
lstm_model.add(Dense(1))
lstm_model.compile(optimizer='adam', loss='mse')

lstm_model.summary()
```

```
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 200)               161600
_________________________________________________________________
dense (Dense)                (None, 1)                 201
=================================================================
Total params: 161,801
Trainable params: 161,801
Non-trainable params: 0
_________________________________________________________________
```

```
Epoch 1/20
452/452 [==============================] - 23s 47ms/step - loss: 0.0238
Epoch 2/20
452/452 [==============================] - 21s 46ms/step - loss: 0.0147
Epoch 3/20
452/452 [==============================] - 21s 47ms/step - loss: 0.0093
Epoch 4/20
452/452 [==============================] - 21s 47ms/step - loss: 0.0079
Epoch 5/20
452/452 [==============================] - 21s 46ms/step - loss: 0.0087
Epoch 6/20
452/452 [==============================] - 21s 46ms/step - loss: 0.0075
Epoch 7/20
130/452 [=======>......................] - ETA: 15s - loss: 0.0072
```

LSTM기반

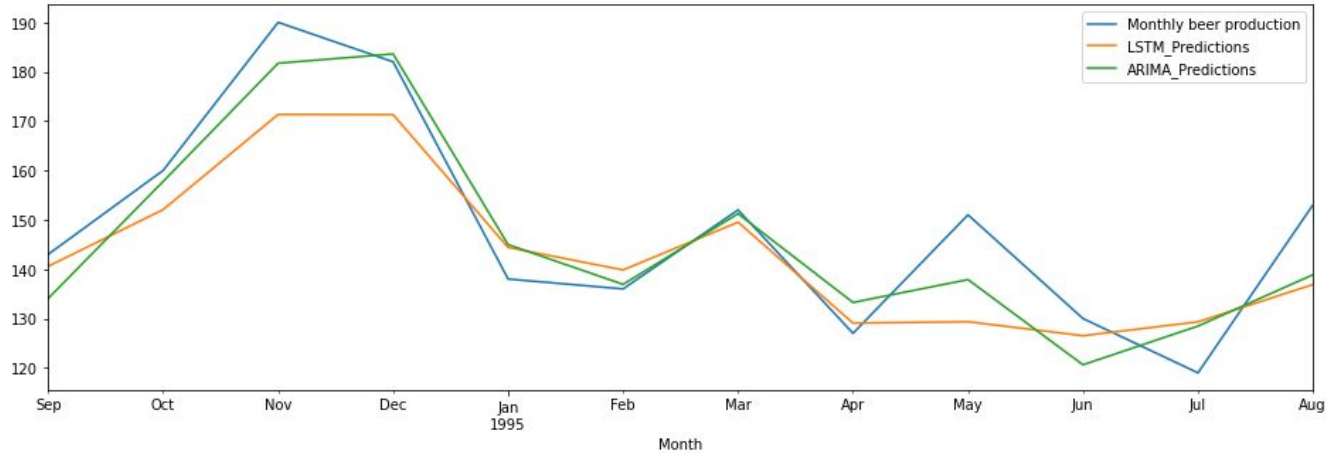
12개월 후

예측 모델

=>Beer 예측

4. Deep learning

LSTM 쓴다!



LSTM기반

12개월 후
예측 모델

LSTM 과 ARIMA를 비교

=>적은 파라미터, 적은 데이터에서는 ARIMA가 효율성이 높다