




KU-BIG

의료데이터분석 스터디

:Time Series Forecasting

박재찬, 조윤경



Summarization of Forecasts using different Models

	Model Name	Root Mean Squared Error
8	SARIMA Model	16237.397606
9	Facebook's Prophet Model	18274.936691
4	Holt's Winter Model	29173.730614
7	ARIMA Model	30690.141269
3	Holt's Linear	31981.387657
5	Auto Regressive Model (AR)	41037.564773
6	Moving Average Model (MA)	42472.191701
1	Polynomial Regression	478411.931699
0	Linear Regression	6783550.068817
2	Support Vector Machine Regressor	7337154.623732

Time series data

Additive models

$$y_t = m_t + s_t + \epsilon_t$$

불규칙
Random

Multiplicative models

$$y_t = m_t \times s_t \times \epsilon_t$$

추세
trends

계절
seasonality

계절 패턴 크기가 데이터 크기에 따라 변동 $x \rightarrow$ additive

계절 패턴 크기가 데이터 크기에 따라 변동 \rightarrow multiplicative

Smoothing method

Time series data에 있는 무작위적인 변화로 인해 생기는 효과를 줄이기 위한 방법론

e.g. 시계열 자료에 평균을 취함 - 가장 단순한 평활법(smoothing method), 모든 과거 관측값을 동일한 가중치로 다루기 때문에 추세(trends) 반영 불가

→ 1) Moving average smoothing method

: trend=이동평균, seasonality=자료값/이동평균, random=관측값/예측값
(예측값=이동평균*seasonality)

→ 2) exponential smoothing method: 단순지수/이중지수(추세 추가)/삼중지수(추세, 계절성 추가)

Holt's Linear Model

추세있는 데이터 예측

linear exponential smoothing methods의 일종

이전 시점의 (과거) 관측값에 지수적으로 감소하는 가중치 곱하여 미래값 예측

Forecast equation

$$\hat{y}_{t+h|t} = \ell_t + hb_t$$

Level equation

$$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$$

Trend equation

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

ℓ_t 는 시간 t 에서 시계열의 수준 추정 값, b_t 는 시간 t 에서의 시계열의 추세(기울기) 추정 값, $0 \leq \alpha \leq 1$ 은 수준에 대한 매개변수, $0 \leq \beta^* \leq 1$ 은 추세에 대한 매개변수

→ 미래에도 계속 일정한 추세 가짐(증가/감소). 그러나 과도하게 예측하는 경향 O

Holt's Winter Model

계절성 파악 위해 Holt의 기법 확장

1. **Additive** Holt-Winter's

개별요인의 효과 구분하고 더하여 additive(가법적으로) 모형화

$$\hat{y}_{t+h|t} = \hat{\mu}_t + h \cdot b_t + s_{t+h-p}, \quad h \leq p$$

(p : 계절성의 주기)

<seasonality pattern 크기가 데이터의 크기에 따라 달라지지 않는 경우=변동폭이 일정>

2. **Multiplicative** Holt-Winter's

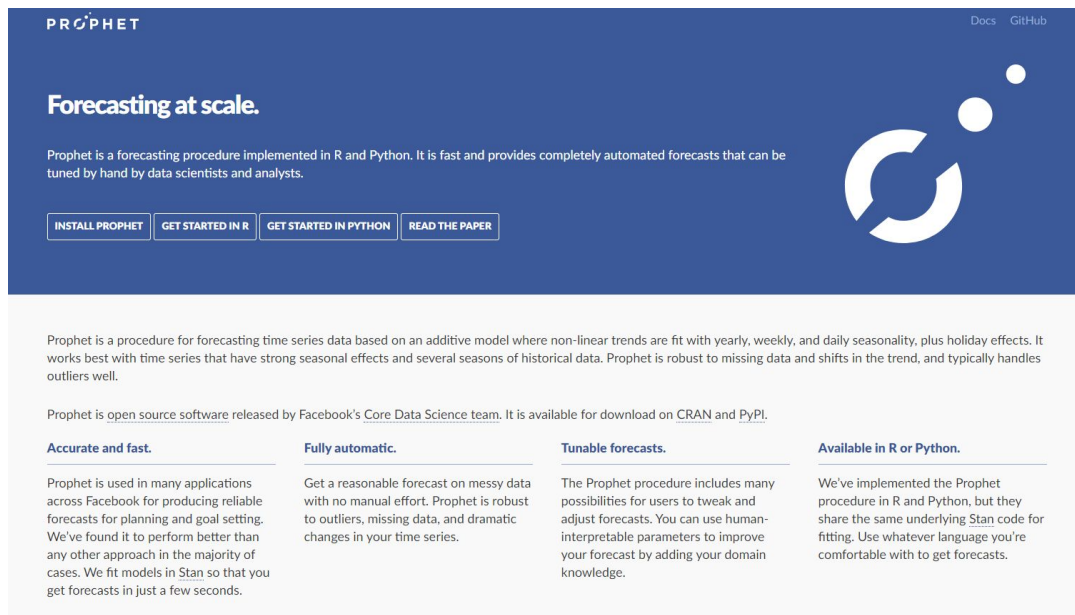
효과를 곱하여 모형화 - 데이터 증가하면 seasonality pattern도 증가한다고 가정(→ 대부분의 실제 time series data의 패턴과 동일)

$$\hat{y}_{t+h|t} = (\hat{\mu}_t + h \cdot b_t) \times s_{t+h-p}, \quad h \leq p$$

(p : 계절성의 주기)

<seasonality pattern 크기가 데이터 크기에 따라 달라지는 경우=변동폭이 시간에 따라 커지는 경우>

Facebook's Prophet Model

The image is a screenshot of the Facebook Prophet website. The header is dark blue with the word 'PROPHET' in white. Below the header, the text 'Forecasting at scale.' is displayed. A paragraph describes Prophet as a forecasting procedure implemented in R and Python, noting its speed and automation. Four buttons are visible: 'INSTALL PROPHET', 'GET STARTED IN R', 'GET STARTED IN PYTHON', and 'READ THE PAPER'. On the right side of the header, there are links for 'Docs' and 'GitHub'. A large white circular logo with a stylized 'P' is also present. Below the main content area, there is a detailed description of the Prophet model, its capabilities, and its availability as open source software. The bottom section is divided into four columns, each with a title and a brief description: 'Accurate and fast.', 'Fully automatic.', 'Tunable forecasts.', and 'Available in R or Python.'

- Python, R
- **'Fully automatic'**
- **curve-fitting** 방식(주어진 데이터 가장 적절히 표현할 수 있는 함수식을 계산 → ARIMA같은 모델과 달리 시간에 종속적이지 않음)
- **GAMs(Generalized additive modles)** 아이디어 활용
- **유연성** - 필요에 따라 새로운 요소 추가하는 것 용이 (e.g. 계절성의 새로운 원천 확인되면 prophet은 이를 고려하여 빠르게 적합시킴)

Facebook's Prophet Model

$$y(t) = g(t) + s(t) + h(t) + e_t$$

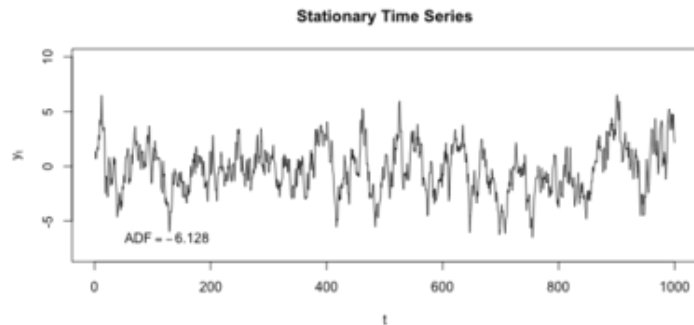
- $g(t)$ = 반복적인 요소를 가지지 않은 트렌드
- $s(t)$ = 요일 혹은 연 계절성과 같은 반복적 변화
- $h(t)$ = Holiday와 같이 가끔 불규칙하게 영향을 미치는 요
- e = 정규분포를 따르는 잔차

→ 시계열 모형, 방법론에 대한 지식이 거의 없는 비전문가들도 해당 '데이터'에 대한 도메인 지식만 충분하다면 쉽게 튜닝 가능하도록 직관적인 모수들로 고안된 식

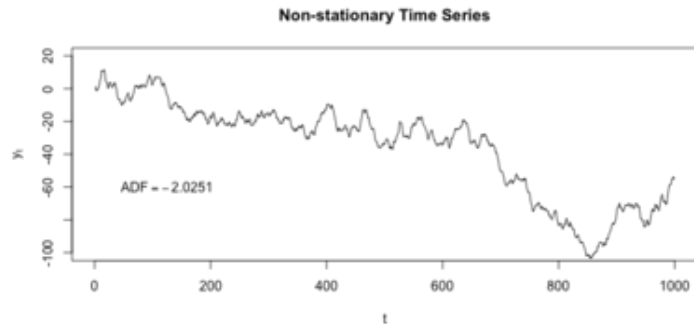
Stationarity

평균, 분산 등의 통계적인 특징이
시간에 따라 변하지 않는 시계열

Stationary



non
Stationary



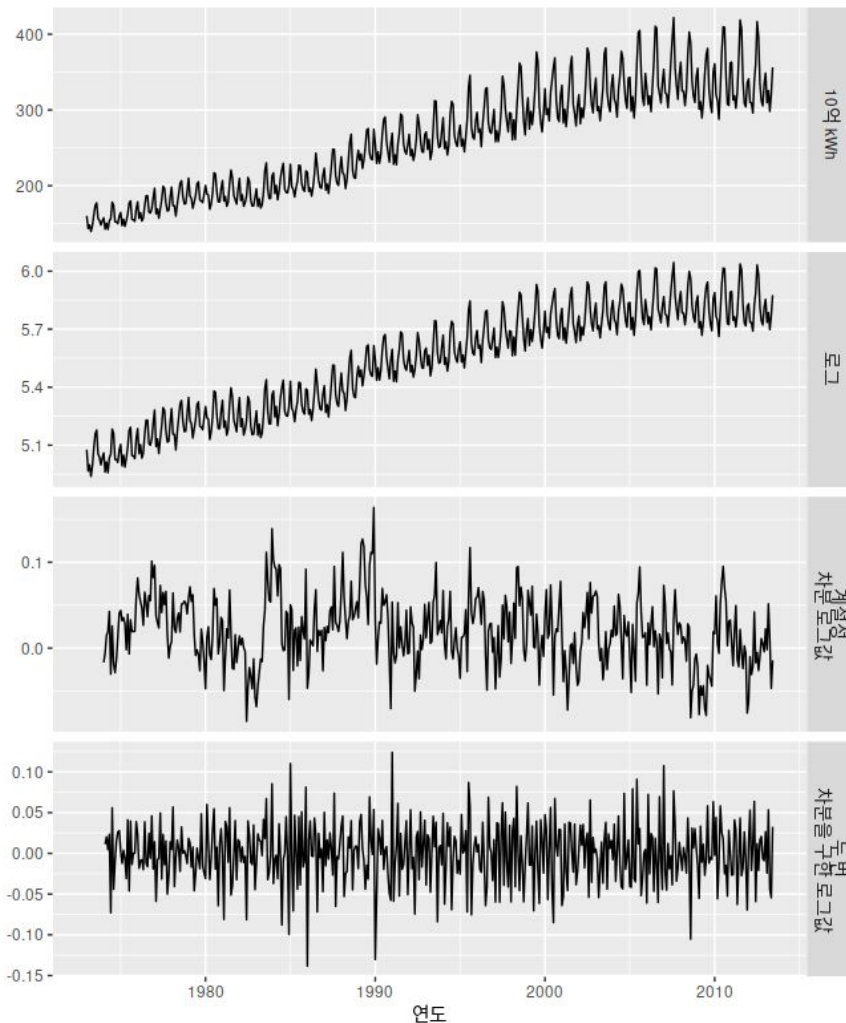
Differencing

stationary한 시계열이 분석이 편함.

non-stationary한 시계열을 stationary한
변환하자!

log를 취하거나
증가량 (미분)을 구하거나
시즌별 변화량을 구하거나

미국 월별 순 전기 생산량



Stationary 시계열을 어떻게 예측할까?

AR (Auto-Regression)

MA (Moving Average)

AR모델 (Auto Regressive:자기 회귀)

Regression - input 변수와 output변수의 관계를 표현
linear reg = 두 변수의 관계를 선형으로

The diagram illustrates the linear regression equation $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ with the following labels and components:

- Dependent Variable**: Points to Y_i .
- Population Y intercept**: Points to β_0 .
- Population Slope Coefficient**: Points to β_1 .
- Independent Variable**: Points to X_i .
- Random Error term**: Points to ϵ_i .
- Linear component**: A bracket under $\beta_0 + \beta_1 X_i$.
- Random Error component**: A bracket under ϵ_i .

AR모델 (Auto Regression: 자기 회귀)

Regression - input 변수와 output 변수의 관계를 표현
linear reg = 두 변수의 관계를 선형으로

Auto Regression - 과거 시간의 변수 값을 input, 예측 시간의 변수값을 output으로 하여 관계 표현

$$\text{AR[1]} \quad Y_t = \phi Y_{t-1} + e_t$$

$$\text{AR[2]} \quad Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$

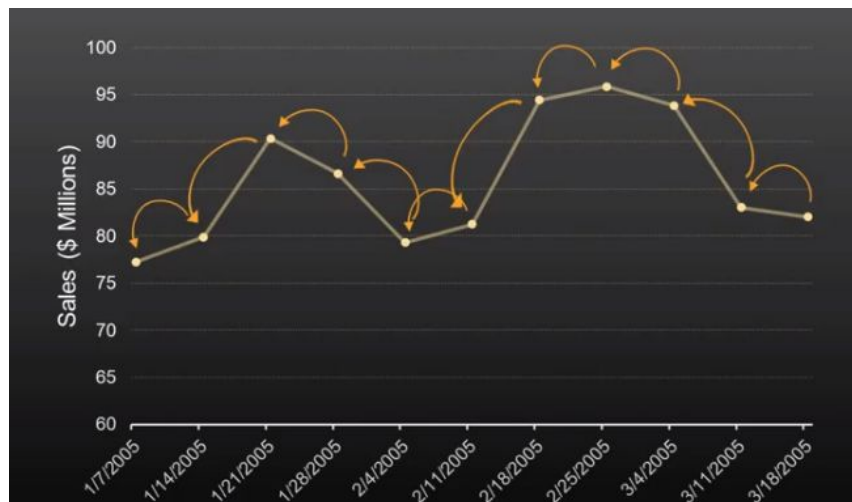
AR모델 (Auto Regression: 자기 회귀)

Auto Regression - 과거 시간의 변수 값을 input, 예측 시간의 변수값을 output으로 하여 관계 표현

가중치 t-1 시간의 변수값

$$\text{AR[1]} \quad Y_t = \phi Y_{t-1} + e_t \quad \text{오차} \sim N(0, \sigma_2)$$

$$\text{AR[2]} \quad Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$



MA 모델 (Moving Average)

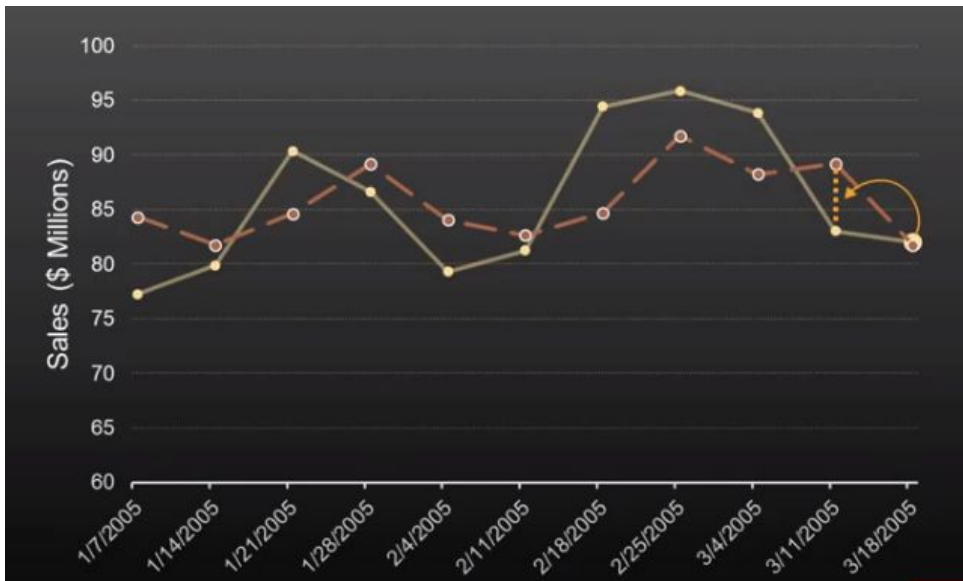
과거 시간의 오차값들의 합으로 예측

t-1 시간일때의
오차값

$$\text{MA(1)} \quad Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

$$\text{MA(2)} \quad Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}$$

t-2 시간일때의
오차값



ARIMA 모델

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

AR (p)

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

+

I

integrated(d)

stationary한 시계열이 되기까지 미분한 횟수

+

MA (q)

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

ARIMA 모델

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

(p,d,q) 순서쌍으로 표현 가능

(0,0,0) 오차만 남음-white noise

(1,0,0) (2,1,0) q=0이므로 전부 AR모델을 표현

(0,1,2) (0,2,3) p=0이므로 전부 MA 모델을 표현

(1,2,3) p,d,q가 모두 존재하는 ARIMA 모델

SARIMA 모델

Seasonal ARIMA

ARIMA 모델에서 계절성 주기가 추가된 모델

계절성 차분

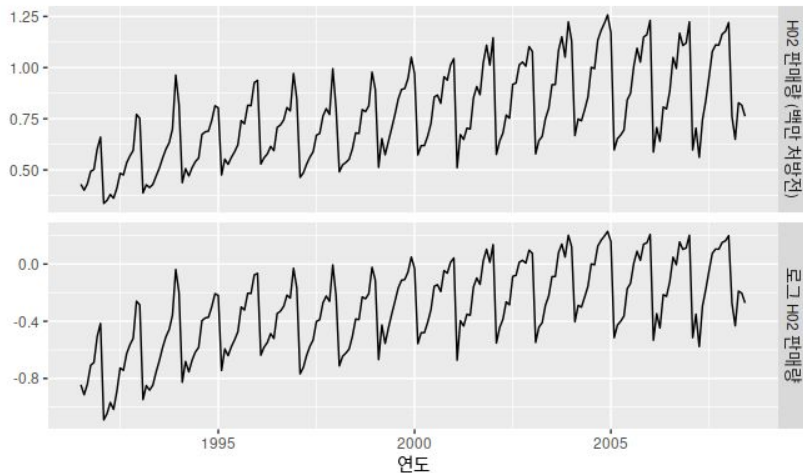
(p,d,q) $(p,d,q)[m]$ 꼴로 표현

비계절성 모델 (p,d,q) +

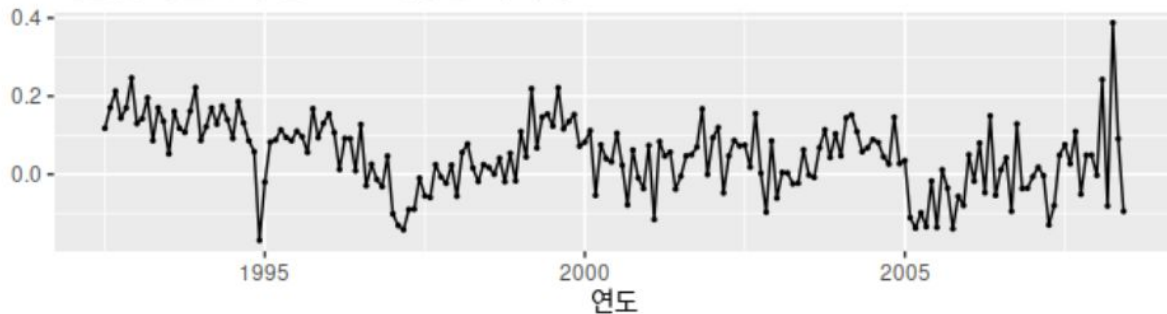
계절성 모델 (p,d,q)

m =(1년에 몇 번 주기인가?)

e.g. $m=4$: 분기별 $m=12$: 월별



계절성 차분을 구한 H02 처방전 데이터



auto-ARIMA

ARIMA는 (p,d,q) 상수에 따라 달라지는 모델

적절한 상수를 찾는 것도 해야할 일!

가능한 (p,d,q)를 모두 시도하여 가장 적절한 (p,d,q)를
찾아주는 것이 바로 auto-ARIMA 함수

```
Performing stepwise search to minimize aic
ARIMA(1,2,1)(0,0,0)[0] : AIC=4296.526, Time=0.06 sec
ARIMA(0,2,0)(0,0,0)[0] : AIC=4331.370, Time=0.01 sec
ARIMA(1,2,0)(0,0,0)[0] : AIC=4303.235, Time=0.04 sec
ARIMA(0,2,1)(0,0,0)[0] : AIC=4294.639, Time=0.04 sec
ARIMA(0,2,2)(0,0,0)[0] : AIC=4296.419, Time=0.07 sec
ARIMA(1,2,2)(0,0,0)[0] : AIC=4294.376, Time=0.12 sec
ARIMA(2,2,2)(0,0,0)[0] : AIC=4289.199, Time=0.14 sec
ARIMA(2,2,1)(0,0,0)[0] : AIC=4288.071, Time=0.09 sec
ARIMA(2,2,0)(0,0,0)[0] : AIC=4290.197, Time=0.05 sec
ARIMA(3,2,1)(0,0,0)[0] : AIC=4289.718, Time=0.15 sec
ARIMA(3,2,0)(0,0,0)[0] : AIC=4288.639, Time=0.08 sec
ARIMA(3,2,2)(0,0,0)[0] : AIC=4290.937, Time=0.30 sec
ARIMA(2,2,1)(0,0,0)[0] intercept : AIC=4289.902, Time=0.22 sec
```

Best model: ARIMA(2,2,1)(0,0,0)[0]

Total fit time: 1.378 seconds

auto-ARIMA

auto-ARIMA가 적절한 상수항을 찾는 기준?

상수값이 큰 모델에 penalty

$$AIC = -2 \log(L) + 2(p + q + k + 1)$$

모델의 가능도

가능한 ARIMA 모델들 중 AIC 값이 가장 작은 모델을 선택함.

```
Performing stepwise search to minimize aic
ARIMA(1,2,1)(0,0,0)[0]      : AIC=4296.526, Time=0.06 sec
ARIMA(0,2,0)(0,0,0)[0]      : AIC=4331.370, Time=0.01 sec
ARIMA(1,2,0)(0,0,0)[0]      : AIC=4303.235, Time=0.04 sec
ARIMA(0,2,1)(0,0,0)[0]      : AIC=4294.639, Time=0.04 sec
ARIMA(0,2,2)(0,0,0)[0]      : AIC=4296.419, Time=0.07 sec
ARIMA(1,2,2)(0,0,0)[0]      : AIC=4294.376, Time=0.12 sec
ARIMA(2,2,2)(0,0,0)[0]      : AIC=4289.199, Time=0.14 sec
ARIMA(2,2,1)(0,0,0)[0]      : AIC=4288.071, Time=0.09 sec
ARIMA(2,2,0)(0,0,0)[0]      : AIC=4290.197, Time=0.05 sec
ARIMA(3,2,1)(0,0,0)[0]      : AIC=4289.718, Time=0.15 sec
ARIMA(3,2,0)(0,0,0)[0]      : AIC=4288.639, Time=0.08 sec
ARIMA(3,2,2)(0,0,0)[0]      : AIC=4290.937, Time=0.30 sec
ARIMA(2,2,1)(0,0,0)[0] intercept : AIC=4289.902, Time=0.22 sec

Best model:  ARIMA(2,2,1)(0,0,0)[0]
Total fit time: 1.378 seconds
```