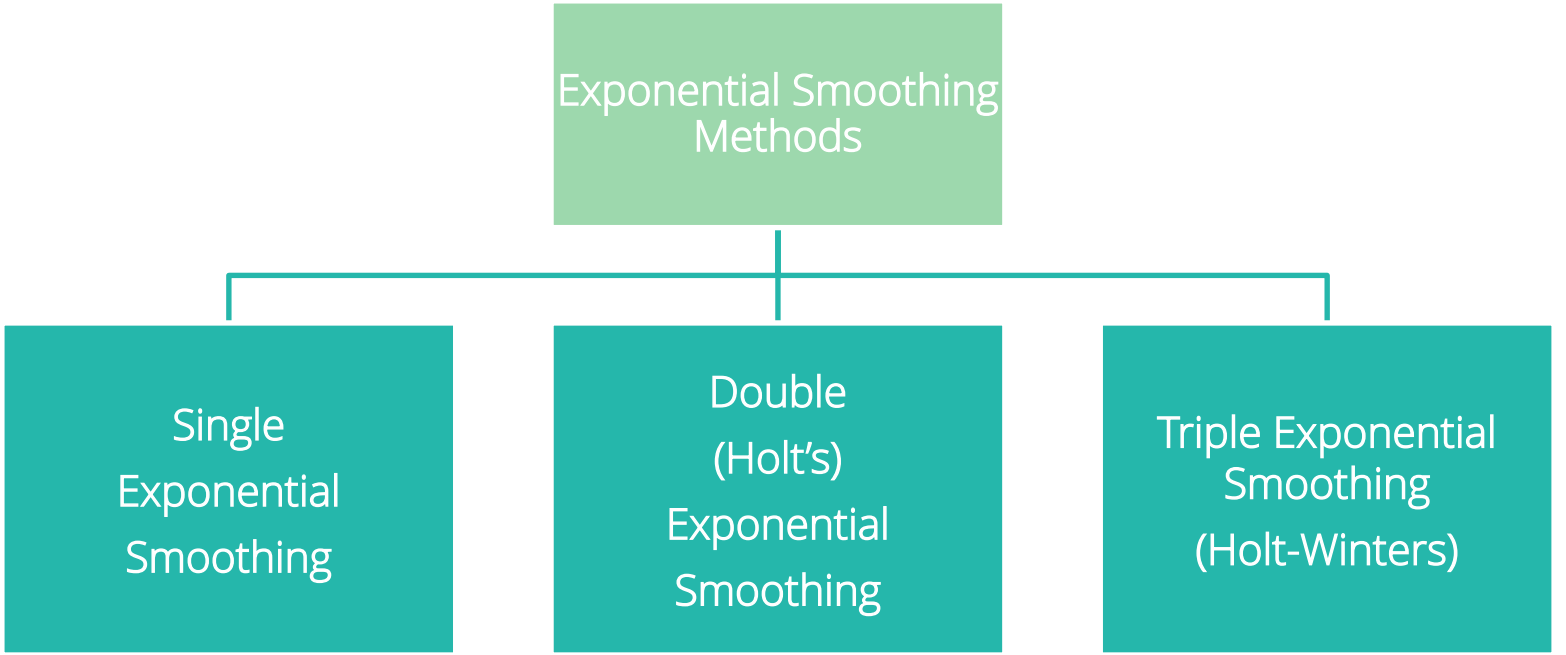
Time Series Analysis –Exponential Smoothing Methods for Forecasting

Contents

Forecasting Using Smoothing Methods

- Random, unexplained variation in a time series can have an undesirable impact on forecasts
- Smoothing can cancel or reduce such impacts
- Smoothing can either be Simple (using Moving Averages) or Exponential



DATA SCIENCE INSTITUTE

Single Exponential Smoothing Model

Mathematical Model :

$$F_{t+1} = \alpha Y_t + (1 - \alpha) F_t$$

Where,

F_{t+1} : Forecast value for period t + 1

F_t : Forecast value for period t

Y_t : Actual value for period t

α : Alpha (Smoothing constant)

Single Exponential Smoothing Model

Assume α=0.8

t	yt	Ft	
1	23	-	
2	24	23	
3	26	23.80	=0.8*24+0.2*23
4	23.5	25.56	=0.8*26+0.2*23.8
5	27	23.91	
6	26.1	26.38	
7	28	26.16	
8	27	27.63	
9	29	27.13	
10	29.3	28.63	
11	28.2	29.17	
12	27	28.39	
		27.28	

1st future value will always be the previous value & then for rest future values exponential smoothing mathematical formula is applied.

Single Exponential Smoothing Model - Smoothing Constant α

Values of α

close to one ✉ have less of a smoothing effect and give greater weight to recent changes in the data

closer to zero ✉ have a greater smoothing effect and are less responsive to recent changes

- There is no formally correct procedure for choosing α. Sometimes the statistician's judgment is used to choose an appropriate factor.

- Alternatively, α can be decided based on statistical measure such as Root Mean Squared Error.

Get an Edge!

Why the Name "Exponential"?

- This method gives weights to past observation in exponentially decreasing manner.

$$F_{t+1} = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \alpha(1-\alpha)^3 y_{t-3} \text{-----}$$

$$= \alpha y_t + (1-\alpha)[\alpha y_{t-1} + \alpha(1-\alpha) y_{t-2} + \alpha(1-\alpha)^2 y_{t-3} \text{----}]$$

$$= \alpha y_t + (1-\alpha) F_t$$

- Larger alpha gives more weight to recent values.

Case Study

Background

- Sales Data for 3 Years (2013, 2014, 2015)

Objective

- To apply Decomposition & Exponential Smoothing to Time Series data using different methods.

Available Information

- Sample size is 36
- Variables: Year, Month, Sales

Data Snapshot

Sales Data for 3 Years

Variables

Year	Month	Sales
2013	Jan	123
2013	Feb	142
2013	Mar	164
2013	Apr	173
2013	May	183
2013	Jun	192
2013	Jul	199
2013	Aug	203
2013	Sep	207
2013	Oct	209
2013	Nov	214
2013	Dec	255

nly Observations

Columns	Description	Type	Measurement	Possible values
Year	Year	factor	2013, 2014, 2015	3
Month	Month	factor	Jan - Dec	12
Sales	Sales in USD Million	numeric	USD Million	Positive values

DATA SCIENCE INSTITUTE

Simple Exponential Smoothing in Python

```
# Import data

import pandas as pd
salesdata = pd.read_csv("Sales Data for 3 Years.csv")
rng = pd.date_range('2013','2016',freq='M')
s = salesdata.Sales.values
salesseries = pd.Series(s, rng)
```

- **freq =** tells Python the frequency of time period in the data, 'M' for monthly data.
- **pd.Series()** converts a column from a data frame to a simple time series object.

Simple Exponential Smoothing in Python

```
#Single Exponential Smoothing
```

```
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
model = SimpleExpSmoothing(salesseries)
fit1 = model.fit()

fit1.predict()

fit1.summary()
```

SimpleExpSmoothing() from holtwinters in statsmodels undertakes exponential smoothing.

Simple Exponential Smoothing in Python

```
# Output

2016-01-31    313.35045
Freq: M, dtype: float64
```

Predicted value

SimpleExpSmoothing Model Results

Dep. Variable:	None	**No. Observations:**	36
Model:	SimpleExpSmoothing	**SSE**	10897.844
Optimized:	True	**AIC**	209.661
Trend:	None	**BIC**	212.828
Seasonal:	None	**AICC**	210.951
Seasonal Periods:	None	**Date:**	Tue, 17 Oct 2023
Box-Cox:	False	**Time:**	14:05:33
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.7351707	alpha	True
initial_level	129.80567	l.0	True

alpha

Interpretation :
It returns predicted future value & value of alpha.

DATA SCIENCE INSTITUTE

Double (Holt) and Triple(Holt-Winters) Exponential Smoothing Methods

Double exponential smoothing has two equations

First equation is similar to single exponential smoothing method

Second equation updates trend using constant beta.

Double exponential smoothing method is used when there is a trend in the time series.

Triple exponential smoothing has three equations

First 2 equations are similar to double exponential smoothing method

Third equation updates seasonal component using constant gamma.

Triple exponential smoothing method is used when there is trend + seasonality in the time series.

Double Exponential Smoothing Model

Mathematical Model :

Where,

F_{t+1}	:	Forecast value for period t +1
F_t	:	Forecast value for period t
T_t	:	Trend component for period t
T_{t+1}	:	Trend component for period t +1
Y_t	:	Actual value for period t
α	:	Alpha (Smoothing constant)
β	:	Beta (Second smoothing constant)

Double Exponential Smoothing in Python

```
#Double Exponential Smoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model = ExponentialSmoothing(salesseries, trend='add',seasonal = None)
fit2 = model.fit()
print(fit2.predict())
fit2.summary()
```

```
# Output
2016-01-31    295.970308
Freq: M, dtype: float64
```

Predicted value

ExponentialSmoothing Model Results

Dep. Variable:	None	**No. Observations:**	
Model:	ExponentialSmoothing	**SSE**	8649.636
Optimized:	True	**AIC**	205.343
Trend:	Additive	**BIC**	211.677
Seasonal:	None	**AICC**	208.240
Seasonal Periods:	None	**Date:**	Tue, 17 Oct 2023
Box-Cox:	False	**Time:**	14:15:55
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.3039444	alpha	True
smoothing_trend	0.3039444	beta	True
initial_level	127.50907	l.0	True
initial_trend	11.565240	b.0	True

alpha, b

Interpretation :
It returns predicted future value, value of alpha and beta.

Triple Exponential Smoothing Model

Mathematical Model :

$$F_{t+1} = \alpha \frac{Y_t}{S_{t+1-k}} + (1-\alpha) F_t - T_t$$

where,

S_{t+1-k}	:	Seasonal smoothing value for period t +1
F_t	:	Forecast value for period t
F_{t+1}	:	Forecast value for period t +1
F_t	:	Forecast value for period t
T_t	:	Trend component for period t
T_{t+1}	:	Trend component for period t +1
Y_t	:	Actual value for period t

α : Alpha (Smoothing constant) β : Beta (Second smoothing constant)
: Gamma (Third smoothing constant)

Triple Exponential Smoothing in Python

```
#Triple Exponential Smoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing
model = ExponentialSmoothing(salesseries,seasonal_periods=12,
trend='add', seasonal='add')
fit3 = model.fit()
print(fit3.predict())
fit3.summary()
```

```
# Output
2016-01-31    293.447983
Freq: M, dtype: float64
```

Predicted value

Interpretation :
It returns predicted future value & value of alpha, beta and gamma.

ExponentialSmoothing Model Results

Dep. Variable:	None	**No. Observations:**	
Model:	ExponentialSmoothing	**SSE**	539.668
Optimized:	True	**AIC**	129.468
Trend:	Additive	**BIC**	154.804
Seasonal:	Additive	**AICC**	169.703
Seasonal Periods:	12	**Date:**	Tue, 17 Oct 2023
Box-Cox:	False	**Time:**	14:06:06
Box-Cox Coeff.:	None		

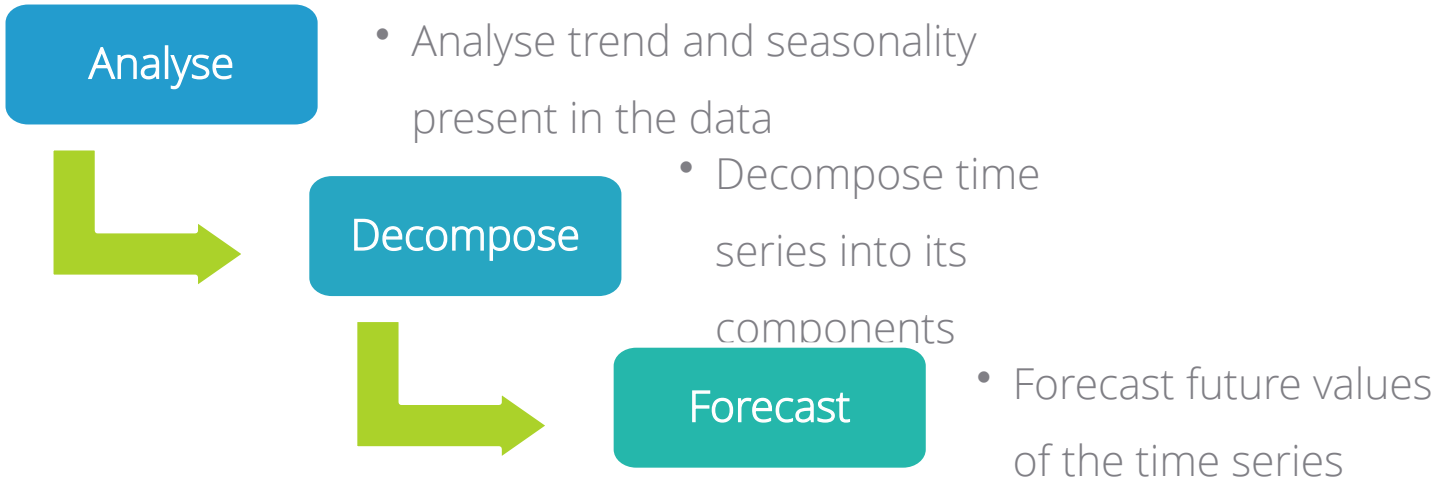
	coeff	code	optimized
smoothing_level	1.0000000	alpha	True
smoothing_trend	0.3703030	beta	True
smoothing_seasonal	1.9375e-08	gamma	True
initial_level	121.18636	l.0	True
initial_trend	13.392973	b.0	True

alpha, beta, gamma

DATA SCIENCE INSTITUTE

Get an Edge!

Always approach time series analysis in a systematic manner



- For vector time series, investigate connections between two or more time series with the aim of using values of some of the processes to predict those of the others. (Eg. Pairs trading in stock market)

Quick Recap

In this session, we learnt about **exponential smoothing**:

Smoothing	• Smoothing gives weights to past observations, in order to give more significance to seasonality and trend components of a time series
Smoothing in Python	• From **`statsmodels.tsa.holtwinters :`** • Use **`SimpleExpSmoothing()`** to carry out simple exponential smoothing • Use **`ExponentialSmoothing()`** to carry out double and triple exponential smoothing