Holt Winters

July 31, 2025 1:09 PM

Triple Exponential Smoothing

Accounts for -

- Level - Estimate of Baseline
- Trend - Rate of change (How much series goes up & down)
- Seasonality - Seasonal component (Repeating / Cyclical patterns)
- Period of seasonality - #Periods in season

We predict y_{t+h} (h forecast steps)

Step 0 - Initialization

- $y_1, y_2, y_3 \ldots y_T$ - Time Series (given)
- m - Season length (e.g. yearly seasonality & monthly data, $m=12$)
- α, β, γ - Smoothing parameter (between 0 & 1)

α - Level Smoothing
β - Trend "
γ - Seasonal "
} → Triple Smoothing

α, β, γ
Control how quickly model updates respective (Level, Trend, Season)
Lower α, β, γ - Slower changes in trend (assuming series is more stable)
Longer model memory.

- L_0 - Avg. of first season

e.g. $m=4$

$y_i = [\, y_1, y_2, y_3, y_4, y_5, y_6 \ldots \,] \quad i \in [1, T]$

$y = [\, \underbrace{30, 21, 29, 40}_{m=4}, 32, 43, \ldots \,]$

$$L_0 = \frac{\sum_{i=1}^{m} y_i}{m} = \frac{30+21+29+40}{4} = \frac{120}{4} = 30$$

- T_0 - Avg. change b/w 2 seasons

e.g. $m=4$

$y = [\, \underbrace{30, 21, 29, 40}_{m=4\ S_1}, \underbrace{32, 43, 50, 41}_{m=4\ S_2} \ldots \,]$

$$T_0 = \frac{\sum_{S_2} y_i - \sum_{S_1} y_i}{m^2} = \frac{166 - 120}{4^2} = \frac{46}{16} = 2.88$$

- $S_1, S_2 \ldots S_m$ - Deviation of each point from L_0

$30 - 30 = 0$

- $S_1, S_2 \ldots S_m$ – Deviation of each point from L_0

$$S_i = y_i - L_0 \Rightarrow \begin{cases} S_1 = y_1 - L_0 = 30 - 30 = 0 \\ S_2 = y_2 - L_0 = 21 - 30 = -9 \\ S_3 = y_3 - L_0 = 29 - 30 = -1 \\ S_4 = y_4 - L_0 = 40 - 30 = 10 \end{cases} \quad m=4$$

Step 1 – Level – Adjusting the Baseline

$$L_t = \alpha (y_t - S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1})$$

- L_t: New Level at time 't'
- α: Level smoothing
- y_t: Current Value
- S_{t-m}: Remove Last season
- $(1-\alpha)$: weight of previous values (to create a trend)
- L_{t-1}: Old level (at t-1)
- T_{t-1}: Trend for last time (at t-1)

Step 2 – Trend – Growth Rate of Series

$$T_t = \beta (L_t - L_{t-1}) + (1-\beta) T_{t-1}$$

- T_t: New Trend at time 't'
- β: Trend Smoothing
- L_t: Current Level
- L_{t-1}: Last Level
- Baseline change
- T_{t-1}: old Trend (at 't-1')

Step 3 – Seasonality – Learn Repeating Pattern

$$S_t = \gamma (y_t - L_t) + (1-\gamma) S_{t-m}$$

- S_t: Seasonality at time 't'
- γ: seasonal smoothing
- y_t: Current Value at 't'
- L_t: Current Level
- S_{t-m}: Seasonal factor same period last season

Step 4 – Forecast

$$\hat{y}_{t+h} = L_t + h T_t + S_{t+h-m(k+1)}$$

- L_t: Average value at time t
- h: Forecast horizon (steps)
- T_t: Rate of change at time t
- Projected Baseline with trend
- $S_{t+h-m(k+1)}$: Seasonal Adjustment – which seasonal factor to use

$k = \left\lfloor \frac{h-1}{m} \right\rfloor$ → How many complete cycles are contained in 'h' when one cycle is of length 'm'

★ e.g. if m=4 (Quarterly)

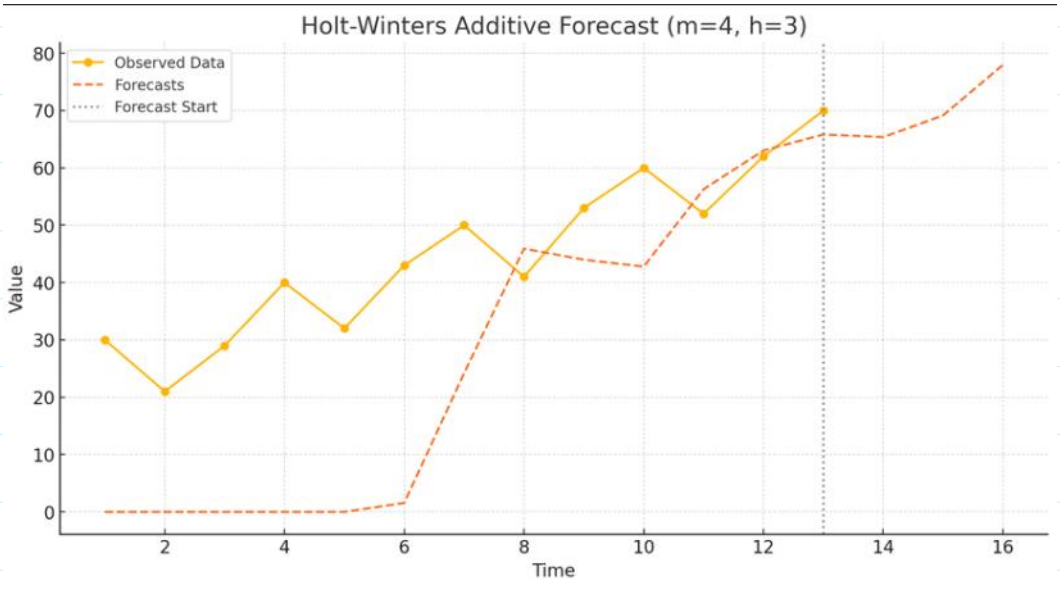
Forecast horizon (steps)

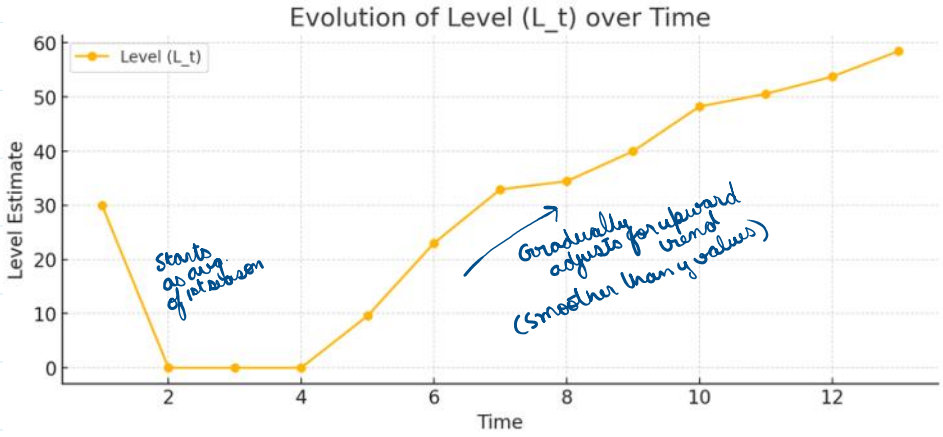
* e.g. if m=4 (Quarterly)
we'll get S_1, S_2, S_3, S_4
If we have an h=3 (we can predict for 3 mo) at time = 8

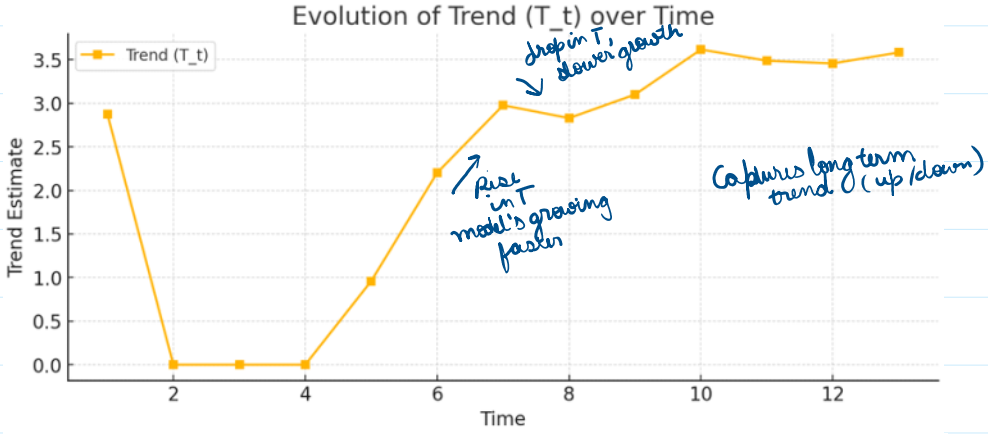
$t+h-m(k+1) = 8+3-4\left(\left\lfloor \frac{3-1}{4} \right\rfloor + 1\right)$
$= 8+3-4(\lfloor 1/2 \rfloor + 1)$ ↳ 1/2 a cycles contained in h=3 so 0 full cycle
$= 11-4(0+1)$
$= 7$

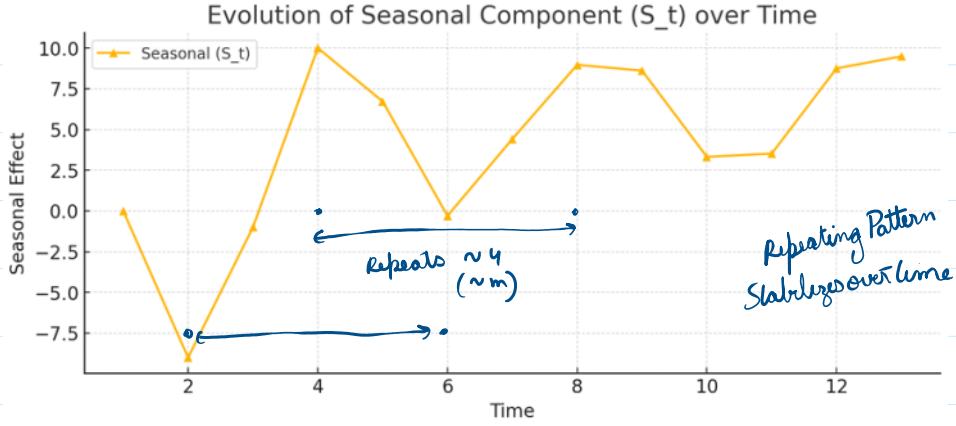
So, we should use $S_7 \Rightarrow S_{4+3} = S_3$ for t=8 & h=3

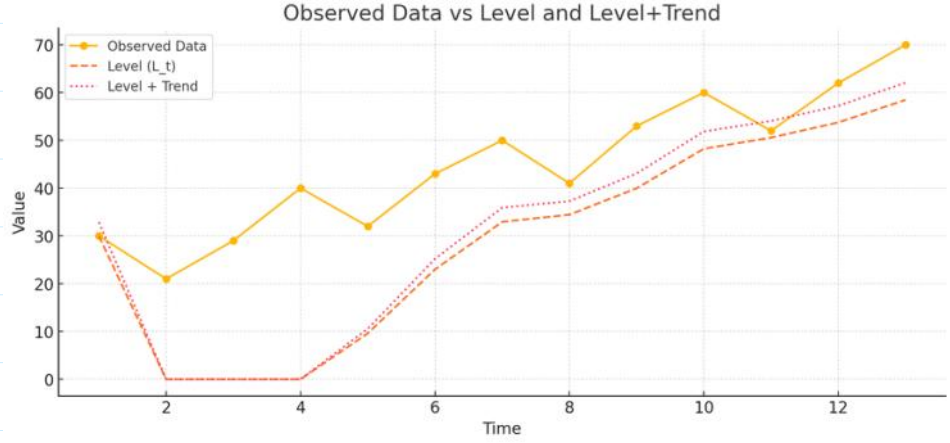
```
y = np.array([30, 21, 29, 40, 32, 43, 50, 41, 53, 60, 52, 62, 70])
t = len(y)
m = 4  # seasonality period
h = 3  # forecast horizon

alpha, beta, gamma = 0.3, 0.1, 0.3
L = np.zeros(t)
T = np.zeros(t)
S = np.zeros(t)

L[0] = np.mean(y[:m])
T[0] = (np.mean(y[m:2*m]) - np.mean(y[:m])) / m
for i in range(m):
    S[i] = y[i] - L[0]
F = np.zeros(t + h)

for i in range(m, t):
    L[i] = alpha * (y[i] - S[i - m]) + (1 - alpha) * (L[i - 1] + T[i - 1])
    T[i] = beta * (L[i] - L[i - 1]) + (1 - beta) * T[i - 1]
    S[i] = gamma * (y[i] - L[i]) + (1 - gamma) * S[i - m]
    F[i] = L[i - 1] + T[i - 1] + S[i - m]

for i in range(1, h + 1):
    idx = t + i - 1
    F[idx] = L[t - 1] + i * T[t - 1] + S[t - m + (i - 1) % m]

df = pd.DataFrame({
    "Time": np.arange(1, t + h + 1),
    "Observed": np.append(y, [np.nan] * h),
    "Level": np.append(L, [np.nan] * h),
    "Trend": np.append(T, [np.nan] * h),
    "Seasonal": np.append(S, [np.nan] * h),
    "Forecast": F
})

plt.figure(figsize=(12, 6))
plt.plot(df["Time"], df["Observed"], marker="o", label="Observed Data")
plt.plot(df["Time"], df["Forecast"], linestyle="--", marker="x", label="Forecasts")
plt.axvline(x=t, color="gray", linestyle=":", label="Forecast Start")
plt.title("Holt-Winters Additive Forecast (m=4, h=3)")
plt.xlabel("Time")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.show()

import ace_tools as tools; tools.display_dataframe_to_user(name="Holt-Winters Step-by-Step Table", dataframe=df)
```











ADTo answers why we call this model exponential

Exponential

∴ These weights $(1-\alpha)$ decay exponentially for past terms

We can write the above equations as

Exponential

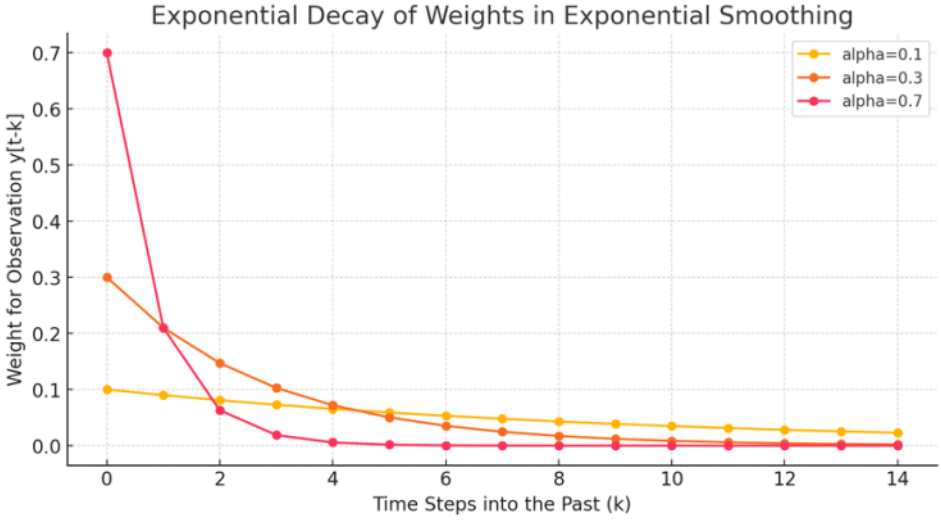
∵ These weights $(1-\alpha)$ decay exponentially for past terms

We can write the above equations as

e.g. $L_t = \alpha y_t + (1-\alpha) L_{t-1}$ $\quad L_{t-1} = \alpha y_{t-1} + (1-\alpha) L_{t-2}$ $\quad L_{t-2} = \alpha y_{t-2} + (1-\alpha) L_{t-3}$

$\Rightarrow L_t = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \alpha(1-\alpha)^3 y_{t-3} + \dots$

Weight $= \alpha(1-\alpha)^k$



This is why we say $\alpha, \beta, \gamma \in (0,1)$

& lower it is, smoother the model & more slowly it learns

Multiplicative

Used when seasonality 'scales' with level

Usually better models where trends sharp & high & seasonal swing changes with trend

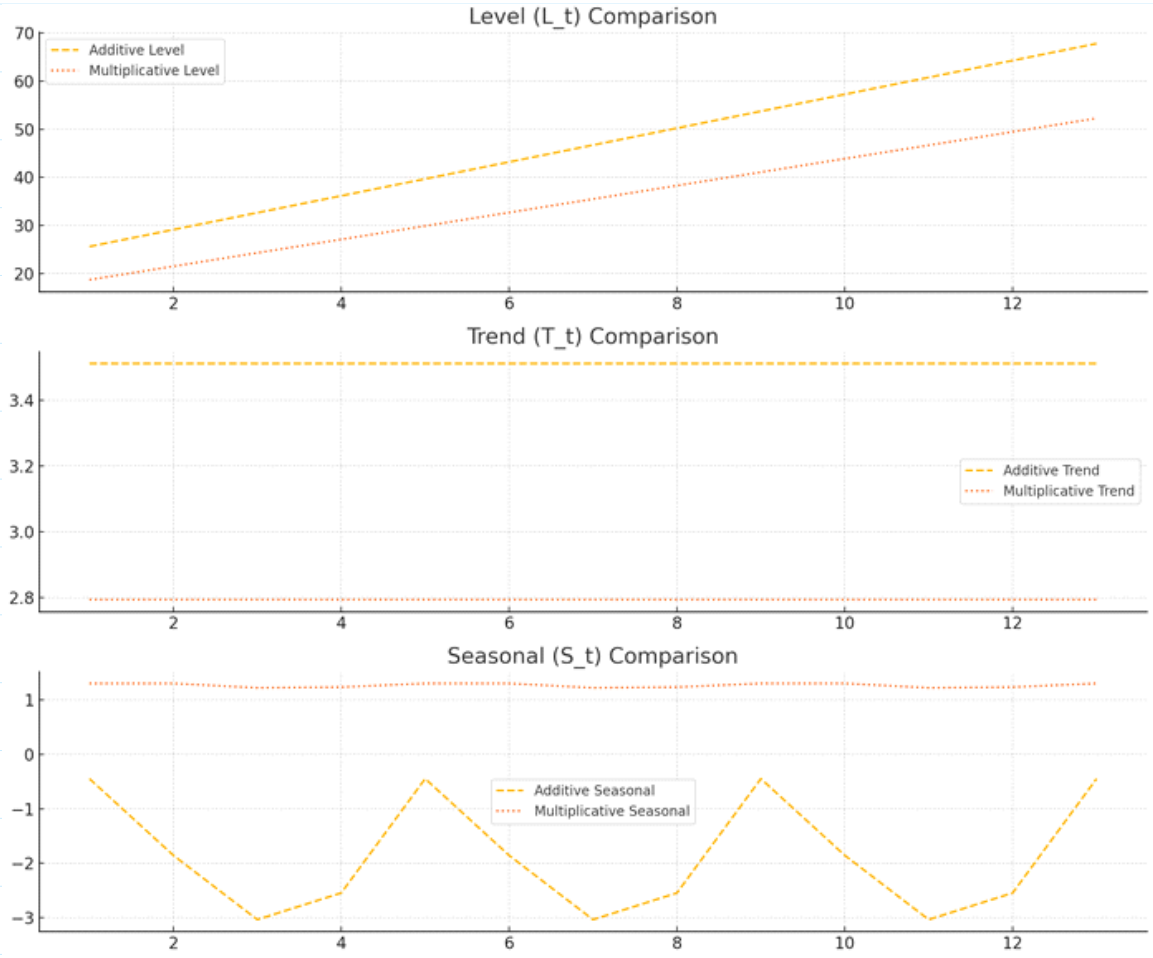
$L_t = \alpha \frac{y_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + T_{t-1})$ → Divide instead of subtract

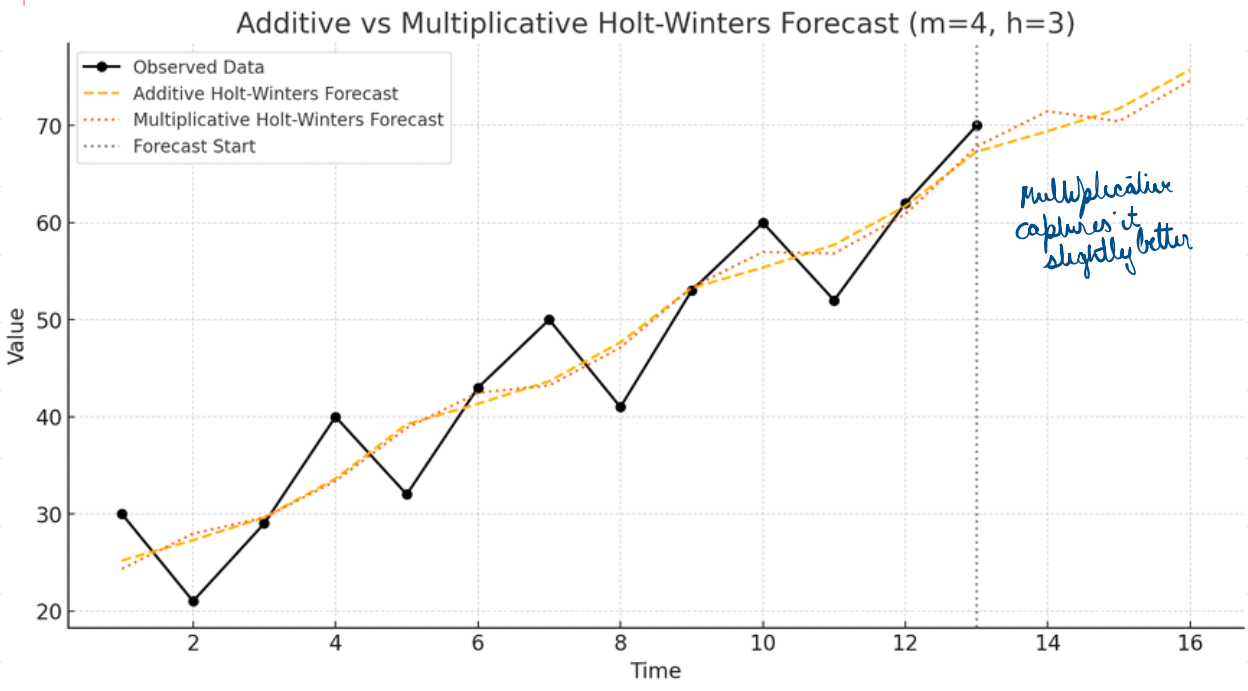
$T_t = \beta(L_t - L_{t-1}) + (1-\beta) T_{t-1}$] Same

$S_t = \gamma \frac{y_t}{L_t} + (1-\gamma) S_{t-m}$ → Divide instead of subtract

$\Rightarrow y_{t+h} = (L_t + hT_t) \times S_{t+h-m(k+1)}$

Multiplied instead of addition


Level (L_t) Comparison
Additive Level
Multiplicative Level
Trend (T_t) Comparison
Additive Trend
Multiplicative Trend
Seasonal (S_t) Comparison
Additive Seasonal
Multiplicative Seasonal


Additive vs Multiplicative Holt-Winters Forecast (m=4, h=3)
Observed Data
Additive Holt-Winters Forecast
Multiplicative Holt-Winters Forecast
Forecast Start
Value
Time
multiplicative captures it slightly better