



# ARISA Learning Material

**Educational Profile and EQF level: DATA SCIENTIST EQF 6**

**PLO: 1, 2, 3, 4, 5**

**Learning Unit (LU): MACHINE LEARNING: SUPERVISED**

**Topic: 1. INTRODUCTION**



[www.aiskills.eu](http://www.aiskills.eu)

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

**Copyright © 2024 by the Artificial Intelligence Skills Alliance**

**All learning materials (including Intellectual Property Rights) generated in the framework of the ARISA project are made freely available to the public under an open license [Creative Commons Attribution–NonCommercial](#) (CC BY-NC 4.0).**

**ARISA Learning Material 2024**

**This material is a draft version and is subject to change after review coordinated by the European Education and Culture Executive Agency (EACEA).**

**Authors: Universidad Internacional de La Rioja**

**Disclaimer: This learning material has been developed under the Erasmus+ project ARISA (Artificial Intelligence Skills Alliance) which aims to skill, upskill, and reskill individuals into high-demand software roles across the EU.**



This project has been funded with support from the European Commission. The material reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- **About ARISA**

- The Artificial Intelligence Skills Alliance (ARISA) is a four-year transnational project funded under the EU's Erasmus+ programme. It delivers a strategic approach to sectoral cooperation on the development of Artificial Intelligence (AI) skills in Europe.
- ARISA fast-tracks the upskilling and reskilling of employees, job seekers, business leaders, and policymakers into AI-related professions to open Europe to new business opportunities.
- ARISA regroups leading ICT representative bodies, education and training providers, qualification regulatory bodies, and a broad selection of stakeholders and social partners across the industry.

[ARISA Partners & Associated Partners](#) | [LinkedIn](#) | [Twitter](#)

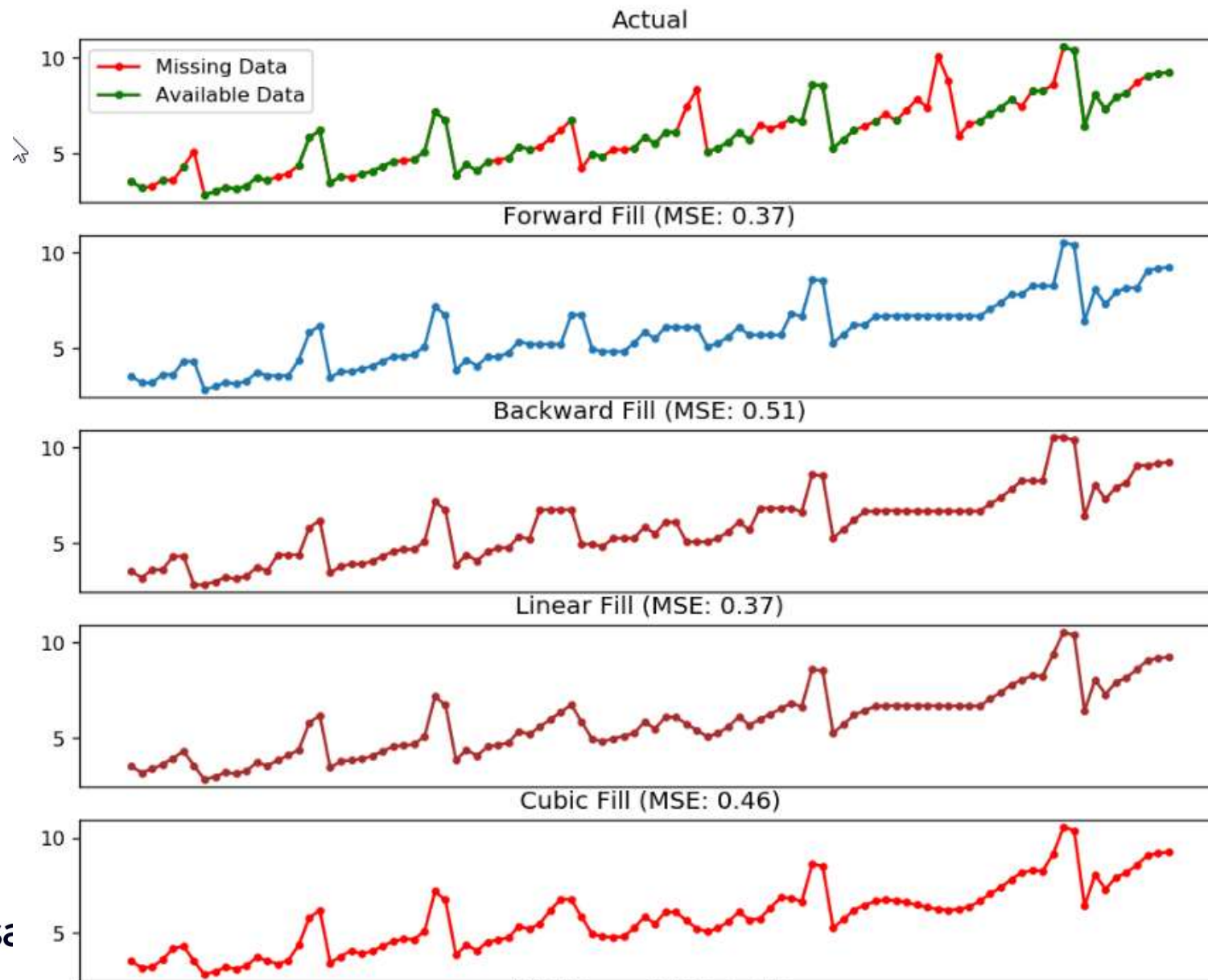
## Conceptos de ST

- Transformación de los datos
- Fecha como índice (datetime)
- Frecuencia
- Valores Faltantes
- Training-Test
- Agregando en otras frecuencias
- Suavizado de Señal

Alias	Description
B	business day frequency
C	custom business day frequency (experimental)
D	calendar day frequency
W	weekly frequency
M	month end frequency
SM	semi-month end frequency (15th and end of month)
BM	business month end frequency
CBM	custom business month end frequency
MS	month start frequency
SMS	semi-month start frequency (1st and 15th)
BMS	business month start frequency
CBMS	custom business month start frequency
Q	quarter end frequency
BQ	business quarter endfrequency
QS	quarter start frequency
BQS	business quarter start frequency
A	year end frequency
BA	business year end frequency
AS	year start frequency
BAS	business year start frequency
BH	business hour frequency
H	hourly frequency
T, min	minutely frequency
S	secondly frequency
L, ms	milliseconds
U, us	microseconds
N	nanoseconds

## Valores faltantes

- A veces, la ST tendrá fechas/horas faltantes. Eso significa que los datos no fueron capturados o no estuvieron disponibles durante esos períodos.
- Cuando se trata de series temporales, típicamente NO se deberían reemplazar los valores faltantes con la media de la serie, especialmente si la serie no es estacionaria.
- En su lugar, como una solución rápida y simple es rellenar hacia adelante el valor previo. Sin embargo, dependiendo de la naturaleza de la serie, querrás probar múltiples enfoques antes de llegar a una conclusión.



## Suavizado

- El filtrado de una ST consiste en suavizar la curva para facilitar su análisis y modelado.
- **SMA: Media móvil simple**(Inglés: promedio móvil simple, **SMA**)
- **EWMA: Exponentially Weighted Moving Average: Promedio móvil ponderado exponencialmente**
- Suavizado exponencial simple (solo se aplica un factor de suavizado)
- Holt-Winters – suavizado doble y triple que considera la tendencia y la estacionalidad para hacer mejor suavizado
- Son métodos de predicción básicos
- Existen otros métodos (EMA, WMA, ...)



## Características fundamentales de ST

- Ruido Blanco
- Autocorrelacion
- Caminata Aleatoria
- Estacionariedad vs. Estacionalidad
- Autocorrelacion
- Descomposicion

# Ruido Blanco

- Tipo especial de serie temporal donde los datos dependen del tiempo pero no siguen ningún patron
- La idea de predicción de serie temporales es que los patrones del pasado se repiten en el futuro
- EL ruido blanco **no tiene patrones => no se puede predecir**
- Una serie se considera ruido blanco si:
  - Media constante (valor alrededor del cual varia)
  - Varianza constante (como varia alrededor de la media)
  - No tener autocorrelaciones en ningún periodo

## Autocorrelacion

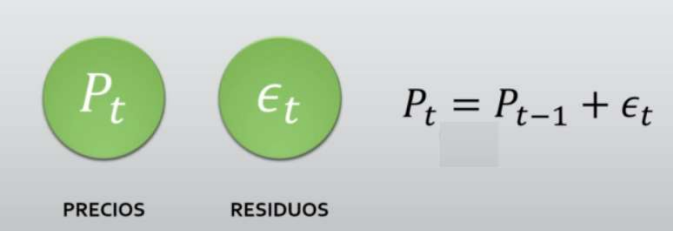
- Mide cuan correlacionadas es una serie con versiones anteriores de si misma

$$\rho = \text{corr}(X_t, X_{t-1})$$

La falta de autocorrelacion (No autocorrelacion) indica que no hay una relación entre valores pasados y presentes. EL ruido blanco es una **secuencia de datos aleatorios**.

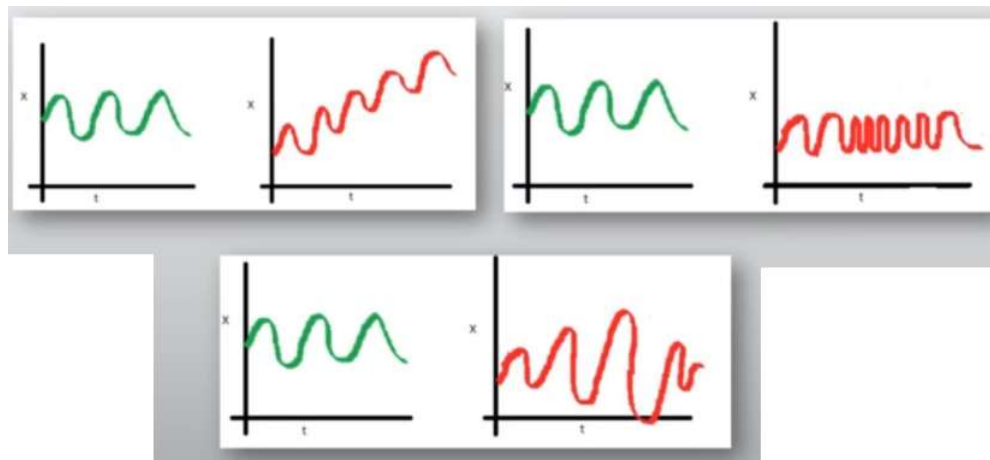
# Caminata Aleatoria / Random Walk

- Las **caminatas aleatorias** son unas ST donde la posición de una partícula en cierto instante depende solo de su posición en algún instante previo y alguna variable **aleatoria**.
- Random Walk inicialmente se refiere a una serie temporal que no capta tendencia ni estacionalidad, dicho de otra forma, únicamente encontramos la serie temporal con el error ( $\epsilon_t$ ).
- Muchas ST son caminatas aleatorias, en particular los de los precios de los valores a lo largo del tiempo.

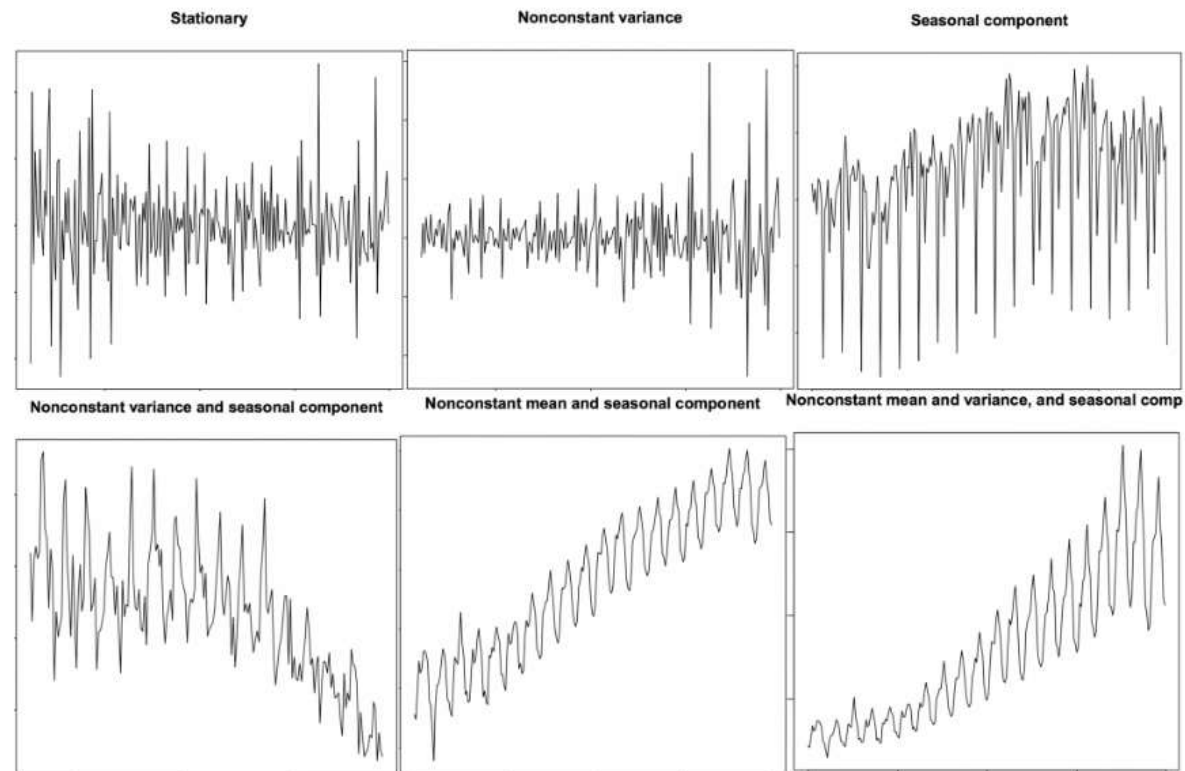

$$P_t = P_{t-1} + \epsilon_t$$

# Estacionariedad vs. Estacionalidad

- Serie estable a lo largo del tiempo: media y varianza son constantes en el tiempo y sin tendencias.
- Serie no estacionaria: Tendencia y/o varianza cambian en el tiempo
  - Cambios en la media determinan una tendencia (crecer o decrecer)
  - Varianza cambia la velocidad de cambio



# Estacionariedad vs. Estacionalidad



Stationary and Non-Stationary Time Series

## Test de Dickey Fuller (DF) de Estacionariedad

- Método para identificar si una serie es estacionaria o no
- La prueba es un contraste de hipótesis:
  - Hipótesis Nula ( $H_0$ ) – serie no es estacionaria
  - Hipótesis Alternativa ( $H_1$ ) – serie estacionaria
  - La prueba es contrastar que la hipótesis nula ( $H_0$ ) se verifica o no
- Contraste de Hipótesis:
  - La muestra representa a una población mayor de la que ha sido obtenida
  - EN el caso de ST esos datos son una muestra de un proceso que los ha generado
  - La inferencia aborda el problema de extender las muestras al futuro (inferencia en base a la muestra)
- La hipótesis nula es que la serie no estacionaria ( $H_0$ )
  - $H_0$  Coeficiente de autocorrelación de un retraso es 1
  - $H_1$  el coeficiente de autocorrelación de un retraso es menor que 1

# Descomposición

- Se supone que cualquier periodo  $t$  el valor observado de la ST es la suma de Tendencia+Estacionalidad+residuo
- Descomposición Multiplicativa
  - Producto de los tres efectos
- Python incluye seasonal decompose y la divide en estos tres elementos.



## Filtro Heidrick-Precott (SI)

- FILTRO DE HEIDRICK-PRECOTT DESCOMPOEN UNA SERIE EN DOS COMPONENTES: TENDECIA Y CICLICO

Separa una serie de tiempo en un componente de tendencia y un componente cíclico.

$$x_t = \mu_t + c_t$$

Depende de un parámetro  $\lambda$  que mide las variaciones en la tasa de crecimiento del componente de tendencia.

Trimestrales: 1600  
Anuales: 6.25  
Mensuales: 129600

## ¿Qué viene después de la autoregresión?

- **Modelos AR (Autoregresivos):** Predicen el valor actual usando valores pasados.
  - **MA (Media Móvil):** Usa errores pasados para ajustar predicciones.
  - **ARMA / ARIMA:** Combinan autoregresión y media móvil, con integración para manejar no estacionariedad.
  - **SARIMA:** Extiende ARIMA para capturar estacionalidad
- Estos modelos son poderosos, pero presentan desafíos importantes, especialmente cuando las series son complejas o tienen patrones estacionales variables

## *¿Por qué buscar alternativas?*

- **Requieren Estacionariedad:** Necesidad de transformar los datos (diferenciación, etc.).
- **Complejidad en la Parametrización:** Difícil identificar  $p$ ,  $d$ ,  $q$  (y  $P$ ,  $D$ ,  $Q$  en SARIMA). AutoARIMA es una solución parcial.
- **Sensibles a Valores Atípicos:** Problemas si hay
- **Limitaciones en Series con Tendencias No Lineales o Estacionalidad Compleja.**
- Pueden ser difíciles de ajustar correctamente, especialmente cuando trabajamos con datos reales que son más 'desordenados'.

## ¿Qué buscamos en un buen modelo de series temporales?

1. **Capacidad para manejar estacionalidad compleja** (diaria, semanal, anual).
2. **Robustez ante valores atípicos** y cambios abruptos.
3. **Facilidad de uso:** menos necesidad de ajustar parámetros manualmente.
4. **Flexibilidad para incorporar información externa** (vacaciones, eventos).

## El futuro de las predicciones de series temporales

- **Necesitamos modelos que:**

- Se ajusten automáticamente a tendencias y estacionalidades.
- Permitan la incorporación de eventos externos (festivos, promociones).
- Sean robustos y fáciles de interpretar por usuarios no expertos en estadística.

- **Aquí entra DL / Librerías como Prophet:**

- Diseñado para ser intuitivo, flexible y capaz de manejar datos complejos con mínima configuración

## Prophet

- Diseñada para pronosticar datos comerciales de Facebook: frecuencia alta (minuto)
- Pensada para evaluar los cambios diarios en los datasets que genera
- Se puede usar en datasets que no son de alta frecuencia
- No es ni mejor ni peor que ARIMA – es una alternativa
- Su principal ventaja es su sencillez (vs ARIMA)
- Se basa en un modelo aditivo descomponible donde las tendencias no lineales se ajustan a la estacionalidad, también tiene en cuenta los efectos de las vacaciones.

# Prophet

- En esencia, el procedimiento Prophet es un modelo de regresión aditiva con cuatro componentes principales:
  - Una tendencia de curva de crecimiento lineal o logística a trozos. Prophet detecta automáticamente los cambios en las tendencias seleccionando puntos de cambio de los datos.
  - Componente estacional anual modelado utilizando series de Fourier.
  - Un componente estacional semanal que utiliza variables ficticias.
  - Una lista de días festivos importantes proporcionada por el usuario.

Documento : <https://peerj.com/preprints/3190.pdf>

# Codigo basico

## Format the Data

```
In [12]: df.columns = ['ds', 'y']
```

```
In [13]: df['ds'] = pd.to_datetime(df['ds'])
```

```
In [14]: df
```

## Create and Fit Model

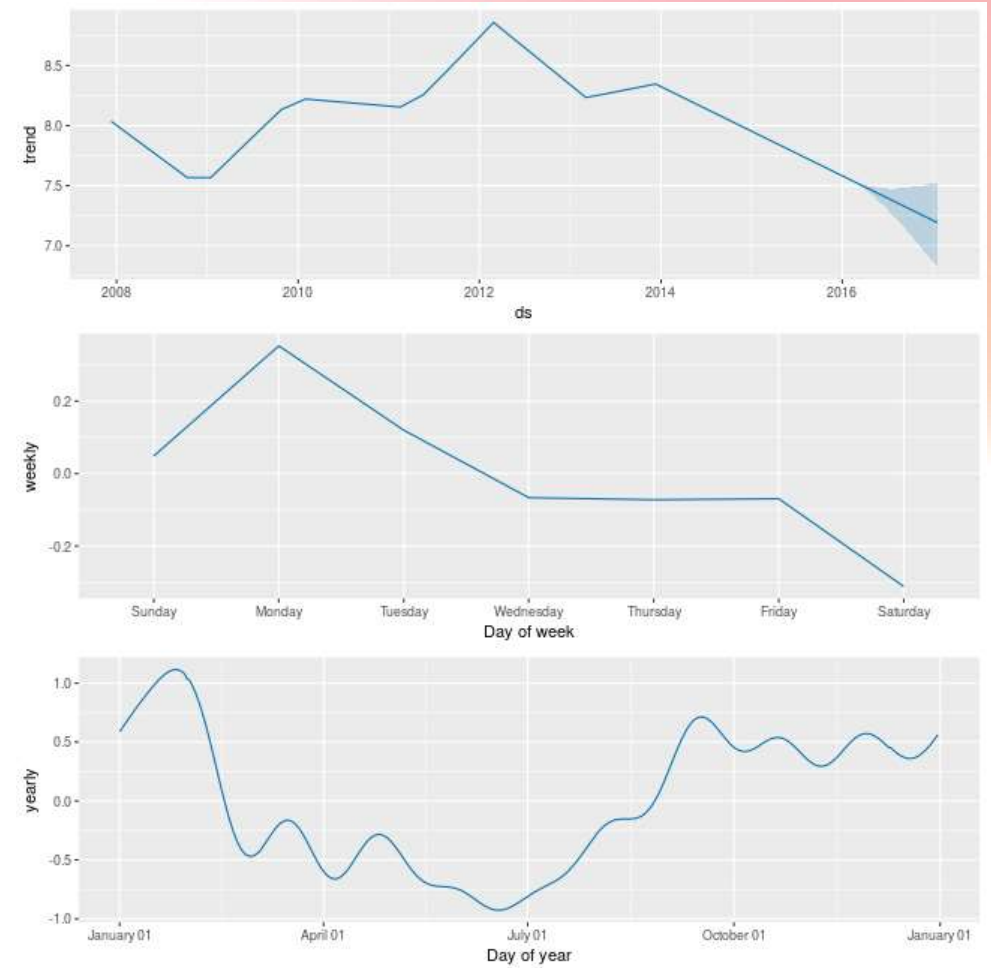
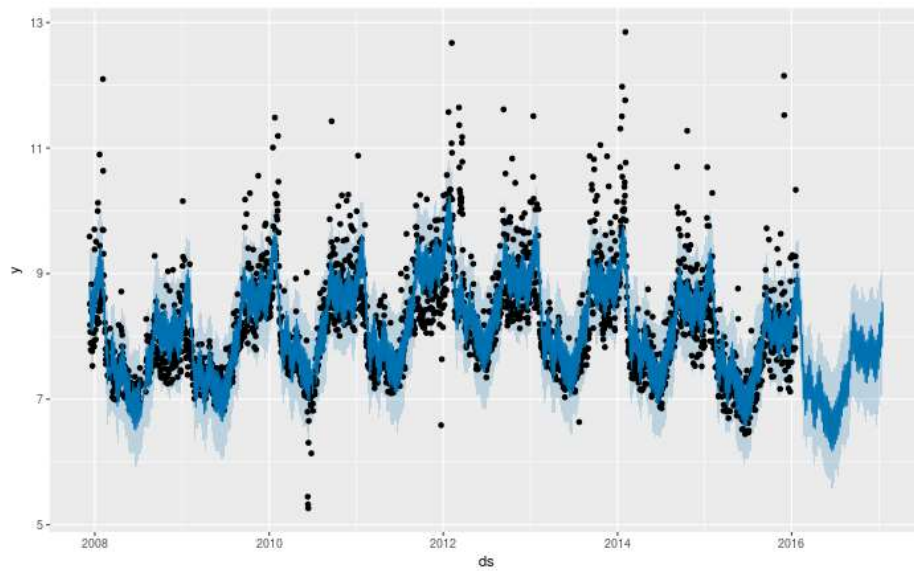
```
In [15]: m = Prophet()  
m.fit(df)
```

```
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.  
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Out[15]: <prophet.forecaster.Prophet at 0x22b16d217c0>
```



# Decomposición



# Parámetros

- [https://www.sktime.org/en/stable/api\\_reference/auto\\_generated/sktime.forecasting.fbprophet.Prophet.html](https://www.sktime.org/en/stable/api_reference/auto_generated/sktime.forecasting.fbprophet.Prophet.html)
- Los valores por defecto funcionan bastante bien – y sino los identifica Prophet
- **freq**: str, default=None A DatetimeIndex frequency. For possible values see [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/timeseries.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html)
- **add\_seasonality**: dict or None, default=None Dict with args for Prophet.add\_seasonality(). Dict can have the following keys/values:
  - name: string name of the seasonality component.
  - period: float number of days in one period.
- **add\_country\_holidays**: dict or None, default=None
  - Dict with args for Prophet.add\_country\_holidays(). Dict can have the following keys/values:
    - country\_name: Name of the country, like 'UnitedStates' or 'US'
- **growth**: str, default="linear" String 'linear' or 'logistic' to specify a linear or logistic trend.
- **changepoints**: list or None, default=None. List of dates at which to include potential changepoints. If not specified, potential changepoints are selected automatically.
- **n\_changepoints**: int, default=25 Number of potential changepoints to include. Not used if input changepoints is supplied. If changepoints is not supplied, then n\_changepoints potential changepoints are selected uniformly from the first changepoint\_range proportion of the history.
- **changepoint\_range**: float, default=0.8 Proportion of history in which trend changepoints will be estimated. Defaults to 0.8 for the first 80%. Not used if changepoints is specified.
- **yearly\_seasonality**: str or bool or int, default="auto" Fit yearly seasonality. Can be 'auto', True, False, or a number of Fourier terms to generate.

## Growth

- This parameter is the easiest to understand and implement as you only have to plot your data to know what it should be. If you plot your data and you see a trend that keeps on growing with ***no real saturation insight*** (or if your domain expert tells you there is no saturation to worry about) you will set this parameter to ***“linear”***.
- If you plot it and you see a curve that is ***showing promise of saturation*** (or if you are working with values that you know must saturate, for example CPU usage) then you will set it to ***“logistic”***.



# Changepoints

- Changepoints are the points in your data where there are sudden and abrupt changes in the trend. There are four hyperparameters for changepoints: *changepoints*, *n\_changepoints*, *changepoint\_range* and *changepoint\_prior\_scale*.
- The ***changepoints*** parameter is used when you supply the changepoint dates instead of having Prophet determine them. Once you have provided your own changepoints, Prophet will not estimate any more changepoints.
- ***changepoint\_prior\_scale***, is there to indicate how flexible the changepoints are allowed to be. In other words, how much can the changepoints fit to the data. If you make it high it will be more flexible, but you can end up overfitting.
- ***changepoint\_range*** usually does not have that much of an effect on the performance. (around 0.8)

## Seasonality

- These parameters are where Prophet shines as you can make big improvements and gain great insights by changing only a few values
- ***seasonality\_mode***. This parameter indicates how your seasonality components should be integrated with the predictions. The default value here is ***additive*** with ***multiplicative*** being the other option
- ***seasonality\_prior\_scale*** parameter. This parameter will again allow your seasonalities to be more flexible. Values between 10 and 25 tend to work well.



#AIskills4all |  @AIskillsEU |  [linkedin.com/AIskillsEU](https://www.linkedin.com/AIskillsEU)



[www.aiskills.eu](https://www.aiskills.eu)

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.