UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA
FACULTAD DE ECONOMÍA Y PLANIFICACIÓN
UNIDAD DE EXTENSIÓN Y PROYECCIÓN SOCIAL
Ingeniería en Gestión Empresarial, Economía y Estadística Informática

En convenio con
BUSINESS SCHOOL

BIG DATA & DATA SCIENCE

PROGRAMA DE ESPECIALIZACIÓN

Machine Learning para Series Temporales

¿Qué es una serie temporal y qué tiene de especial?

Una serie temporal es un conjunto de muestras tomadas a intervalos de tiempo regulares. Es interesante analizar su comportamiento al mediano y largo plazo, intentando detectar patrones y poder hacer pronósticos de cómo será su comportamiento futuro.

Lo que hace <<especial>> a una Time Series a diferencia de un "problema" de Regresión son dos cosas:

1. **Es dependiente del Tiempo**. Esto rompe con el requerimiento que tiene la regresión lineal de que sus observaciones sean independientes.

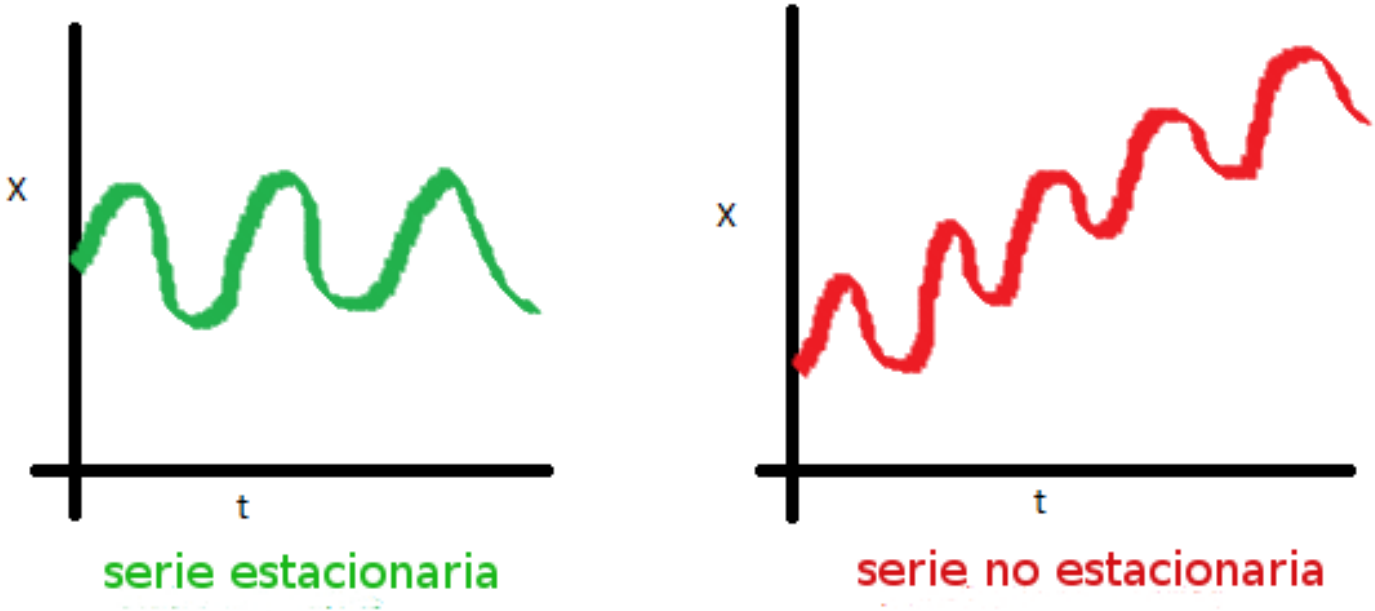
2. **Suelen tener algún tipo de estacionalidad**, o de **tendencias** a crecer o decrecer. Pensemos en cuánto más producto vende una heladería en sólo 4 meses al año que en el resto de estaciones.

Suelen estar **autocorrelacionadas**; la mayoría de los procesos físicos presentan una inercia y no cambian tan rápidamente. Esto, combinado con la frecuencia del muestreo, a menudo hace que las observaciones consecutivas estén correlacionadas. Esta correlación entre observaciones consecutivas se llama autocorrelación. Cuando los datos están **autocorrelacionados**, la mayoría de los métodos estadísticos estándares basados en la suposición de observaciones independientes pueden arrojar resultados engañosos o incluso ser inútiles.

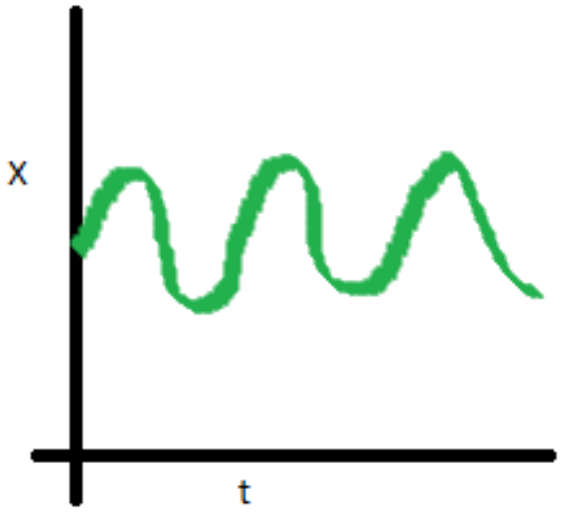
Series de tiempo estacionarias

Un tipo muy importante de series de tiempo son las series de tiempo estacionarias. Una series de tiempo se dice que es estrictamente estacionaria si sus propiedades no son afectadas por los cambios a lo largo del tiempo. Es decir, que se deberían cumplir tres criterios básicos para poder considerar a una series de tiempo como estacionaria: **La media de la serie no debe ser una función de tiempo, La varianza de la serie no debe ser una función del tiempo, La covarianza de la serie no debe ser una función del tiempo**

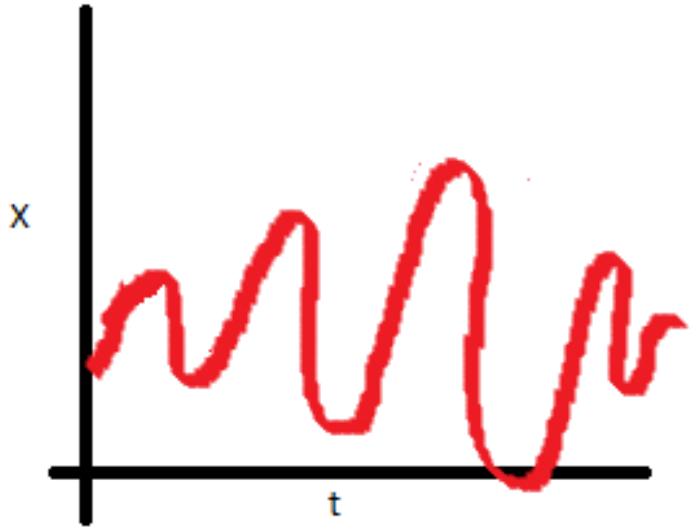
La media de la serie no debe ser una función de tiempo; sino que debe ser constante. La siguiente imagen muestra una serie que cumple con esta condición y otra que no la cumple.



La varianza de la serie no debe ser una función del tiempo. El siguiente gráfico representa una serie cuya varianza no esta afectada por el tiempo (es estacionaria) y otra que no cumple con esa condición.

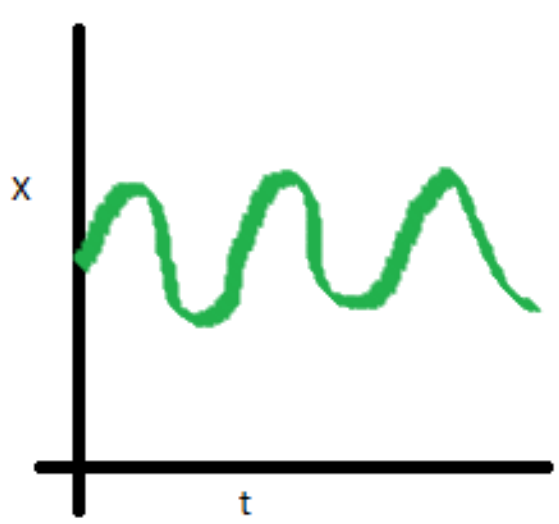


serie estacionaria

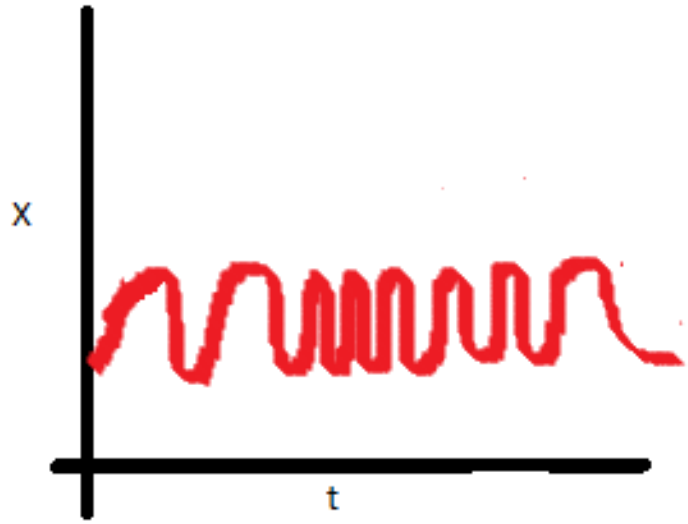


serie no estacionaria

La covarianza de la serie no debe ser una función del tiempo. En el gráfico de la derecha, se puede observar que la propagación de la serie se va encogiendo a medida que aumenta el tiempo. Por lo tanto, la covarianza no es constante en el tiempo para la serie roja.



serie estacionaria



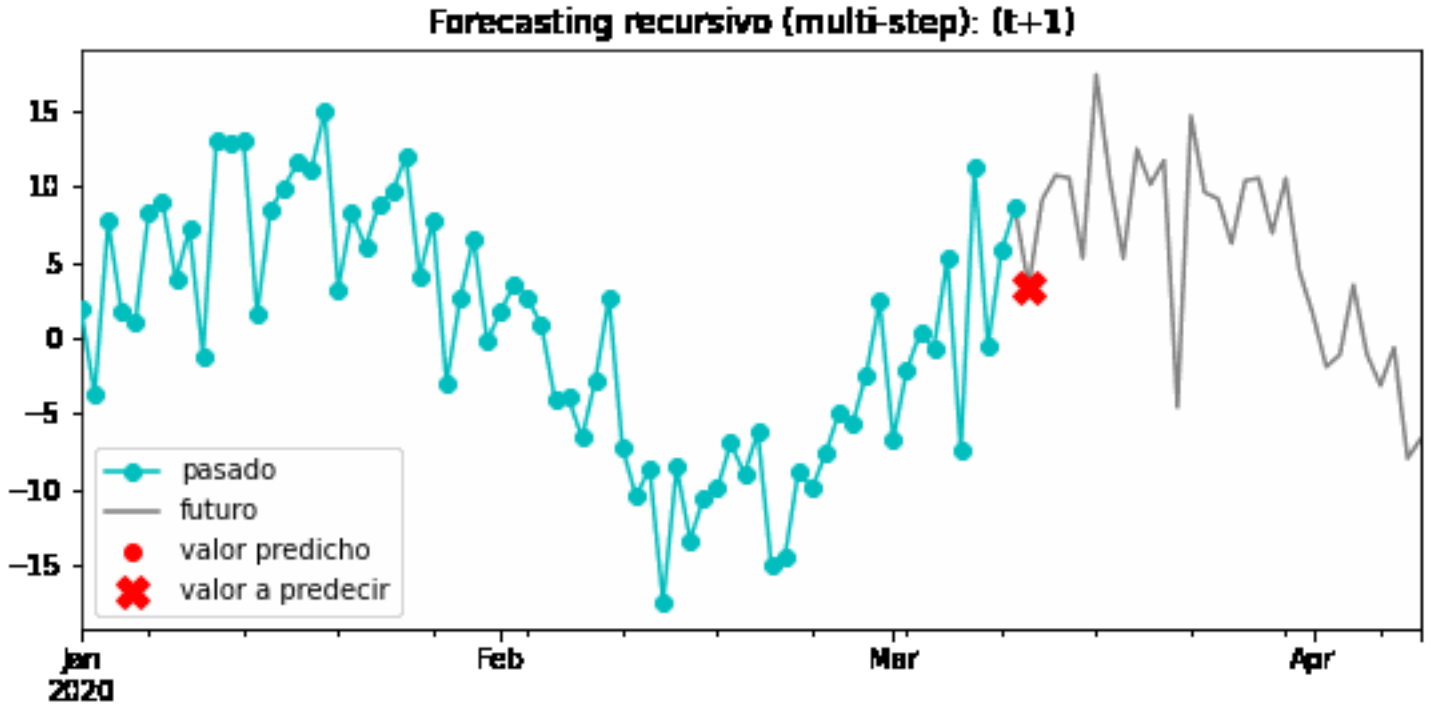
serie no estacionaria

¿Por qué son importantes las series de tiempo estacionarias?

La razón por la que estas series son importantes es que la mayoría de los modelos de series de tiempo funcionan **bajo el supuesto de que la serie es estacionaria**. Intuitivamente, podemos suponer que si una serie tiene un comportamiento particular en el tiempo, hay una probabilidad muy alta de que se comportamiento continúe en el futuro. Además, las teorías relacionadas con las series estacionarias son más maduras y más fáciles de implementar en comparación con series no estacionarias. A pesar de que el supuesto de que la serie es estacionaria se utiliza en muchos modelos, casi ninguna de las series de tiempo que encontramos en la práctica son estacionarias. Por tal motivo **la estadística tuvo que desarrollar varias técnicas para hacer estacionaria, o lo más cercano posible a estacionaria, a una serie.**

¿Qué es el Forecasting en una serie Temporal?

Una serie temporal (time series) es una sucesión de datos ordenados cronológicamente, espaciados a intervalos iguales o desiguales. **El proceso de forecasting consiste en predecir el valor futuro de una serie temporal**, bien modelando la serie únicamente en función de su comportamiento pasado (autorregresivo) o empleando otras variables externas.

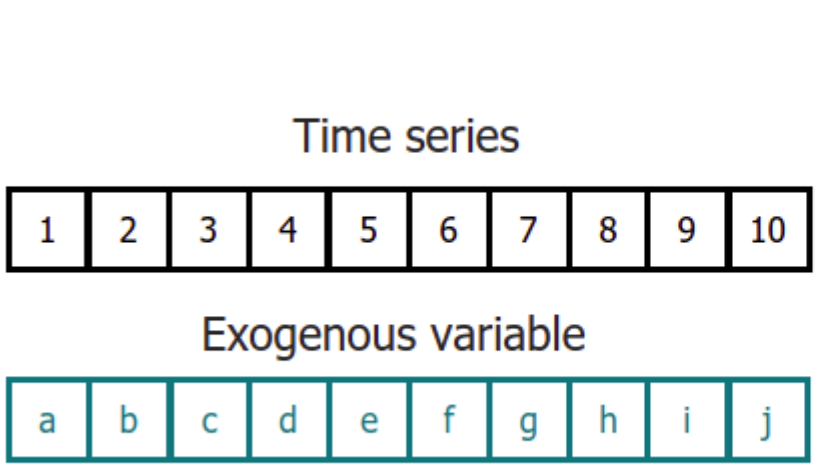

Forecasting recursivo (multi-step): (t+1)
15
10
5
0
−5
−10
−15
pasado
futuro
valor predicho
valor a predecir
Jan
2020
Feb
Mar
Apr

Entrenamiento de un modelo de forecasting

La principal adaptación que se necesita hacer para aplicar modelos de machine learning a problemas de forecasting es transformar la serie temporal en un matriz en la que, cada valor, está asociado a la ventana temporal (lags) que le precede.

1	2	3	4	5	6	7	8	9	10

Este tipo de transformación también permite incluir variables exógenas a la serie temporal.



					X	y
1	2	3	4	5	f	6
2	3	4	5	6	g	7
3	4	5	6	7	h	8
4	5	6	7	8	i	9
5	6	7	8	9	j	10

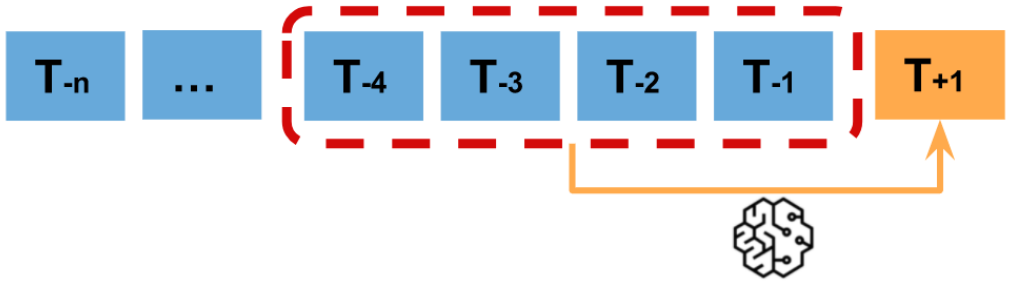
Una vez que los datos se encuentran reordenados de esta forma, se puede entrenar cualquier modelo de regresión para que aprenda a predecir el siguiente valor de la serie.

Predicciones multi-step

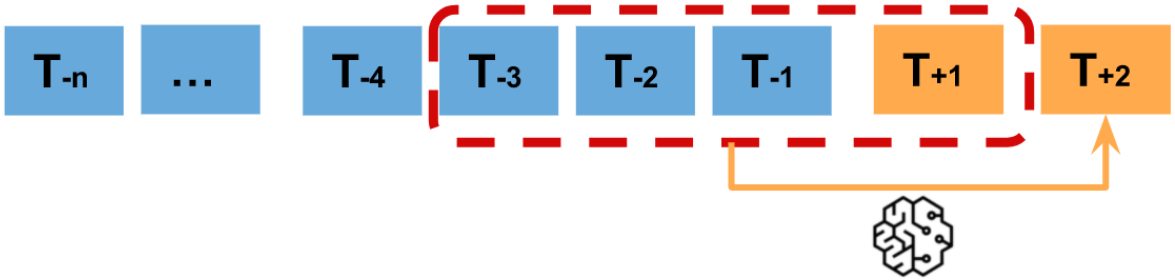
Cuando se trabaja con series temporales, raramente se quiere predecir solo el siguiente elemento de la serie (t+1), sino todo un intervalo futuro o un punto alejado en el tiempo (t+n). A cada paso de predicción se le conoce como *step*. Existen varias estrategias que permiten generar este tipo de predicciones múltiples.

Recursive multi-step forecasting

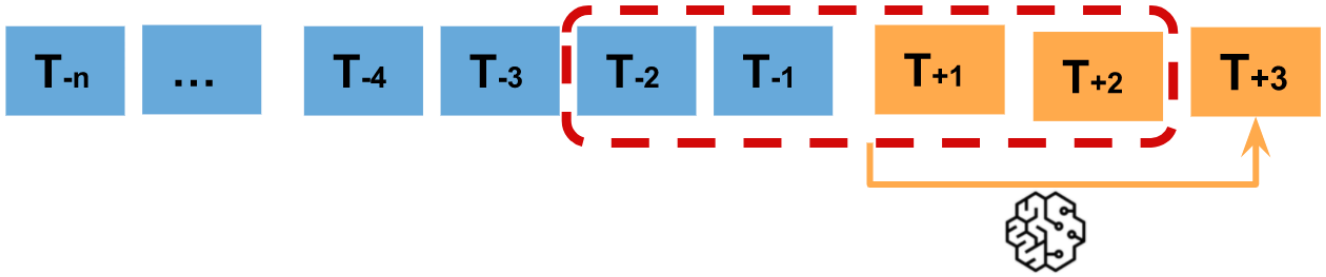
Dado que, para predecir el momento tn se necesita el valor de t(n−1), y t(n−1) se desconoce, se sigue un proceso recursivo en el que, cada nueva predicción, hace uso de la predicción anterior. A este proceso se le conoce como *recursive forecasting* o *recursive multi-step forecasting*



Predicción step 1



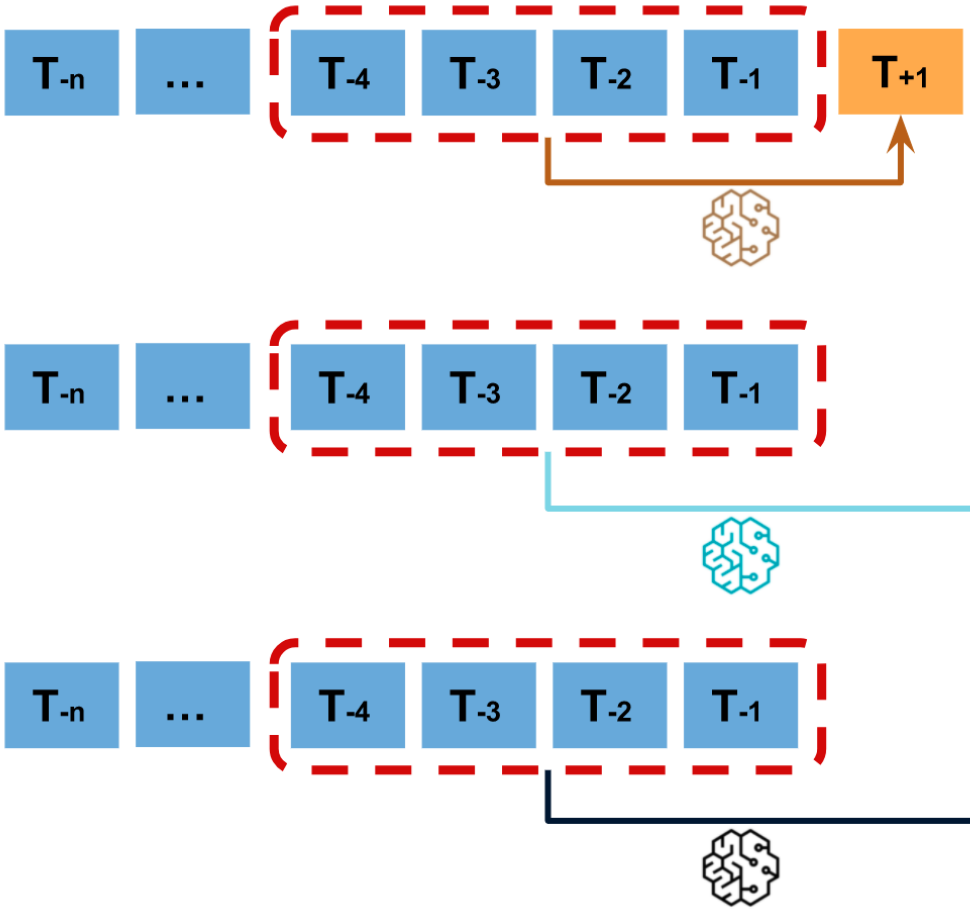
Predicción step 2



Predicción step 3

Direct multi-step forecasting

El método *direct multi-step forecasting* consiste en entrenar un modelo distinto para cada *step*. Por ejemplo, si se quieren predecir los siguientes 5 valores de una serie temporal, se entrenan 5 modelos distintos, uno para cada *step*. Como resultado, las predicciones son independientes unas de otras.



Predicción step 1

Modelo para step 1

Predicción step 2

Modelo para step 2

Predicción step 3

Modelo para step 3

La principal complejidad de esta aproximación consiste en generar correctamente las matrices de entrenamiento para cada modelo.

También es importante tener en cuenta que esta estrategia tiene un coste computacional más elevado ya que requiere entrenar múltiples modelos. En el siguiente esquema se muestra el proceso para un caso en el que se dispone de la variable respuesta y dos variables exógenas.

y	1	2	3	4	5	6	7	8	9
exog 1	A	B	C	D	E	F	G	H	I
exog 2	a	b	c	d	e	f	g	h	i

`lags=3, steps=2, create_train_X_y()`

y_train	
step 1	step 2
4	5
5	6
6	7
7	8
8	9

X_train						
X_lags			X_exog_1		X_exog_2	
lag 1	lag 2	lag 3	step 1	step 2	step 1	step 2
3	2	1	D	E	d	e
4	3	2	E	F	e	f
5	4	3	F	G	f	g
6	5	4	G	H	g	h
7	6	5	H	I	h	i

`filter_train_X_y_for_step(step=1)`

`filter_train_X_y_for_step(step=2)`

y_train
step 1
4
5
6
7
8

X_train				
X_lags			X_exog_1	X_exog_2
lag 1	lag 2	lag 3	step 1	step 1
3	2	1	D	d
4	3	2	E	e
5	4	3	F	f
6	5	4	G	g
7	6	5	H	h

Training data for model of step 1

y_train
step 2
5
6
7
8
9

X_train				
X_lags			X_exog_1	X_exog_2
lag 1	lag 2	lag 3	step 2	step 2
3	2	1	E	e
4	3	2	F	f
5	4	3	G	g
6	5	4	H	h
7	6	5	I	i

Training data for model of step 2

Modelos para series de tiempo

Queremos pronosticar la cantidad de viajes por hora del día de una app de alquiler de bicicletas. Para esto, utilizamos la variable dependiente o target (la cantidad de viajes) en un tiempo anterior como variable independiente o feature. Por ejemplo, la cantidad de viajes que se realiza un día a las 13 horas puede estar relacionada con la cantidad de viajes que se realizaron a las 13 horas de uno o dos días antes.

El tipo de modelos que sigue esta filosofía se llaman **AutoRegresivos o AR.** Utilizaremos una variante más avanzada llamada **SARIMA** (Seasonal AutoRegressive Integrated Moving Average). Estos modelos consideran además la estacionalidad, lo que permite incorporar el patrón repetitivo de cada día.

En Python existe una implementación del SARIMA en el módulo statsmodels. Este módulo posee además clases y funciones para la estimación de diferentes modelos estadísticos.

Esta familia de modelos suele denotarse como SARIMA(p,d,q)(P,D,Q)[m]. Los términos (p,d,q) se refieren a los términos autoregresivos, integrados y de media móvil respectivamente.

$$SARIMA(p,d,q) \times (P,D,Q)^s$$

p => El orden del Modelo Autoregresivo (AR).

d => Número de diferenciaciones aplicadas a la serie original

q => El orden del Moving Average model (MA).

P => El orden del Modelo Autoregresivo Estacional.

D => Número de diferenciaciones aplicadas a la serie original de manera Estacional

Q => El orden del Seasonal Moving Average model.

S => Número de pasos de tiempo que mide la estación de la serie.

El modelo genera una relación de recurrencia para y_t en el presente contra los valores pasados y_{t-1}, y_{t-2}, ... y_{t-p}:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots\ldots\ldots\ldots + \varphi_p y_{t-p} + \varepsilon_t$$

Para la componente AR

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots\ldots\ldots\ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

para la componente MA, aquí podemos notar los "p" parámetros en la componente **A**uto**R**egresiva AR del model y los "q" parámetros de la componente **M**oving **A**verage. Podemos combinar ambos modelos para obtener un modelo más robusto:

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

El cual en términos del "Lag Operator (L)" o operador de retraso, es decir que retorna el valor un paso anterior al paso de tiempo dado "t" o lo que es lo mismo **L * y_t = y_{t-1}**, el modelo puede ser leído como:

Proceso estocástico

- Un proceso estocástico es una secuencia infinita de variables aleatorias ..., y_{-3}, y_{-2}, y_{-1}, y_0, y_1, y_2, ... Lo representaremos abreviadamente como $\{y_t\}_{t=-\infty}^{\infty}$, o también $\{y_t\}_{-\infty}^{\infty}$, o simplemente $\{y_t\}$.
- Notación:

1. $E(y_t) = \mu_t$
2. $Var\left[y_t\right] = \gamma_{t,t}$
3. $Cov\left[y_t, y_s\right] = E\left[y_t - E\left(y_t\right)\right]\left[y_s - E\left(y_s\right)\right] = \gamma_{t,s}$, para $t \neq s$
4. $Corr\left[y_t, y_s\right] = \frac{Cov[y_t, y_s]}{\sqrt{Var[y_t]Var[y_s]}} = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}} = \rho_{t,s}$

Serie temporal

- Una serie temporal es una realización de un proceso estocástico. Como en un proceso estocástico hay infinitas realizaciones, una serie temporal es una entre todas las realizaciones posibles del proceso.

Estacionariedad

- Se dice que un proceso estocástico es estacionario en sentido débil (o débilmente estacionario, o estacionario en covarianza) si existen y son finitas las esperanzas, varianzas y covarianzas de las v.a. que forman el proceso y se cumple que:

1. $E[y_t] = \mu, \ \forall t$
2. $Var[y_t] = \gamma_0, \ \forall t$

1 ***Proceso puramente aleatorio o ruido blanco***

Proceso sobre el que no se puede predecir (ej. Lotería)

$$X_t = \varepsilon_t \begin{cases} E(\varepsilon_t) = 0 \\ Var(\varepsilon_t) = E(\varepsilon_t^2) = \sigma_\varepsilon^2 \quad \forall t \\ Cov(\varepsilon_t, \varepsilon_{t'}) = 0 \quad \forall t \neq t' \end{cases}$$

2 ***Proceso autorregresivo de orden p AR(p)***

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t$$

$$\phi(L^p) X_t = \varepsilon_t$$

❸ *Proceso media móvil de orden q MA(q)*

$$X_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

$$X_t = \theta\left(L^q\right)\varepsilon_t$$

❹ *Procesos mixtos ARMA(p,q)*

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \ldots - \phi_p X_{t-p} = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

$$\phi\left(L^p\right) X_t = \theta\left(L^q\right)\varepsilon_t$$

$$\Rightarrow X_t = \frac{\theta\left(L^q\right)}{\phi\left(L^p\right)}\varepsilon_t \quad o \quad \varepsilon_t = \frac{\phi\left(L^p\right)}{\theta\left(L^q\right)} X_t$$

Modelo ARIMA para la predicción de series de tiempo

ARIMA significa modelo de promedio móvil integrado autorregresivo y se especifica mediante tres parámetros de orden: *(p, d, q)*.

- **AR(*p*) Autoregresión** : un modelo de regresión que utiliza la relación dependiente entre una observación actual y las observaciones durante un período anterior Un componente auto regresivo(*AR(p)*) se refiere al uso de valores pasados en la ecuación de regresión para la serie de tiempo .
- **I(*d*) Integración** : utiliza la diferenciación de observaciones(restando una observación de la observación en el paso de tiempo anterior) para hacer estacionaria la serie de tiempo. La diferenciación implica la resta de los valores actuales de una serie con sus valores anteriores d número de veces.
- **Media móvil MA(*q*)** : un modelo que utiliza la dependencia entre una observación y un error residual de un modelo de media móvil aplicado a observaciones retrasadas. Un componente de media móvil representa el error del modelo como una combinación de términos de error anteriores. El orden *q* representa el número de términos que se incluirán en el modelo.

Tipos de modelo ARIMA

- **ARIMA:** medias móviles integradas autorregresivas no estacionales
- **SARIMA:** ARIMA estacional
- **SARIMAX:** ARIMA estacional con variables exógenas

Series de tiempo con Python

Las principales librerías que nos ofrece Python para trabajar con series de tiempo son:

Statsmodels: Esta librería contiene muchos objetos y funciones de suma utilidad para el análisis de series de tiempo. Algunos de los modelos que están cubiertos por Statsmodels incluyen: el modelo autorregresivo (AR); el modelo autorregresivo de vectores (VAR); y el modelo autorregresivo de media móvil (ARMA). También incluye funciones de estadística descriptiva de series de tiempo, como por ejemplo la autocorrelación, así como las correspondientes propiedades teóricas de ARMA o procesos relacionados. Por último, también ofrece las pruebas estadísticas relacionadas y algunas funciones auxiliares muy útiles.

Pandas: Pandas proporciona un amplio soporte para trabajar con datos de series de tiempo. Generalmente cuando trabajamos con series de tiempo realizamos un amplio abanico de tareas, como: convertir fechas, estandarizar el tiempo de acuerdo a la zona horaria, crear secuencias a determinados intervalos o frecuencias, identificar datos faltantes, desplazar las fechas hacia atrás o hacia adelante por un determinado valor, calcular resúmenes agregados de valores a medida que el tiempo cambia, etc. Pandas nos brinda las herramientas para poder realizar estas y muchas otras tareas en forma muy sencilla.

Skforecast es una biblioteca de Python que facilita el uso de regresores de scikit-learn como pronosticadores de varios pasos. También funciona con cualquier regresor compatible con la API scikit-learn (pipelines, CatBoost, LightGBM, XGBoost, Ranger...).