

ACÀMICA

¡Bienvenidos/as a Data Science!



Agenda

¿Cómo anduvieron?

Repaso: Clases

Explicación: Scikit-Learn y Transformación de Datos con Scikit-Learn

Hands-on training

Break

Aplicaciones Prácticas

Actividad: Explorando Mis Datos

Cierre



¿Cómo anduvieron?



Repaso: Clases



¿Qué es un objeto?

Un objeto es como un paquete de variables y funciones que conviene tener agrupados por consistencia y comodidad.

Persona_nombre = 'Juan'

Persona_edad = 23

Persona_profesion = 'vendedor'



Objeto

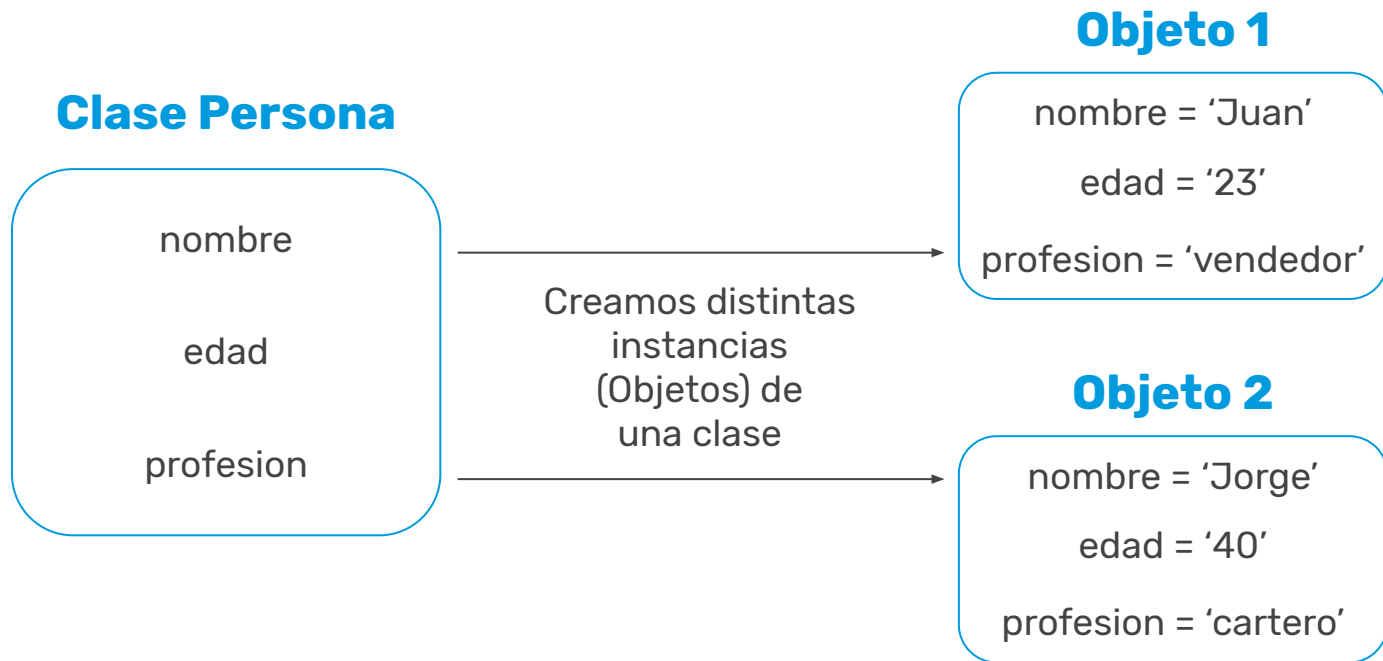
nombre = 'Juan'

edad = 23

profesión = 'vendedor'

¿Qué es una clase?

Los objetos suelen crearse a partir de unas *plantillas* a las que llamamos clases.



Métodos

Los métodos son funciones propias de una clase. Es decir son funciones que definimos para que actúan sobre un tipo de objeto determinado.

```
[15]: class Persona:
    """
    Esta es una clase donde se agregan todos los datos
    respecto a una persona
    """
    def __init__(self, nombre, edad):
        # Todo lo que definamos en __init__ se corre
        # al crear una instancia de la clase
        self.nombre = nombre
        self.edad = edad
    def mePresento(self):
        print("Hola, me llamo " + self.nombre)
```




Métodos: Beneficios

La programación orientada a objetos nos propone una **manera de trabajar** a la hora de escribir nuestro programa.

No sólo es práctico y ordenado, sino que también muchas veces nos ayuda a mantener la consistencia del código.

```
[1]: class Departamento:
      def __init__(self, calle, altura, sup_total, sup_cubierta):
          self.calle = calle
          self.altura = altura
          self.sup_total = sup_total
          if sup_cubierta < sup_total:
              self.sup_cubierta = self.sup_cubierta
          else:
              self.sup_cubierta = self.sup_total

[3]: depto_1 = Departamento('Humboldt', 1122, 50, 455)
      depto_1.sup_cubierta
```

```
[3]: 50
```

Scikit-Learn



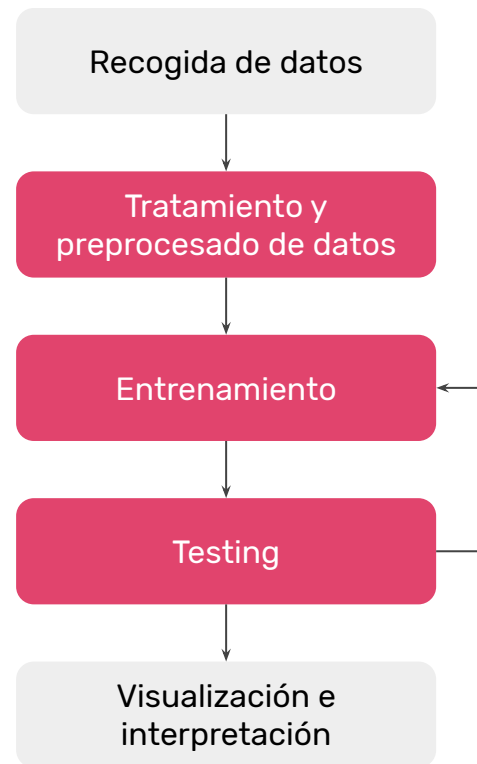
Clases: Beneficios

Hoy veremos que el formato de clases y métodos propuestos por la librería scikit-learn es muy útil y es el más usado en la comunidad de data science.



SCIKIT learn

- Librería base para Machine Learning en Python
- Podemos usarla para:
 - Preprocesamiento de datos
 - Modelos de Clasificación y Regresión
 - Métricas de Evaluación
- *Fuerza a seguir un Pipeline*
- Excelente [documentación](#)



SCIKIT learn: ¿Cómo se usa?

La API (application programming interface) de Scikit-Learn trabaja con **Objetos** y Clases, donde se implementa de manera uniforme métodos y atributos de esos objetos.

Los principales objetos son:

- **Estimadores (*estimators*)** → Tienen un método *.fit*
- **Predictores (*predictors*)** → Tienen un método *.predict*
- **Transformadores (*transformers*)** → Tienen un método *.transform*
- **Modelos (*model*)** → Tienen un método *.score*

¡La mayoría de los objetos pertenecen a más de un tipo!



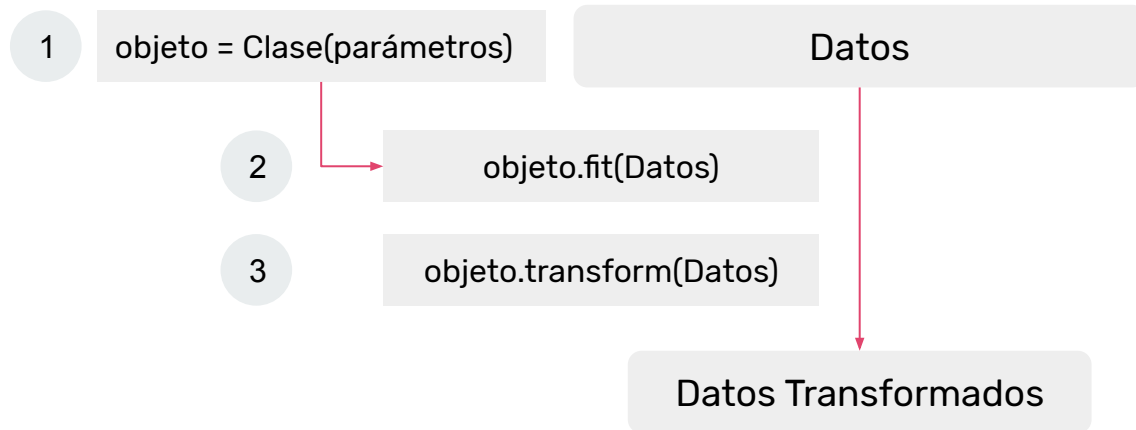
¡Qué abstracto!

**¿Y la explicación en
español cómo sería?**

Veamos un ejemplo

En general:

1. Se crea un *objeto* con ciertos *parámetros*.
2. Se implementa un método *.fit* que aprende de los datos
3. Se implementa un método *.transform* que transforma los datos.





Transformación de Datos con Scikit-Learn



¿Qué aprendimos de los videos?

Para preprocesar datos, Scikit-Learn nos provee de las siguientes herramientas:

Imputer

Para rellenar valores faltantes

OneHotEncoder

Para pasar de variables categóricas a dummies. ¿Sobre qué tipos de datos lo usaremos?

LabelEncoder

Para pasar de variables categóricas a valores numéricos (cuando no necesitemos usar OneHotEncoder).

Notar que cada herramienta es, desde el punto de vista de la programación, una clase. Prestar atención a sus parámetros y sus métodos.

¿Qué aprendimos de los videos?

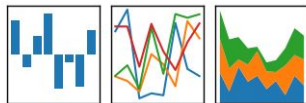
Nota: algunas funcionalidad de la librería fueron actualizadas desde que se hicieron los videos. Recomendamos ir siempre a la documentación. Por ejemplo, el Imputer ahora está como SimpleImputer en el módulo `sklearn.impute`

¿Cuándo usar cada una?

La realidad es que usar una u otra librería es una cuestión de gustos, y la mayoría de las veces usamos las dos.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



vs



-Generalmente más útil durante EDA

- Generalmente más útil en un flujo de Machine Learning

- Algunas funcionalidades 'aprenden solas' de los datos

Hands-on training



Instalar Scikit-Learn y luego

DS_Clase_12_TDD_Sklearn.ipynb



A close-up photograph of a white ceramic cup filled with a latte. The coffee has a thick layer of white foam with intricate brown latte art designs, including a heart shape and concentric circles. The cup sits on a matching white saucer. In the background, a white napkin with a silver fork is visible, all set against a dark, textured surface.

¡BREAK!

Ph. Credit: Drew Coffmann



Aplicaciones prácticas



Actividad: Explorando mis datos



Paso 1 • Elegir un dataset propio o de alguno de los siguientes links

1. **Kaggle**: probablemente el sitio más famoso para obtener datasets (en formatos bastante amigables) y participar de competencias.
¡Recomendamos mirar las competencias!
2. **Datos abiertos de Presidencia Argentina**
3. **Datos abiertos de la Ciudad de Buenos Aires**
4. Y muchos lugares más. Si conocen, no duden en compartir, recomendar, etc.

Paso 2 • Una vez elegido el dataset...

¡Comiencen a explorarlo con las herramientas que aprendimos!

¿Qué preguntas me *gustaría* responder? ¿Qué preguntas *podrán* responder con ese dataset? **¡Lamentablemente, ambas no siempre coinciden!**

TIP: No sientan que se tienen que casar con ese dataset, después si quieren pueden cambiar. Pero si eligen uno que realmente los interpele, van a poder trabajar con esos datos durante la carrera.

Recordatorio sobre condiciones de aprobación



Proyectos • Condiciones de aprobación

Los/as evaluadores/as considerarán una entrega como **Aprobada** cuando el/la estudiante haya cumplido satisfactoriamente el 100% de los puntos que pide el checklist (aunque no los haya hecho a todos perfectos).

Caso contrario, el/la evaluador/a considerará la entrega como **Para rehacer.**

Nota: no hay un límite de iteraciones (el/la estudiante puede tener que rehacer su trabajo todas las veces que sea necesario hasta aprobar).

Para la próxima

1. Ver los videos de la plataforma “Transformación de Datos: Detección de Outliers”.
2. Trabajar en la Entrega 02.
3. Seguir explorando el dataset que eligieron.
4. Completar el notebook de hoy si no lo terminaron.

ACÀMICA