

FUNDACIÓN  
**YPF**

¡Bienvenidas a Ingenias+!  
**Data Science**





**¿Ponemos a grabar  
el taller?**



# Módulo I | Clase 1

## Introducción a Data Science





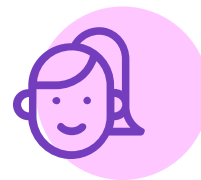
# ¿Nos presentamos?

## El equipo



Docente

**Carolina Allasia**  
mini bio



Tutora

**Maria Belén**  
Laresca

**Mini bio**



# Cronograma

## ¿Qué haremos en el curso?



- Vamos a conocer qué hace un Data Scientist y qué habilidades se necesitan para desarrollarse en este campo.
- Vamos a aprender conceptos fundamentales de Python y Machine Learning.

### Estructura del curso

Teoría  
Práctica  
Proyecto Final

### ¡Speak Up!

Las invitamos a presentarse en el siguiente padlet.  
¡También pueden abrir el micrófono y contarnos sobre ustedes!

## En esta clase vamos a ver...

- Qué es Data Science.
- Cuales son las habilidades de un Data Scientist.
- Etapas de un proyecto de Data Science.
- Librerías científicas de Python más usadas.
- Jupyter Notebook y Anaconda, que son, como instalarlos y crear un espacio de trabajo.
- PRIMERA PARTE DE INTRODUCCIÓN A PYTHON:
  - ¿Qué es Python?
  - Sintaxis de Python
  - Dato, Variables y Operadores





# ¿Qué es Data Science?

## ¿Qué es Data Science?

¿Qué crees que es Data Science?

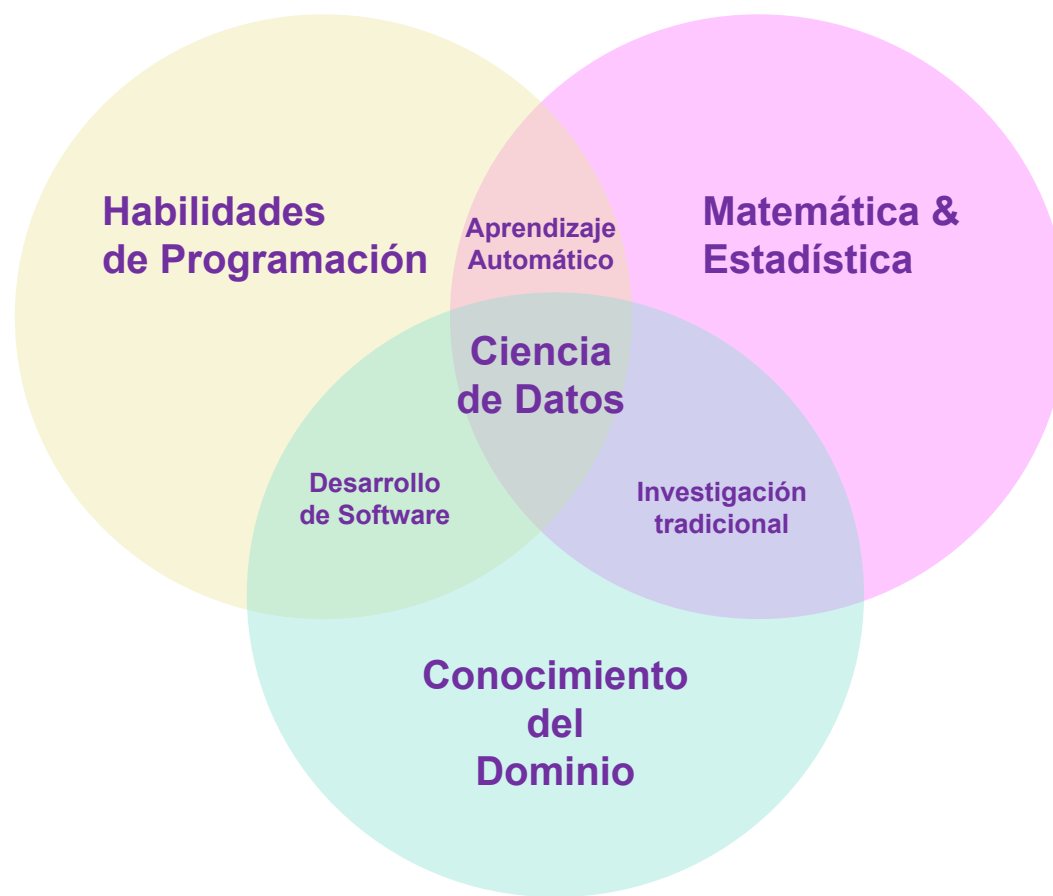
¿Qué te motiva a estudiar Data Science?

## ¿Qué es Data Science?

Data Science es **extraer conocimiento útil de nuestros datos.**



## ¿Qué es Data Science?



### **Sistemas de recomendación**

Spotify, Facebook, Netflix,  
Amazon, Mercado Libre, Google.

### **Predicción de tiempo de viaje**

Google Maps, Waze, Uber,  
Cabify.

### **Reconocimiento del habla**

Siri, Cortana, Google Now,  
Amazon Echo, Alexa.

### **Detección de fraude**

PayPal, Mercado Libre,  
Bancos.

### **Diálogo y Generación de Texto**

ChatGTP.

### **Clasificación de mensajes**

Google Maps, Waze, Uber,  
Cabify.

### **Chatbots**

E-commerce, servicio al  
cliente, bancos.



# ¿Qué skills necesita un Data Scientist?

## Skills en DS

### Conocimientos de programación y base de datos

- Pensamiento computacional
- Python y/o R
- Bases de datos, SQL y no-SQL
- Cloud

### Matemática y Estadística

- Conceptos de machine learning
- Modelado estadístico
- Diseño de experimentos
- Inferencia Bayesiana
- Aprendizaje supervisado y no supervisado
- Álgebra lineal, Optimización

### Comunicación

- Storytelling: Habilidad de contar historias con datos
- Traducir conceptos complejos según la audiencia

### Conocimiento del dominio y habilidades blandas

- Curiosidad
- Trabajo independiente
- Pensamiento analítico
- Resolutivo
- Proactivo, estratégico, creativo y colaborativo



# Etapas de un proyecto de DS

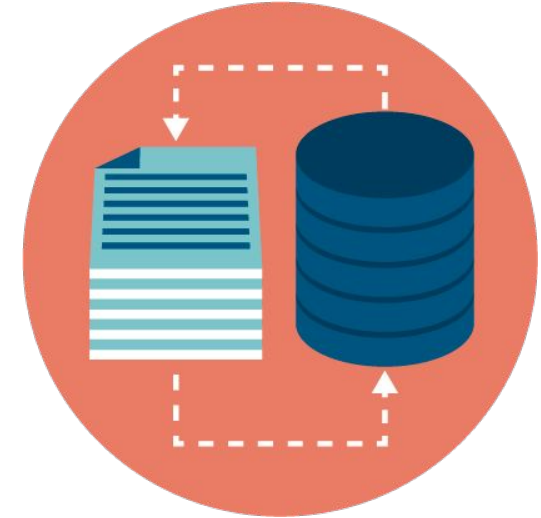


## Etapas de un proyecto

- 01 Recolección de datos
- 02 Exploración y procesamiento
- 03 Modelado
- 04 Puesta en producción

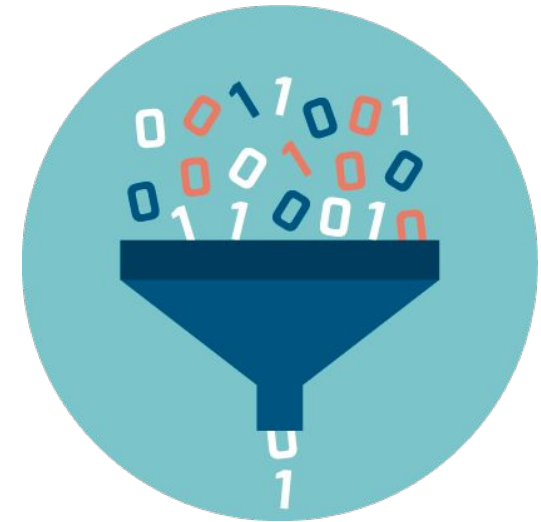
## 1- Recolección de datos

- Obtención de los datos mediante sensores, scrapeo de la web, peticiones a APIs, formularios.
- Creación de base de datos.



## 2- Exploración y procesamiento

- Exploración de los datos.
- Discreción de variables.
- Normalización.
- Limpieza.
- Visualización previa.



### 3- Modelado

- Construir y testear modelos para predecir o clasificar información o encontrar patrones en los datos.



## 4- Puesta en producción

- Predecir nuevos datos.
- Comunicar los resultados.
- Integrar los resultados con aplicación.





# Descanso

Nos vemos en 5 minutos



# Herramientas Utilizadas

## Python

- Lenguaje Interpretado de propósito general.
- Sintaxis de código legible: Fácil aprendizaje/depuración.
- Muchas librerías desarrolladas para la manipulación y visualización de datos, e implementación de algoritmos de ML.



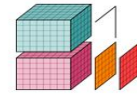
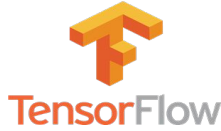


## Dataset

- Dataset es el conjunto de datos que se utiliza en el flujo de trabajo de data science.



## Herramientas de Python



### Numpy

Estructura de datos: Array

Operaciones eficientes sobre los datos: mayor velocidad y menor espacio.

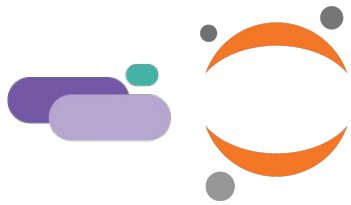
### Pandas

Estructura de datos: DataFrame

Permite trabajar con datasets con datos de distintos tipos y realizar operaciones sobre filas y columnas por nombre.

### Scikit Learn

Librería que facilita la manipulación, limpieza, preparado de datos para algoritmos de Machine Learning como así también tiene implementaciones de los algoritmos más comunes.



## Jupyter Notebook

- Ejecución de código Python/R
- Visualización de resultados
- Ecuaciones (LaTeX)
- Compartir resultados
- Markdown



## Anaconda

Es un gestor de paquetes y entornos

- Gestor de entornos: Permite compartimentalizar herramientas o bibliotecas para cada proyecto que hagamos
- Gestor de paquetes: Sobre todo en el caso de Windows, nos permite instalar y manejar paquetes y librerías de una manera fácil y efectiva.





# Sección práctica

Configurar el entorno de Jupyter  
Notebook y Python

## Anaconda

- 01 Visitamos la página de conda: <https://www.anaconda.com/>
- 02 Hacemos click en Get Started y luego en Download
- 03 Elegimos qué sistema operativo tenemos
- 04 Bajamos el Graphical Installer de Python 3.8

## Anaconda (Si tenemos Windows)

- 05 Cuando termine de bajar. Abrir y correr la instalación.
- 06 Aceptamos todas las licencias. Y destildamos “Add Anaconda to my PATH environment variable”.
- 07 Abrimos Anaconda Prompt.
- 08 Se abrirá una terminal.

## Anaconda (Si tenemos Mac)

- 05 Abrimos el archivo .pkg
- 06 Seguimos las instrucciones que nos indica la ventana.
- 07 Una vez finalizada la instalación, abrimos una terminal. (Desde las aplicaciones buscar Terminal)



## Requerimientos

01

Python

Se instala al instalar conda

02

Pandas y Numpy

conda install pandas

conda install numpy

03

Matplotlib y Seaborn

conda install matplotlib

conda install seaborn

04

Scikit-learn

conda install sklearn

## Como abrir Jupyter Notebook

- 01 Abrimos Anaconda Navigator.
- 02 Se abrirá una ventana. Elegimos Jupyter Notebook.
- 03 Se abrirá una pestaña en el navegador. Luego clicar en Nuevo y seleccionar Python3.
- 04 Se abrirá un nuevo notebook.



# Actividad práctica

Configuración del entorno.

## Trabajamos en clase

### Configuración del entorno

Siguiendo los pasos anteriores, cada alumna comenzará a configurar su entorno.

 *20 minutos de actividad.*





# Desafío 1

Para la siguiente clase de les pediremos que:  
**Tengan instalado en su computadora anaconda  
y configurado el entorno.**



# Proyecto Final

## Proyecto Final

- 01 Conformar grupos de trabajo.
- 02 Elección del tema.
- 03 Investigar y Seleccionar datasets apropiados para el tema elegido.
- 04 Creación del repositorio y configuración del ambiente de trabajo.

## Proyecto Final

- 05 Análisis Exploratorio de Datos.
- 06 Feature Engineering.
- 07 Creación y Entrenamiento de Modelo(s).
- 08 Presentación de resultados (Storytelling).

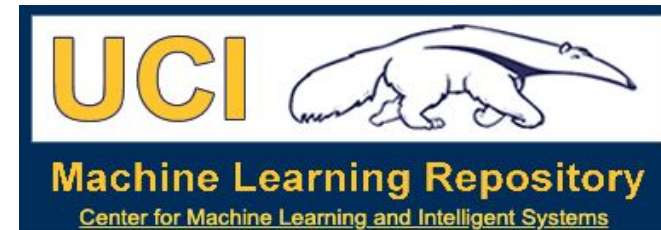


## Proyecto Final

Donde obtener Datasets

kaggle

Google  
Dataset Search Beta



... Y en otros bancos de datos abiertos.



# Proyecto Final

## ¿Cómo se verá el proyecto final?

master

1 branch

0 tags

Go to file

Add file

<> Code

## Analysis of Twitter

Read about this project in Medium:

- How to build a PostgreSQL database to store tweets
- Keras Challenges the Avengers
- Visualizing interactions with NetworkX

## Introduction

Twitter is used every day by people to express their feelings or thoughts, specially about something that it is happening at the moment or has just occurred; by companies to promote products or services; by journalists to comment on events or write news; and the list can go on. There is no doubt that analyzing twitter and tweets is a powerful tool that can give us a sense on the public opinion about a topic that we are interested in. Fortunately,

master • [nlp\\_analysis\\_twitter / Jupyter Notebook files / Interaction Network.iynb](#)

Go to file

ugis22 Analysis of Twitter

Latest commit 649c69 on May 3, 2019 [History](#)

A1 contributor

955 Lines (955 sloc) | 823 KB

<> | Raw | Blame | [Edit](#) | [Copy](#)

### Read json to DataFrame

The information that we've collected is stored in the file `tweets.txt`. Because this file has a JSON format, we'll take advantage of the `read_json` function of the pandas module.

```
In [2]: pd.set_option('display.float_format', lambda x: '%.1f' % x)
```

```
In [3]: # Read json into a pandas dataframe
tweets_df = pd.read_json('tweets.txt', lines=True)
```

According to [Twitter API website](#), the Tweet object retrieved, provided in JSON format, has a long list of mixed root-level attributes, including basic information such as `id`, `created_at`, and `text`. Tweet objects are also the parent object to several child objects. Tweet child objects include `user`, `entities`, and `extended_entities`.

In order to have a better idea of the information we are dealing with, let's take a look at the `DataFrame` columns.

```
In [4]: # Let's check the name of the columns
tweets_df.columns
```

```
Out[4]: Index(['contributors', 'coordinates', 'created_at', 'display_text_range',
              'entities', 'extended_entities', 'extended_tweet', 'favorite_count',
              'favorited', 'filter_level', 'geo', 'id', 'id_str',
              'in_reply_to_screen_name', 'in_reply_to_status_id',
              'in_reply_to_status_id_str', 'in_reply_to_user_id',
              'in_reply_to_user_id_str', 'is_quote_status', 'lang', 'place',
              'possibly_sensitive', 'quote_count', 'quoted_status',
              'quoted_status_id', 'quoted_status_id_str', 'quoted_status_permalink',
              'reply_count', 'retweet_count', 'retweeted', 'retweeted_status',
              'source', 'text', 'timestamp_ms', 'truncated', 'user'],
              dtype='object')
```

From the displayed columns, we can observe some of them that look interesting for our future analysis. If we look more in detail, certain columns, such as `retweeted_status`, `entities` provide us with information about interactions regarding user mentions and user retweets. So, we are going to create a new `DataFrame` where to store all this important information that will come in handy when we build our Network.



# ¿Alguna consulta?



# Descanso

Nos vemos en 10 minutos



## Módulo II | Clase 1

# Python para Data Science. Introducción a Python





# ¿Qué es Python?

## Python

PYTHON es un lenguaje de programación de **propósito general**, creado en 1991 por Guido Van Rossum.

Actualmente es **ampliamente** utilizado en Data Science y Machine Learning debido a sus características:

- Lenguaje de alto nivel, interpretado o de script
- De tipado dinámico
- Fuertemente tipado
- Multiparadigma y Multiplataforma
- De código abierto



## Sintaxis de Python

La clave de la sintaxis de Python es la **INDENTACIÓN**.

La indentación es la sangría inicial de un bloque de código, compuesta por 4 espacios o un tab.



0      4      8

```
def hello_world(word):  
    if word == 'Hello':  
        print(f"{word} World!")  
    else:  
        print(f"Goodbye")
```



## Sintaxis de Python

Los comentarios permiten documentar y explicar el código.


En Python hay 3 tipos: de una **línea**, de **media línea** o de **múltiples líneas**.



```
def comentarios(word):  
    assert isinstance(word, str)  
    '''  
    Eso es un comentario  
    de  
    multiples lineas  
    '''  
    if word == 'Hello': # esto es de media linea  
        print('Hola')  
    else:  
        # Esto es un comentario de linea  
        print('Chau')
```

## Sintaxis de Python

Si un comentario está justo al comienzo de la definición de una función, clase o bucle, y detalla lo que se ejecutará en ese bloque de código, se lo denomina **docstring**.



```
def comentarios(word):  
    """  
    Esto es un docstring  
    y sirve para documentar  
    que hace la funcion, clase, etc  
  
    :param word: esto es el input
```



# Variables y Operadores

## Dato

Expresión general que describe las características de una entidad sobre la que se opera. Es la **mínima** parte de la **información**.



## Tipos de Datos en Python

### Primitivos

- INT
- FLOAT
- COMPLEX
- STR
- BOOL

### Colecciones

- LIST
- TUPLE
- DICT
- SET

## Primitivos en Python

### INT

Son números **enteros**. No poseen parte flotante luego de la coma.

➔ 18

### STR

Es una **cadena de caracteres**, que pueden contener números, letras y/o símbolos.

➔ "Esto es 1 string"

### FLOAT

Son números con **parte entera y parte flotante**.

➔ 38.78

### BOOL

Son valores booleanos que representan un **valor de verdad**.

➔ True  
False

### COMPLEX

Son números con parte real y parte imaginaria.

➔ 3 + 8i

## Colecciones en Python

### LIST

Es un conjunto **ordenado y mutable** de elementos (números, strings, etc), a los que se accede por un índice.

➡ `[1, "Hola", True]`

### TUPLE

Es un conjunto **ordenado e inmutable** de elementos, a los que se accede por un índice.

➡ `(1, 2)`

### DICT

Es un conjunto no ordenado (Desde Python 3.7+ los elementos son insertados en orden) y mutable de asociaciones clave-valor.

➡ `{"Persona": [1, 2]}`

### SET

Es un conjunto **no ordenado de elementos únicos**

➡ `Set(1, 2, "Rojo")`

## Variable

Espacio de **memoria** que ocupa un dato al **almacenarse**.





# identificador = valor



Debe ser **corto** y **representativo**.  
Puede estar compuesto por letras,  
números y underscores pero no puede  
empezar con un número. Se deben  
**evitar** las **palabras reservadas**.



Es cualquier tipo de **dato**  
admisible en Python con su  
sintaxis correcta.

**identificador = valor**

```
esto_es_un_identificador = [1, 2, "Hola", True]
```

## Operador

Símbolo que se **aplica** a uno o varios **datos** o variables en una **expresión** con el fin de obtener cierto **resultado**.



## Operadores

### Aritméticos

- +
- -
- \*
- \*\*
- /
- //
- %

### De asignación

- =
- +=
- -=
- \*=
- \*\*=
- /=
- //=
- %=

## Operadores

### Relacionales

- ==
- !=
- <
- >
- <=
- >=

### Lógicos

- and
- or
- not

## Operadores

```
numero_1 = 304
numero_2 = 989

sumatoria = numero_1 + numero_2

check = numero_1 != numero_2

otro_check = (numero_1 + 2 != numero_2) and (numero_1 * 2 == numero_1 ** 2)
```



¿Alguna consulta?

FUNDACIÓN  
**YPF**

¡Muchas gracias!

