# UNIVERSIDAD SANTO TOMÁS

## PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

**VIGILADA MINEDUCACIÓN** - SNIES 1732

Acreditación Institucional Internacional

OTORGADA POR EL IAC CINDA ACUERDO 55 DEL 9 DE MAYO-VIGENCIA 5 AÑOS

ACREDITACIÓN INSTITUCIONAL DE ALTA CALIDAD MULTICAMPUS

Res. MEN No. 01456 del 29 de enero de 2016

Vigencia por seis años

QS STARS RATED FOR EXCELLENCE

ISO 9001 icontec SC4289-1

CERTIFIED IQNet MANAGEMENT SYSTEM

UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

**Faculty: systems engineer**

**Course: Deep Learning**

**Topic: p1-Introducción – de ciencia de datos (Numpy & Pandas)**

---


**Professor: Luis Fernando Castellanos Guarin**

**Email:** Luis.castellanosg@usantoto.edu.co

**Phone:** 3214582098


**Github:** https://github.com/luisFernandoCastellanosG/Machine_learning

# CONTENIDO

1. ¿Qué es el Data Science?
2. Librerías estándar y especializadas para Python
3. Numpy
4. pandas.

¡Siempre hacia lo alto!

# ¿Qué es el Data Science?

El **Data Science** es un campo interdisciplinario que involucra:

- Métodos científicos
- Estadística y Matemática,
- programación (PYTHON, R, C, etc.), Software (libre) para tratamiento, Machine Learning y visualización de resultados.

Para extraer conocimiento, mejorar el entendimiento de datos, crear algoritmos para la minería de datos, el aprendizaje automático y la analítica predictiva.

# Librerías estándar Ptyhon

Python permite trabajar con funcionalidades que permiten optimizar y minimizar la escritura de código fuente, para utilizarlas se utiliza la siguiente estructura:

```
import modulo # importar un módulo
import paquete.modulo1 # importar un módulo que está dentro de un paquete
import paquete.subpaquete.modulo1 # importar un módulo que está dentro de un subpaquete

# Si las rutas (lo que se conoce como "namespace" son largas,se pueden generar alias por medio del modificado "as"
import modulo as m
import paquete.modulo1 as pm
import paquete.subpaquete.modulo1 as psm
```

# Librerías estándar

Python viene con una biblioteca de módulos estándar, sobre la podemos encontrar toda a información en The Python Standard Library.

Si queremos profundizar sobre la sintaxis y la semántica también nos vendrá bien tener a mano The Python Language Reference.

Por ejemplo, el módulo **OS** ofrece las típicas funciones que permiten interactuar con el sistema operativo

```
import os
os.getcwd() #directorio de trabajo actual
os.chdir('/server/accesslogs') #cambia el directorio
os.system('mkdir today') #ejecuta el comando 'mkdir'
dir(os) #lista de todas las funciones del módulo
help(os) #devuelve un manual de ayuda
```

# Librerías estándar

**Entornos virtuales:**

Para algunos proyectos vamos a necesitar unas versiones especializadas de algunos paquetes o librerías, para ello lo mejor es usar los ENTORNOS VIRTUALES y así no genere conflictos con versiones más recientes.

https://docs.python.org/es/3/tutorial/venv.html

# Librerías Especializadas

## Librerías no estándar

Dado que el objetivo de nuestra asignatura es desarrollar e implementar algoritmos de machine Learning / Deep Learning con Python a unos determinados **dataset's**, para eso vamos a necesitar algo más que las librerías de la biblioteca estándar:

- *NumPy*
- *SciPy*
- *Matplotlib*
- *Seaborn*
- *Pandas*
- *Scikit Learn*
- *Statsmodels*
- *Scrapy*
- *SymPy*
- *Nltk*
- *Stopwords*
- *Pyprind*
- *Tweepy*
- *Tensorflow*
- *OpenCv*

# Librerías Especializadas

- **NumPy : Acrónimo de *Numerical Python*.** Su características más potente es que puede trabajar con matrices (array) de n dimensiones. También ofrece funciones básicas de algebra lineal, transformada de Fourier, capacidades avanzadas con números aleatorios, y herramientas de integración con otros lenguajes de bajo nivel como Fortran, C y C++
- **SciPy: Acrónimo de *Scientific Python***. SciPy está construida sobre la librería NumPy. Es una de las más útiles por la gran variedad que tiene de módulos de alto nivel sobre ciencia e ingeniería, como transformada discreta de Fourier, álgebre lineal, y matrices de optimización
- **Matplotlib: es una librería de gráficos**, desde histogramas, hasta gráficos de líneas o mapas de calor.
- **Seaborn: basada en matplotlib**, se usa para hacer más atractivos los gráficos e información estadística en Python
- **Pandas**: se utiliza para operaciones y manipulaciones de datos estructurados. Es muy habitual usarlo en la fase de depuración y preparación de los datos.
- **Scikit Learn para machine learning:** Construida sobre NumPy, SciPy y matplotlib, esta librería contiene un gran número de eficientes herramientas para machine learning y modelado estadístico
- **Statsmodels:** para modelado estadístico. Es un módulo de Python que permite a los usuarios explorar datos, hacer estimaciones de modelos estadísticos y realizar test estadísticos

# Librerías Especializadas

- **Scrapy:** se usa para rastrear la web. Es un entorno muy útil para obtener determinados patrones de datos.
- **SymPy:** se usa para cálculo simbólico, desde aritmética, a cálculo, álgebra, matemáticas discretas y física cuántica.
- **Nltk (Natural language toolkit ):** es la biblioteca más popular para el procesamiento del lenguaje natural (NLP) que fue escrita en Python
- **Stopwords**: Una vez instalado **nltk** debemos descargar los diccionarios de palabras que necesitamos, para este caso las stopwords, que son las palabras conectoras que repetimos con frecuencia en un idioma
- **Pyprind (Indicador de progreso de Python):** proporciona una barra de progreso y un objeto de indicador de porcentaje que le permiten realizar un seguimiento del progreso de una estructura de bucle u otro cálculo iterativo.
- **Tweepy:** es una librería de **Python** que nos va a permitir realizar acciones en TWITTER (descargar tweets, publicar, borrar o modificar un tweet, etc)
- **Tensorflow:** librería que permite crear modelos de aprendizaje automático enfocado en el DEEP LEARNING (REDES NEURONALES).
- **OpenCv:** Librería en PYTHON o C, diseñada para  visión artificial.

# Instalando librerías

Pandas es una librería Python especializada en el tratamiento, manipulación y análisis de datos.

Estos datos se organizan en **FORM** de un **DATAFRAME**, que es una tabla organizada, como un archivo CSV.

Para instalar Pandas, es suficiente con hacer:

Colab-> pip install pandas

S.O-> pip3 install pandas

# Conociendo las versiones de librerías

Pandas es una librería Python especializada en el tratamiento, m

```
# Python version
import sys
print('Python: {}'.format(sys.version))
# scipy
import scipy
print('scipy: {}'.format(scipy.__version__))
# numpy
import numpy
print('numpy: {}'.format(numpy.__version__))
# matplotlib
import matplotlib
print('matplotlib: {}'.format(matplotlib.__version__))
# pandas
import pandas
print('pandas: {}'.format(pandas.__version__))
# scikit-learn
import sklearn
print('sklearn: {}'.format(sklearn.__version__))
```

# Pandas

Las características de la biblioteca PANDAS son:

- El tipo de datos son DataFrame para manipulación de datos con indexación integrada. Tiene herramientas para leer y escribir datos entre estructuras de dato en memoria y formatos de archivos variados

- Permite la alineación de dato y manejo integrado de datos faltantes, la reestructuración y segmentación de conjuntos de datos, la segmentación vertical basada en etiquetas, indexación elegante, y segmentación horizontal de grandes conjuntos de datos, la inserción y eliminación de columnas en estructuras de datos.

- Puedes realizar cadenas de operaciones, dividir, aplicar y combinar sobre conjuntos de datos, la mezcla y unión de datos.

- Permite realizar indexación jerárquica de ejes para trabajar con datos de altas dimensiones en estructuras de datos de menor dimensión, la funcionalidad de series de tiempo: generación de rangos de fechas y conversión de frecuencias, desplazamiento de ventanas estadísticas y de regresiones lineales, desplazamiento de fechas y retrasos.

# Pandas

Leyendo Dataset’s:

```
import pandas as pd

df = pd.read_csv("……./prestamos_records.csv")

Df.head()
```

Insertando nuevas columnas:

```
df.insert(0, 'col_1', df.mean(1)) #al inicio

df['total']   = ''  #al final
```

Operaciones con los valores:

```
df['total']   =  df[‘valor_bruto']* 0.19  #al final
```

Filtrado:

```
df [df[total']> 20000]
condition = df ['col1']> 20    #asignando en una variable el resultado
```

# Pandas

## Leyendo Dataset’s

```
df.loc [:, df.all()]    #mostrar columnas vacias = null
```

## Aplicando funciones a columnas:

```
def some_func (x):
return x * 2
df.apply (some_func) - # actualiza cada entrada de un DataFrame sin ningún bucle
# lambda también funciona
df.apply (lambda n: n * 2) - # the same

df ['new_col'] = df .apply (lambda n: n * 2)
```

# UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

## ¡Siempre hacia lo alto!

USTATUNJA.EDU.CO

@santotomastunja