**PREWORK**

**SESIÓN 01**

**Introducción:**

En esta sesión veremos las definiciones básicas de Machine Learning, algunas de las aplicaciones que se le han dado y cómo es que podemos utilizarlo para romper las barreras de la programación tradicional por métodos heurísticos. Para ello, necesitaremos que completes algunos pre-requisitos y que tengas algunas de las nociones básicas y conocimientos previos.

**Objetivos:**\* Instalar herramientas básicas  
\* Conocer las definiciones básicas de Machine Learning

**Pre-requisitos técnicos:**

Antes de comenzar la sesión, necesitaremos que **instales** algunas herramientas básicas:

1. Anaconda ó Virtualenv (O algún otro ambiente virtual de Python 3)

Un ambiente virtual de Python sirve para que puedas instalar cualquier cantidad de módulos de Python sin mayores dificultades. Si utilizas M Python en tu trabajo o estudio, o bien, si vas a tomar este curso con la computadora de tu trabajo, te recomiendo ampliamente que tengas un ambiente virtual.

1. Pip3

Pip significa “Package Installer for Python”, o instalador de paquetes para Python. Pip3 se enfoca entonces en instalar paquetes para Python 3 (y créeme, hay una gran diferencia entre usar Python 2 y Python 3). Por lo que vamos a utilizarlo para instalar cualquier otra cosa que lleguemos a necesitar.

1. MatPlotLib

Las herramientas de Matplotlib te ayudan a mostrar gráficas, las cuales nos serán muy útiles durante todo el curso, sobre todo para visualizar puntos en planos (scatterplots). Por lo que te recomiendo que lo descargues. No es un paquete 100% necesario (excepto para la sesión 08), pero si te recomiendo que lo tengas contigo.

Lo instalas con ***pip3 install matplotlib***

1. Jupyter Notebook

Este sistema va a ser el que usaremos durante todo el curso. No solamente es una herramienta para tomar notas, sino que también te servirá para ejecutar código de Python. Es sumamente útil para familiarizarte con el lenguaje de programación y si algo sale mal, no tienes que revisar toooodo tu código para corregir el problema, con solamente correr un fragmento habrás corregido el percance.

Lo instalas con ***pip3 install jupyter***

1. NumPy

NumPy es la herramienta de álgebra lineal optimizada para Python. Sus códigos están previamente optimizados y te aplanarán el camino de formas espectaculares. Es necesario instalarlo, aunque por lo general casi siempre viene instalado por defecto.

Lo instalas con ***pip3 install numpy***

1. Pandas

Pandas es una herramienta de análisis de datos fundamental para Python3. Si bien es cierto no veremos data analysis a fondo (esto es tema de otro módulo de BEDU), sí nos servirá para cargar datos de varias fuentes.

Lo instalas con ***pip3 install pandas***

**Pre-requisitos de conocimientos:**

**Listas y Arrays:** Una variable en Python puede almacenar un dato, pero también puede almacenar una colección de datos ordenados por medio de un índice. Esto se le conoce como Array o Lista. Para utilizar una lista hay dos maneras:

1. Una lista de objetos de Python:

*listaPython = [objeto1, objeto2, objeto3]*

1. o bien un arreglo de NumPy:

*arrayNumpy = np.array([1,2,3], dtype=np.int)*

En ambos casos, es importante que revises cómo se utilizan, de manera que te familiarices con ese concepto, sobre todo si nunca has trabajado con Arrays.

**Funciones:** Una función es un fragmento de código reutilizable que te permite separar un problema en múltiples sub-problemas, más sencillos de tratar por separado que en conjunto. Para ello, lo recomendable es que tomes en cuenta lo siguiente:

* Todas las funciones de python utilizan la sintaxis “def nombreDeLafuncion(parametros)” por ejemplo:

*def sumarDosNumeros(numeroA,numeroB):*

*resultado = numeroA+numeroB*

*return resultado*

* En las funciones la identación importa: todo lo que está separado por un “tab” a la derecha, pertenece a la función.
* En python puedes regresar varios valores de una función (a diferencia de C o java, que solo puedes regresar un valor).

Ya que tienes en mente los conocimientos previos necesarios, comencemos a definir el concepto de **Machine Learning.**

**Machine Learning** es un sub-ámbito de la Inteligencia Artificial en el que se desarrollan sistemas o algoritmos que aprenden de manera automatizada y mejoran su aprendizaje y desempeño de la misma manera; sin necesidad de ser reprogramados o intervenidos por humanos explícitamente. Este subámbito se enfoca en que los programas computacionales tengan acceso a data y que con esta data entonces este pueda tomar decisiones o predecir información.

¿Pero cómo funciona este proceso de aprendizaje? Primero se necesita dar datos, instrucciones o ejemplos para que el programa comience a identificar patrones, información clave y de esta manera pueda tomar decisiones basadas en información factual. Como se mencionó anteriormente, el enfoque principal de Machine Learning es el aprendizaje automatizado e independiente (sin intervención humana).  
  
Hay diferentes tipos de **algoritmos de aprendizaje** en Machine Learning. Estas son las principales categorías:

|  |  |  |  |
| --- | --- | --- | --- |
| **Supervised machine learning algorithms** (Algoritmos supervisados de ML) | **Unsupervised machine learning algorithms** (Algoritmos no supervisados de ML) | **Semi-supervised machine learning algorithms** (Algoritmos semi-supervisados de ML) | **Reinforcement machine learning algorithms** (Algoritmos de aprendizaje por refuerzo) |
| Estos algoritmos primero analizan datasets *históricos* con datos de entrada y salida previamente etiquetados. Esto quiere decir que tenemos *input* data que le corresponde a un *output* data.  Con esta *input data* entonces desarrollan una función para asignar etiquetas a los datos de salida (*output data*). Este tipo de algoritmos se tienen que entrenar con datos históricos para entonces predecir los datos de salida. | En este tipo de algoritmos no existen históricos de data y esta data no se encuentra previamente clasificada/etiquetada. En este tipo de algoritmos no se busca inferir datos de salida (output data) sino que busca explorar los datos y sacar inferencias, patrones para describir estructuras invisibles.  Este aprendizaje busca únicamente describir la estructura de los datos para entonces encontrar esta estructura que ayuda al análisis exploratorio. | En este caso los algoritmos semi-supervisados contienen data de entrenamiento que una parte está etiquetada (normalmente una pequeña parte), mientras que la otra no lo está (la mayor parte). Este tipo de aprendizaje se usa para cuando la data etiquetada necesita recursos relevantes y hábiles para aprender de esta. | Este tipo de algoritmo de aprendizaje se basa en procesos de retroalimentación al interactuar con su ambiente y entonces descubrir o accionar a errores o rewards.  Este modelo permite que el algoritmo determine el comportamiento ideal para mejorar su desempeño basándose en la información de entrada (feedback) para entonces aprender a base de ensayo-error. No entra dentro de la categoría de supervisado porque no necesariamente se tienen datos etiquetados ni es no supervisado ya que sí sabemos de antemano la recompensa esperada. Simplemente llevamos un control y monitoreo de las respuestas a las acciones. |
| ¿Para qué sirve este tipo de algoritmo? 1. Clasificación  2. Regresión   **Ejemplo** de este tipo de algoritmos:  - Identificar emails como “spam” o “no spam” basando info en conjunto de datos de emails previamente etiquetados como tal.  - Detectar fraude de identidad (clasificación) - Predecir el clima (regresión) | ¿Para qué sirve este tipo de algoritmo?  1. Clustering  2. Reducción de dimensionalidad  **Ejemplo** de este tipo de algoritmos:   * Crear perfiles de usuarios o clientes basados (clustering) * Crear sistemas de recomendación. Ejemplo: Netflix recomendando tipos de películas para tu perfil. | ¿Para qué sirve este tipo de algoritmo?  1. Entrenar modelos con poca data de entrenamiento. | ¿Para qué sirve este tipo de algoritmo?  1. Q learning  **Ejemplo** de este tipo de algoritmos:   * Algoritmos que aprenden a jugar videojuegos o juegos de mesa |

Estos tipos de algoritmos permiten el análisis y procesamiento de una gran cantidad de data e información para dar resultados de una manera eficiente. El objetivo de esto es identificar oportunidades y riesgos, así como tomar decisiones al predecir información.

**Revisa los recursos adicionales:**

En este [recurso](https://cleverdata.io/que-es-machine-learning-big-data/) puedes conocer sobre un ejemplo aplicado de Machine Learning para que identifiques exactamente cómo se aplica y su importancia en Big Data en una empresa así como algunas aplicaciones clave del aprendizaje de máquina actualmente.

Esta es una [guía](https://docs.python-guide.org/) que te llevará de lo más sencillo a lo más difícil de la programación con Python. Te servirá como apoyo para recordar conceptos básicos de este lenguaje de programación así como: temas de instalación, bases del lenguaje y estilo del código, ambientes de desarrollo, documentación útil, aplicaciones dentro de Python, y publicación y distribución de código.

Por otro lado, si aún no te quedó clara la definición de Machine Learning y sus características, revisa el siguiente [recurso adicional](https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471) que te ayudará a tener las bases claras antes de participar en la sesión presencial.

**Quiz:**

**¿Para qué sirve Jupyter Notebook?**

1. Para instalar paquetes de Python 3
2. Para utilizar álgebra lineal con Python
3. **Para programar en Python y tomar notas a la vez.**
4. Para usar un ambiente virtual.

**¿Cuál de todos estos NO es un array?**

1. Una lista de datos.
2. Una variable con datos ordenados.
3. Una lista de números.
4. **Un texto legible**

**¿Con qué línea de comandos instalamos librerías de análisis de datos?**

1. pip3 install matplotlib
2. **pip3 install pandas**
3. pip3 install numpy
4. pip3 install opencv

**¿Para qué usamos una función?**

1. **Para dividir un problema en problemas más pequeños.**
2. Para comentar un código y tener menos problemas.
3. Para ordenar la cantidad de problemas que tenemos.
4. Para mantener la funcionalidad de otros programadores.