Curso de Machine Learning Aplicado con Python

Introducción al curso de Machine Learning Aplicado con Python

Bienvenidos al **Curso de Machine Learning Aplicado con Python**, en este curso veremos cómo utilizar **Machine Learning** con distintas librerías de Python. Particularmente estaremos utilizando **Scikit-Learn**, que es una de las más utilizadas de la industria.
En este curso también nos enfocaremos es en entender todo el flujo de trabajo que se hace cuando se resuelve un problema de **Machine Learning**.
Además de entender muy bien los algoritmos de **Machine Learning** que estaremos viendo, veremos otras disciplinas que son tanto o más importantes como el **Feature Engineering** y la selección de modelos.

Importancia de definir el problema en Machine Learning

Errores comunes que se ven cuando no se define bien el problema y se comienza a codear:

No hay problemas por resolver.
Existen soluciones más simples.
No se puede medir el impacto del modelo.
No se sabe si el problema ya ha sido resuelto antes.
El problema es imposible de resolver.

Preguntas clave para reconocer el **tipo de aprendizaje** que se necesita:

¿Qué beneficio se puede generar y para quién?
¿Cuál de las siguientes funcionalidades sería más útil para lograr el objetivo?

a) Predecir alguna métrica (Aprendizaje supervisado)
b) Predecir una etiqueta (Aprendizaje supervisado)
c) Agrupar elementos similares.
d) Optimizar un proceso con prueba y error.

Preguntas clave para aterrizar el **problema de aprendizaje supervisado:**

¿De qué tipo es el valor que se quiere predecir?

a) Continuo
b) Discreto

¿Cuál es la definición de éxito en una predicción?
¿Con qué datos se contaría para hacer esa predicción?
¿La pregunta que se está tratando de resolver pertenece a alguna disciplina en particular?
Considerando nuestra intuición en la disciplina ¿Los datos nos permiten predecir el objetivo?

Terminología de Machine Learning

Terminología

Datos tabulares = Datos en dos dimensiones.
Líneas = Ejemplos
Columna = Feauture. Éstas son importantes porque nos van a ayudar a predecir cosas gracias a los modelos que usemos de Machine Learning.
Cantidad de columnas = Dimensión de los datos
Output de un algoritmo de Machine Learning (ML) = Modelo
Variable objetivo = Target

Materiales del curso: Notebooks de Jupyter

En este material te comparto los notebooks de Jupyter utilizados por el profesor durante el curso, recuerda que lo ideal es que tú mismo desarrolles estos ejercicios pero este material te servirá como apoyo y punto de comparación en caso de que te encuentres con alguna dificultad en el camino.
https://github.com/JuanPabloMF/machine-learning-platzi/tree/master
Espero que sigas disfrutando este curso y no olvides al final tomar el examen y obtener tu certificación.
¡Mucho éxito!

El ciclo de Machine Learning

Muchas veces pensamos que hacer **Machine Learning** corresponde solamente a implementar un algoritmo de cualquiera de las librerías y con ello ya existe la solución a un problema. Pero en realidad existe todo un ciclo de trabajo donde los algoritmos de **Machine Learning** son solo una etapa, sin embargo, las demás etapas también son muy importantes y toman su tiempo para lograr los resultados que esperamos.

Hacer **Machine Learning** corresponde a trabajar en un ciclo, ir trabajando varias etapas e ir iterando.

Ciclo de Machine Learning:

Definición del problema.
Preparación de los datos.
Representación de los datos.
Modelamiento / Algoritmos de ML.
Evaluación.

Este no es el final del proceso, se debe iterar hasta que en alguna de las iteraciones salga la solución al problema.

Producción (Fin del proceso).

Construcción de contenedores de Docker

Datos importantes:

Los contenedores de Docker son similares a una ballena virtual y se pueden crear de forma automática con Docker File, que va a crear una imagen y desde esa imagen se generan todos los contenedores que necesitemos.

ararads - base es un ambiente típico, muy fácil de usar de librerías de PyData.

Antes de instanciar un contenedor hay que asegurarnos de crear un directorio que se llame vol, esto dentro de la carpeta. Esto es importante porque cada vez que se requiera disponibilizar archivos en el contenedor podemos solo hacer un drag and drop y llevarlos a vol.

Qué es y cómo se utiliza Numpy

Datos importantes:

Numpy es una librería muy importante para el ecosistema de **Python** ya que es la base de todos los cálculos científicos y muchas de las librerías de **Machine Learning.**

Scikit-Learn con sus modelos, cuando retorna un resultado, en general lo retorna en un formato **Numpy.**
La API de **Numpy** tiene muchas similitudes con **Pandas.**
Numpy reemplaza de forma más eficiente lo que podría ser un tipo lista. En las listas podemos tener conjuntos de elementos numéricos. Sin embargo las listas no logran manejar datos de dos dimensiones.
Las listas no poseen métodos que son prácticos para hacer aritmética.
Es importante saber que otros lenguajes de programación poseen librerías altamente optimizadas para hacer cálculos numéricos con vectores de datos. **Numpy** es esa librería para el lenguaje de programación de **Python.**
np.linspace es una función que permite crear un array de una dimensión de números entre 0 y 1.
Los **array** a diferencia de otros objetos en **Python** están fuertemente tipificados. Esta tipificación fuerte es necesaria porque es una de las cosas que permite que esta librería sea más rápida que ocupar listas, por ejemplo.

Cargar los datos necesarios para el proyecto

Las librerías con las que vamos a trabajar y que tenemos que cargar son:
a) Numpy
b) Pandas
c) Matplotlib.pyplot
d) Seaborn

Inspección de los tipos de datos

Datos importantes:

La **inspección de los datos** se da para tener conocimiento de la salud de los datos que tenemos, saber si vienen limpios o no, y también porque se quiere tener un entendimiento cuantitativo de ellos. Parte de esto es mirar gráficos estadísticos y entender diferentes propiedades numéricas de las columnas.
A diferencia de **Numpy**, **Pandas** no solo permite cargar datos numéricos, sino también datos de texto.
El método info nos va a mostrar la cantidad completa de columnas con la cantidad de elementos no nulos que hay en esas columnas, y por último muestra el tipo de cada columna.

El objeto estimador de Scikit-Learn

Datos importantes:

Por como fue diseñado **Scikit-Learn,** existe una API muy fácil de ocupar y muy unificada. Esta unificación se da por un objeto que se llama “estimador” que tiene en todos los casos y para el algoritmo de Machine Learning que sea, una API que es común y 3 métodos que son clave.
Scikit-Learn posee muchos modelos, se pueden implementar tanto, regresiones lineales como regresiones regularizadas, árboles de decisión, SDMs, etc.
Scikit-Learn contiene todos los modelos que son usados hoy en día, y una de las virtudes de esta librería es que sigue muy de cerca lo que pasa en la investigación.
Scikit-Learn es la librería más usada de **Machine Learning General**, no de **Machine Learning Especializado**, para ello está la librería de **Tensor Flow** y sirve casi exclusivamente para modelos de **Deep Learning.**