Aprendizaje Automático y Programación para Análisis de Datos

Gerardo Rodríguez

1. **Definiciones básicas:**

   a) **Define el concepto de aprendizaje automático y su relación con la ciencia de datos.**

   Aprendizaje automático es el resultado que obtiene un robot (Software) de analizar una serie de datos y en base a ello identificar un patrón, un algoritmo, una solución a un problema o una decisión para obtener un resultado en particular solo por mencionar algunas. El aprendizaje automático es una de las herramientas utilizadas dentro de la ciencia de datos, que si bien es una de las más utilizadas se complementa con otras como la estadística, algebra lineal, el reconocimiento de patrones, gestión de bases de datos, minería de datos, la creación automática de conocimiento, por mencionar algunas.

   b) **¿Cuál es la diferencia entre aprendizaje supervisado y no supervisado?**

   La principal diferencia radica en que en el aprendizaje supervisado etiquetamos los datos, en otras palabras, le damos a la máquina un patrón de como ciertos insumos nos dan un resultado, con el fin de que la computadora comprenda el mismo y lo reutilice para hacer inferencias.

   Mientras que en el aprendizaje no supervisado, le damos los datos a modelo y este dependiendo de sus reglas podría identificar patrones, agrupar la data, entre otros.

   c) **Explica la importancia del aprendizaje por refuerzo y proporciona un ejemplo de su aplicación.**

   En este escenario, damos premios o castigos al modelo para que reoriente su comportamiento, y ofrezca nuevos resultados basados en los nuevos elementos incorporados. Tiene aplicaciones en los algoritmos de recomendación, en los juegos como el ajedrez, generando de este modo robots “expertos”.

2. **Herramientas y bibliotecas**:

   a) **Describe las funcionalidades principales de la biblioteca Scikit-Learn en el contexto del aprendizaje automático.**
   Es una librería de Python, que se puede cargar para realizar varias funciones de aprendizaje automático. Entre estas: aprendizaje supervisado, aprendizaje no supervisado, división de datos de prueba y datos de aprendizaje, normalización de los datos mediante estadística, autovalidación de los modelos generados para definir el porcentaje de eficacia de los mismos.
   Puede utilizar diferentes métodos como regresión lineal, k vecinos, árboles de decisión, máquinas de vectores de soporte y redes neurales.

   b) **Menciona una diferencia principal entre TensorFlow y PyTorch y explica cuándo podrías preferir usar uno sobre el otro.**
   Nunca los he usado, pero por lo que investigue, Tensorflow es una biblioteca más vieja y

por lo mismo con una comunidad mucho más extensa, con amplias posibilidades de conectarse con software tercero y bajo la sombrilla de Google. Lo que lo hace muy versátil, robusto y de buen performance. Tensorflow cuenta con apoyos visuales lo que facilita su interpretación y depuración.

Sin embargo, Pytorch desarrollado por Facebook, es una librería más nueva y con un diseño versátil que le permite generar desarrollos y modelos de manera acelerada para generar prototipos. Pytorch no utiliza apoyos visuales, y esto a pesar de parecer una desventaja lo hace más veloz.

En lo personal, me gustaría usar los dos, pero me parece que si tengo un proyecto grande al día de hoy TensorFlow sería una mejor herramienta, pero si requiero hacer experimentos o proyectos pequeños Pytorch es una mejor opción.

c) **Keras se integra con otra biblioteca mencionada anteriormente. ¿Cuál es esa biblioteca y cuál es el propósito principal de Keras?**
Keras es un add-on que puede agregarse al modelo de TensorFlow o al de Theano. Su principal objetivo es apoyar en el desarrollo de modelo de Redes neuronales,

3. **Funciones y análisis de datos**:

a) **Explica el papel de las funciones en el análisis de datos y por qué son esenciales para un científico de datos.**
Las funciones nos permiten llamar varias veces a un proceso que se requiere de manera recurrente, incluso podemos programar que se detone de manera automática en ciertas situaciones, nos permite programar pedacitos de código por bloques.

En otras palabras, nos permite diseñar nuestros modelos de manera más dinámica y presentar de mejor manera nuestros resultados.

b) Imagina que estás trabajando con un conjunto de datos en Python y quieres estandarizar varias variables. Escribe una función para estandarizar tus datos usando la siguiente Fórmula para estandarizar variables:

$$z = \frac{x - \mu}{\sigma}$$

Donde:

z es la variable estandarizada

x es el valor original de la variable

μ es la media de la variable

σ es la desviación estándar de la variable

Nota: Al aplicar esta fórmula, se obtiene una nueva serie de valores que siguen una distribución con media 0 y desviación estándar 1.

Código en Python

```
import numpy as np #importamos la libreria Numpy como comando np
x = (1, 2, 3,4, 5, 6,7, 8, 9) #definimos valores de x
def estandarizar(x): #creamos la función
    media = np.mean(x) #sacamos la media con la función mean de numpy
    desviacionEst = np.std(x) #sacamos la desviación estándar con la función std de numpy
    z = (x - media) / desviacionEst #obtenemos los valores de z
    return z

estandarizar(x) #ejecutamos la función para estandarizar los números
```