

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ZUMPANGO

INGENIERÍA EN COMPUTACIÓN

RECONOCIMIENTO DE PATRONES

LABORATORIO CLASIFICACIÓN CON K-NEAREST NEIGHBORS

Dr. Asdrúbal López Chau

El clasificador K-Nearest Neighbors es uno de los métodos más simples de entender. En este laboratorio, implementarás completamente ese método de clasificación.

1. Usando como esqueleto el código orientado a objetos proporcionado, implementa lo siguiente:
 - a. Método `fit(self, X, y, metric=None)`: Debido a que K-NN es un método perezoso, solamente se deben de almacenar los valores `X`, `y` en variables de instancia. El argumento `metric` se refiere a la distancia utilizada.
 - b. Método `predict(self, X)`: Este método recibe como argumento `X`, que puede representar a una sola instancia o a un grupo de objetos. El tipo de dato de `X` puede ser una lista (lista de listas en realidad), un DataFrame de pandas o un array de numpy. En cualquiera de los casos anteriores, el método `predict` regresa un objeto de tipo Serie de pandas, que contiene las predicciones de cada objeto que se pasó al método.

A manera de ejemplo, se muestran algunos posibles casos para `X`

	x0	x1	x2	x3
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

`X` es un DataFrame de pandas con cinco instancias, el método `predict` debe regresar cinco etiquetas como una Serie de pandas

```
In [164]: X
Out[164]: [[4.1, 0.5, 2.4, 0.2]]
```

`X` es una lista con una instancia, el método `predict` debe de regresar una etiqueta como una Serie de pandas

```
In [168]: X
Out[168]: [[4.1, 0.5, 2.4, 0.2], [2.1, 5.7, 3.3, 5.1]]
```

`X` es una lista con dos instancias, el método `predict` debe de regresar dos etiquetas como una Serie de pandas

```
Out[170]:  
array([[4.1, 0.5, 2.4, 0.2],  
       [2.1, 5.7, 3.3, 5.1]])
```

X es un `array de numpy` con dos instancias, el método `predict` debe de regresar dos etiquetas como una `Serie de pandas`

Consideraciones:

- Implementar en una o varias clases las siguientes métricas de distancia:
 - Euclidean (Keren)
 - Manhattan (Arturo)
 - Minkowski (Keren)
 - Distancia Tanimoto (Arturo)
 - En el método `fit` de la clase que implementa el método de clasificación K-NN, se podrá indicar la medida de distancia a emplear.
 - El método `predict`, deberá de comprobar que todos los atributos sean de tipo numérico real, si no es así, entonces el método deberá de lanzar una excepción personalizada.
 - Implementar en una o varias clases, los métodos de conversión de atributos de los tipos no numérico a numérico vistos en clase.
 - Implementar en una clase o varias clases la funcionalidad de convertir los atributos de un conjunto de datos en tipo numérico. El uso de esto debe de ser intuitivo y simple de usar, pudiendo incluso indicar diferentes conversiones para mismos tipos de datos no numéricos. Por ejemplo, para un tipo de variable binaria, se debe de poder indicar si la conversión es simétrica o no simétrica.
- c. Siguiendo las buenas prácticas de programación, crea métodos auxiliares para realizar operaciones necesarias para el método `predict`, es decir, no se recomienda hacer todo el proceso en un solo método.
- d. Prueba el desempeño del método

Rúbrica

1. Código con encabezados y documentación (comentarios) importantes para entender el código 10%
2. Funcionamiento correcto e implementación completa: 80%
3. Buenas prácticas de programación, orientación a objetos, seguir el estándar de Python para escritura de código (ver por ejemplo la referencia abajo): 10%

<https://ihumai.medium.com/pep8-un-est%C3%A1ndar-para-escribir-c%C3%B3digo-en-python-96b7d44d4db3>