

Guía de Ejercicios

Ejercicios de aplicación de NumPy. Varios de ellos se utilizarán más adelante en la materia como parte de problemas más grandes.

Ejercicio #1: Operaciones Matriciales

Dada una matriz en formato numpy array, donde cada fila de la matriz representa un vector matemático:

- Computar las normas l0, l1, l2, l-infinito
 - l0: número de elementos diferentes a cero en el vector
 - l1-l2:

$$\|x\|_p = \left(\sum_1^n |x_i|^p \right)^{\frac{1}{p}}$$

- l-infinito:

$$\|x\|_{\infty} = \max_i |x_i|$$

Ejercicio #2: Sorting

Dada una matriz en formato numpy array, donde cada fila de la matriz representa un vector matemático, se requiere computar la norma l2 de cada vector.

Una vez obtenida la norma, se debe ordenar las mismas de mayor a menor. Finalmente, obtener la matriz original ordenada por fila según la norma l2.

Todas las operaciones debe ser vectorizadas.

Ejercicio #3: Indexing

El objetivo es construir un índice para identificadores de usuarios, es decir *id2idx* e *idx2id*.

Para ello crear una clase, donde el índice se genere en el constructor. Armar métodos *get_users_id* y *get_users_idx*.

- Identificadores de usuarios : `users_id = [15, 12, 14, 10, 1, 2, 1]`
- Índice de usuarios : `users_idx = [0, 1, 2, 3, 4, 5, 4]`

```
1 id2idx = [-1    4    5   -1   -1   -1
           -1   -1   -1   -1    3   -1
2     1   -1    2    0]
   [ 0    1    2    3    4    5
   6    7    8    9   10   11
3  12   13   14   15]
4 id2idx[15] -> 0 ; id2idx[12] -> 1 ;
  id2idx[3]  -> -1
5 idx2id[0]  -> 15 ; idx2id[4]  -> 1
```

Ejercicio #4: Precision, Recall, Accuracy

En los problemas de clasificación, se cuenta con dos arreglos, la **verdad** (ground truth) y la **predicción** (prediction).

Cada elemento de los arreglos puede tomar dos

valores: *True* (representado por 1) y *False* (representado por 0).

Por lo tanto, se pueden definir cuatro variables:

- True Positive (TP): la verdad es 1 y la predicción es 1.
- True Negative (TN): la verdad es 0 y la predicción es 0.
- False Negative (FN): la verdad es 1 y la predicción es 0.
- False Positive (FP): la verdad es 0 y la predicción es 1.

A partir de esas cuatro variables, se definen las siguientes métricas:

- Precision = $TP / (TP + FP)$
- Recall = $TP / (TP + FN)$
- Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Para los siguientes arreglos, representando la **verdad** y la **predicción**,

calcular las métricas anteriores con operaciones vectorizadas en NumPy.

- truth = [1,1,0,1,1,1,0,0,0,1]
- prediction = [1,1,1,1,0,0,1,1,0,0]

Ejercicio #5: Average Query Precision 🏠

En information retrieval o search engines, en general contamos con queries “q” y para cada “q” una lista de documentos que son verdaderamente relevantes.

Para evaluar un search engine, es común utilizar la métrica **average query precision**.

Tomando de referencia el siguiente ejemplo, calcular la métrica con NumPy utilizando operaciones vectorizadas.

```
1 q_id = [1, 1, 1, 1, 2, 2, 2, 3,
          3, 3, 3, 3, 4, 4, 4, 4]
2 predicted_rank = [0, 1, 2, 3, 0, 1, 2, 0,
                   1, 2, 3, 4, 0, 1, 2, 3]
3 truth_relevance = [T, F, T, F, T, T, T, F,
                    F, F, F, F, T, F, F, T]
```

- Precision para q_id 1 = 2 / 4
- Precision para q_id 2 = 3 / 3
- Precision para q_id 3 = 0 / 5
- Precision para q_id 4 = 2 / 4

average query precision = $((2/4) + (3/3) + (0/5) + (2/4)) / 4$