

Análisis del Algoritmo *knn* (*k*-nearest neighbors)

Objetivos:

1. Implementar el algoritmo *knn*.
2. Evaluar la precisión del algoritmo para distintos valores de *k*.
3. Elaborar un informe completo en base a las pruebas realizadas.

Descripción del problema:

Este trabajo consiste en implementar el algoritmo *knn* que, dado un dataset, permita parametrizar la cantidad de vecinos próximos con los que se evaluarán los datos de prueba. La idea básica es definir el modelo que tome los datos, de diferentes tipos, de los atributos del dataset indicado por la cátedra y devuelva los resultados de clasificación para valores de *k* desde 1 hasta 10. Adicionalmente, se deberá implementar una estrategia para la estimación del valor óptimo de *k*. La implementación deberá tener una interfaz amigable que permita al usuario correr pruebas fácilmente y observar los resultados obtenidos.

k-nearest neighbor Learning

k-nearest neighbor (*knn*) is a lazy learning method in the sense that no model is learned from the training data. Learning only occurs when a test example needs to be classified. The idea of *knn* is extremely simple and yet quite effective in many applications, e.g., text classification. It works as follows: Again let *D* be the training data set. Nothing will be done on the training examples. When a test instance *d* is presented, the algorithm compares *d* with every training example in *D* to compute the similarity or distance between them. The *k* most similar (closest) examples in *D* are then selected. This set of examples is called the *k* nearest neighbors of *d*. *d* then takes the most frequent class among the *k* nearest neighbors. Note that *k* = 1 is usually not sufficient for determining the class of *d* due to noise and outliers in the data. A set of nearest neighbors is needed to accurately decide the class.

Liu, Bing. Web Data Mining (Data-Centric Systems and Applications) (p. 67). Springer. Edición de Kindle.

Se deberá entregar:

- Código fuente (que incluya documentación interna) que corra en la plataforma colaborativa Google Colab.
- Informe completo (digital) del desarrollo del trabajo que contenga:
 - Introducción.
 - Descripción del/los algoritmo/s de desarrollado/s.
 - Descripción de la solución implementada (explicando los problemas encontrados).
 - Simulaciones realizadas y resultados obtenidos.
 - Conclusiones.
 - Referencias.

Consideraciones adicionales:

- Se deberá contar con una interfaz de usuario que permitan la total operabilidad de la aplicación.
- El informe del trabajo deberá seguir el formato de LNCS (Lecture Notes in Computer Science) de Springer Verlag. La URL de donde se pueden bajar las plantillas del informe es: <http://preview.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines> en esa página se encuentra, además de las plantillas, una guía para dar formato al informe.
- El informe no deberá exceder las 10 carillas (no incluir la impresión del código fuente).
- Para la nota final del Informe se tendrán en cuenta aspectos como: prolijidad, redacción y ortografía.
- La información de cada fuente utilizada deberá ser citada. En caso que la referencia no coincida con el desarrollo o haya algún tipo de plagio, el trabajo práctico será automáticamente **desaprobado sin derecho a recuperatorio**.

Condiciones generales:

Fecha límite de entrega	<input type="checkbox"/>	01/11/2023 a las 23:59 hs. (*)
Fecha de coloquio grupal	<input type="checkbox"/>	A convenir con los alumnos.
Calificación	<input type="checkbox"/>	Promedio ponderado de informe, programa y coloquio.

(*) Luego de esta fecha y hora se aceptarán los trabajos hasta el 08/11/2023, pero sin la posibilidad de recuperatorio.