Programación Concurrente y Distribuida

Tarea Académica 2 - 2021-1

Instrucciones:

- Desarrollar uno de los algoritmos según indicación de su profesor.
- El trabajo es individual.
- Se usará software para detección de plagio.

Descripción

El trabajo parcial consiste en desarrollar una aplicación que usando GO implemente un algoritmo de Machine Learning de la manera más eficiente possible, haciendo uso de mecanismos de paralelización y sincronización.

Rúbrica de calificación

- (2 puntos) Seleccionar un dataset del portal de datos abiertos del perú (https://www.datosabiertos.gob.pe/) de por lo menos 5 columnas,
- (2 puntos) Leer el dataset en la aplicación GO (El archivo dataset debe estar guardado en un repositorio de github y que pueda ser leido atraves del raw link)
- (8 puntos) Implementar el algoritmo indicado de manera eficiente (se calificará con 0 puntos si hace uso de librerías KNN de terceros).
- (4 puntos) Implementar una interfaz Web que permita configurar los parámetros y mostrar resultados.
- (3 puntos) Implementar una API REST en GO que se comunique con la interfaz Web para recibir los parámetros de configuración, ejecute el algoritmo implementado y devuelva los resultados obtenidos del algoritmo.
- (1 punto) Presentación de la documentación completa.

Documentación:

1.- Presentar un informe conteniendo:

a.- Carátula

b.- Explicación del data set seleccionado.

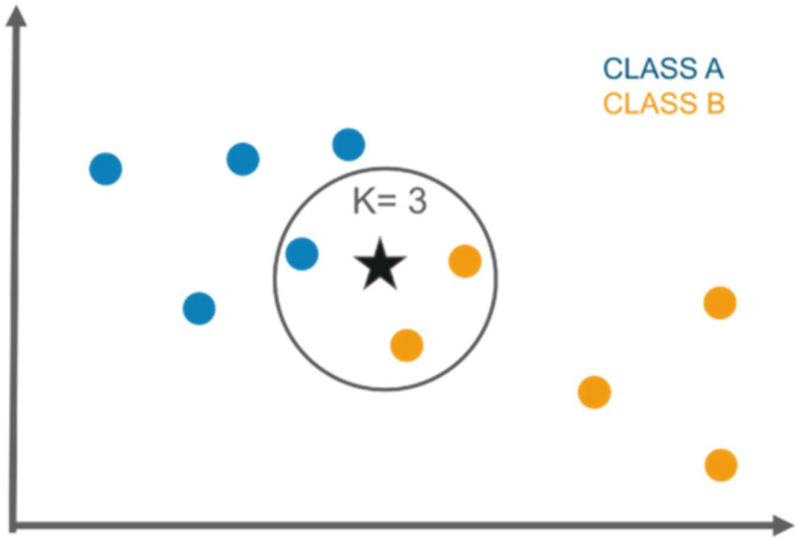
c.- Explicación del algoritmo y el uso de los mecanismos de paralelización y sincronización utilizado.

2.- Elaborar un video de máximo 5 min presentando el funcionamiento de la aplicación y publicarlo en un repositorio en la nube para acceder y visualizar a traves de la url.

Presentación

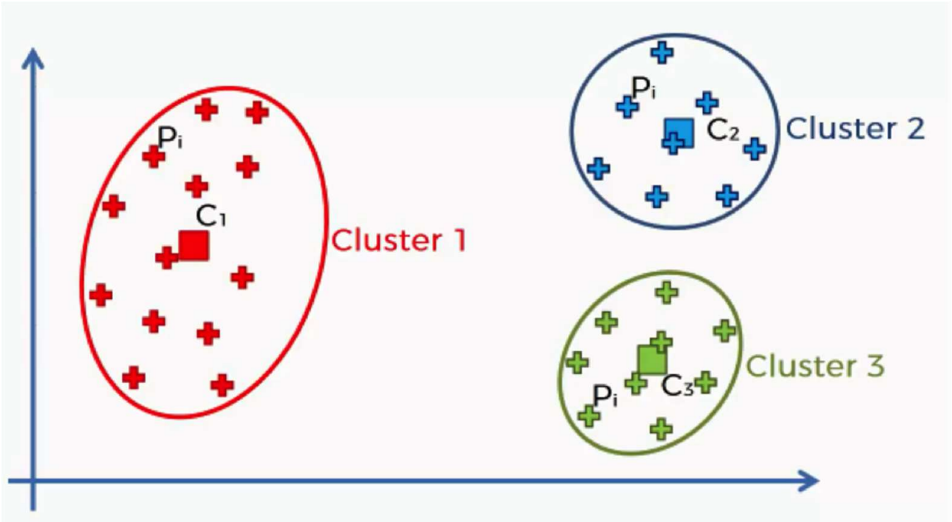
Empaquetar en un unico archivo zip su informe + Código Fuente + url del video y colocar como nombre su Código de alumno (ejemplo 201616054.zip) finalmente subir el archivo al aula virtual en la semana 13.

Algoritmo 1: K Nearest Neighbours



K Nearest Neighbours es un algoritmo no paramétrico usado para clasificación y regresión. En ambos casos el input consiste de un conjunto de ejemplos en un espacio de características N-dimensional (en la figura solo se muestran 2 dimensiones) y un valor K que determina el conjunto de selección. Para un nuevo dato de entrada cuya clase es desconocida, se buscarán los K puntos más cercanos, de acuerdo a una medida de distancia establecida previamente (por ejemplo, Manhattan o Euclidiana), y luego se decidirá la clase del nuevo dato en función a la clase de la mayoría en el grupo de los K escogidos.

Algoritmo 2: K Means



El algoritmo K-means consiste en crear K clusters de datos no etiquetados. Primero se deben escoger K puntos aleatoriamente en el espacio N-dimensional de los datos (en la figura se muestran solo 2 dimensiones) y luego se agrupan los datos según junto al punto más cercano. Luego se debe calcular los centroides de cada grupo, los cuales serán los nuevos K puntos y se repetirá el proceso de agrupar los puntos hasta que los centroides dejen de moverse o se muevan muy poco según un threshold mínimo. Las medidas de distancia deben ser establecidas previamente (por ejemplo Manhattan, euclidiana, etc.)

Recursos de ayuda:

A Simple Introduction to K-Nearest Neighbors Algorithm | by Dhilip Subramanian | Towards Data Science

How kNN algorithm works - YouTube

K-means: A Complete Introduction. K-means is an unsupervised clustering… | by Alan Jeffares | Towards Data Science

How K-Means algorithm works - YouTube

https://help.data.world/hc/en-us/articles/115006300048-GitHub-how-to-find-the-sharable-download-URL-for-files-on-GitHub