**Experimento Clustering**

1. **Realización del experimento**

Una vez finalizado el diseño de experimentos, se procede a ejecutar las pruebas de cada tratamiento. Se realizará un total de 20 repeticiones de cada tratamiento. No se consideró realizar más de 20 ejecuciones por tratamiento puesto que al equipo en cual se llevará a cabo el experimento se le ha fijado un núcleo que le da prioridad a esa tarea y los tiempos entre experimento y experimento no estaban oscilando mucho. Los resultados del experimento se anexan al documento de Word “Experimento K-means vs Bisecting-K-means y al Excel” y al Excel “Resultados Experimento Clustering”

1. **Análisis**

Para este contexto problemático no se consideró usar ANOVA para el análisis del experimento, debido a que una de las variantes del experimento es el tamaño de la entrada, el valor de tiempo en la salida ira aumentando conforme a la entrada se le dan valores más grandes por lo cual se espera que las medias de cada tratamiento difieran en tamaño. Por la tanto, para realizar el análisis se va a usar la prueba de hipótesis para la diferencia de medias. Se quiere analizar a partir de que tamaños de entrada resulta ser más favorable usar un algoritmo que otro, esto tomando como referente el tiempo de respuesta de ejecución de los algoritmos.

Se realizarán 5 pruebas de hipótesis comparando los dos algoritmos en cada uno de sus tamaños de entrada. El resultado de la prueba de hipótesis se consigna en el documento de Excel.

1. **Interpretación**

Para las dos primeras longitudes de entrada (10 y 100) se encontró que el tiempo de ejecución del algoritmo Bisecting es menor que el del K-means, para entradas mayores a 1000 el algoritmo K-means tiene un tiempo de respuesta menor que el algoritmo Bisecting. Las conjeturas iniciales eran que el algoritmo Bisecting resultaba mejor para entradas pequeñas y la prueba de hipótesis ratifica este resultado.

1. **Conclusiones**

El algoritmo de clusterizacion Bisecting es un algoritmo que resulta ser optimo y veloz para entradas de 100 elementos o menos, para entradas más grandes el algoritmo que resulta ser más eficiente es el algoritmo K-means.