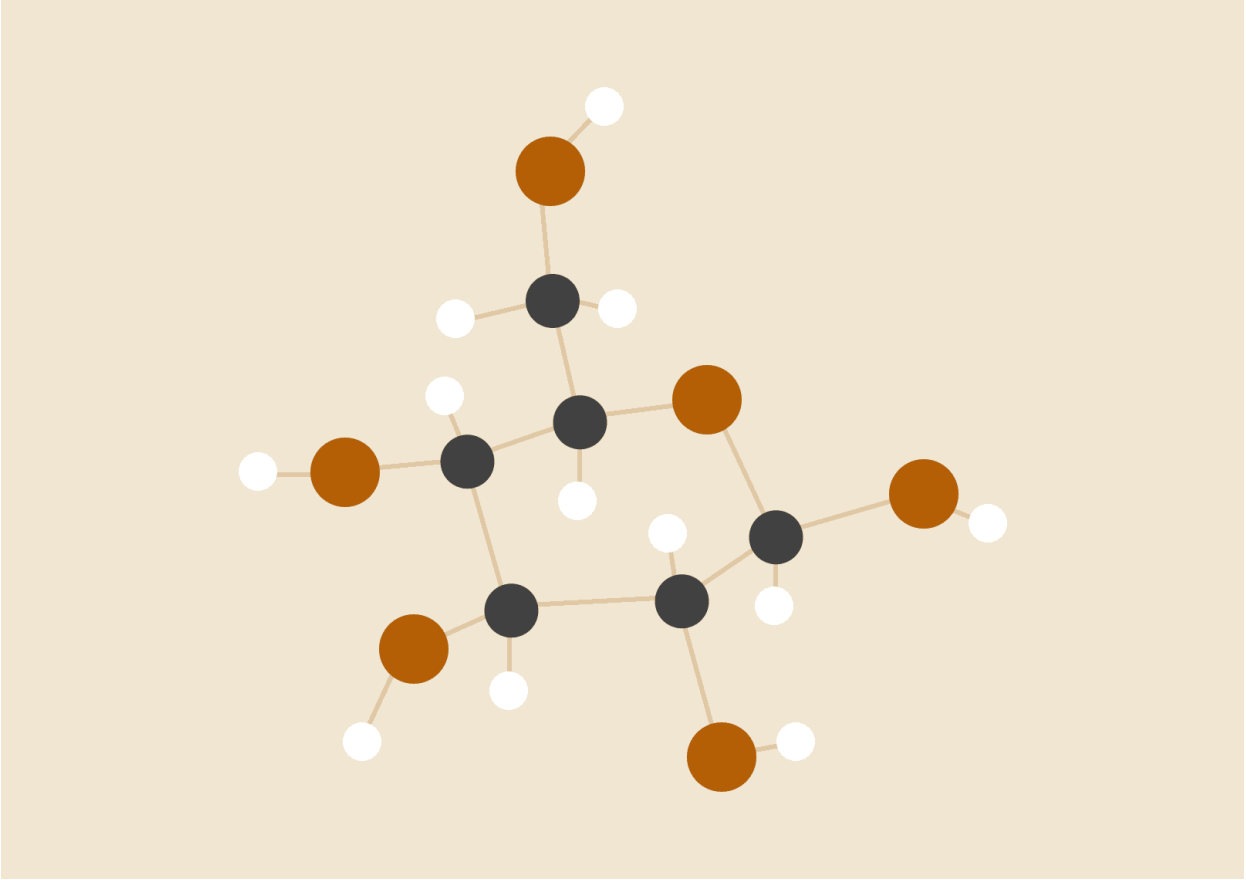
INFORME DE PROYECTO FINAL

SIA - K-Means



Oliveto, Jesús - 42260528

Entrega: 27-02-2026

Profesor: Prof. Esp. Ing. Agustín Fernandez

UBP

UNIVERSIDAD
Blas Pascal

Índice

INTRODUCCIÓN..... 2
¿QUÉ ES K-MEANS?..... 2
DESCRIPCIÓN DEL PROBLEMA..... 4
INTRODUCCIÓN..... 4
REQUERIMIENTOS..... 4
OBJETIVOS GENERALES..... 5
DATASET..... 6
DESCRIPCIÓN DE LOS ATRIBUTOS..... 6
DESARROLLO DEL PROYECTO..... 7
1.PANEL DE CONTROL..... 7
2.EXPLORADOR DE DATOS..... 9
2.DETERMINACIÓN DEL K ÓPTIMO..... 11
3.COMPARATIVA DE RENDIMIENTO Y CALIDAD..... 12
Implementaciones:..... 13
Versión No Vectorizada (KMeansLoop)..... 13
Versión Vectorizada (KMeansNumpy)..... 13
Versión de Control (KMeansSklearn)..... 13
4.PREDICCIÓN..... 14
5.ANÁLISIS DETALLADO..... 15
CONCLUSIONES..... 16
BIBLIOGRAFÍA..... 17

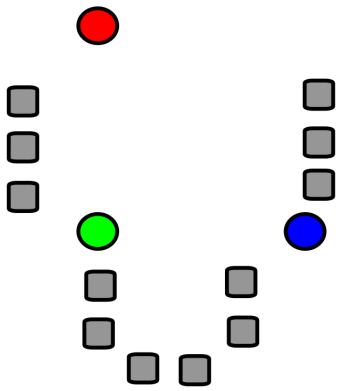
INTRODUCCIÓN

¿QUÉ ES K-MEANS?

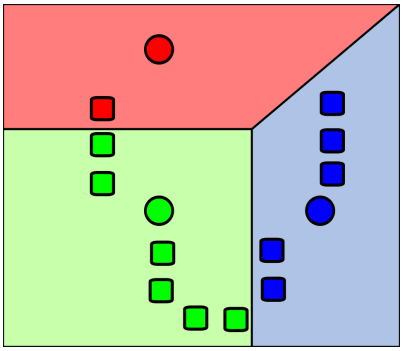
K-means (o K-medias) es un algoritmo de Machine Learning de aprendizaje no supervisado que tiene como objetivo particionar el dataset entre K clusters, distintos y no superpuestos, basándose entre las similitudes de sus cualidades y cada dato pertenece exactamente a un grupo. Cada cluster está representado por un centroide, que es la media aritmética de todos los datos asignados a ese grupo.

El algoritmo funciona mediante la repetición de dos pasos clave hasta que se da la convergencia de los centroides. Sigue el siguiente paso a paso:

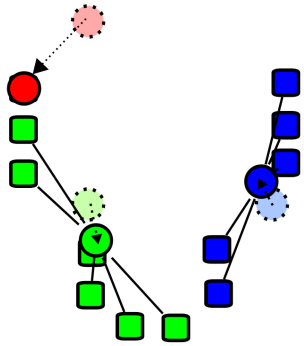
1) INICIALIZACIÓN: Se seleccionan k centroides iniciales, que pueden ser puntos de datos aleatorios del dataset o puntos generados de alguna manera específica entre los datos de entrada.



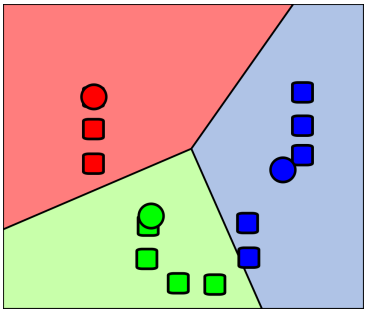
2) ALGORITMO DE LLOYD:
    a) ASIGNACIÓN DE CLUSTERS: asigna cada punto de datos al centroide más cercano, formando los k clústeres.



    b) ACTUALIZACIÓN: Se recalcula la posición de cada centroide tomando la media aritmética asignados a él en el paso de asignación.



3) CONVERGENCIA: Se repiten los pasos del algoritmo de LLoyd hasta que los centroides converjan, o sea, que ya no cambien significativamente de posición entre iteraciones, o hasta que alcance un número máximo de iteraciones.



DESCRIPCIÓN DEL PROBLEMA

INTRODUCCIÓN

El laboratorio de Inteligencia Artificial de la empresa Ultralistic nos ha solicitado realizar nuestra propia implementación del algoritmo K-Medias, ya que posee un dataset sobre la calidad vitivinícola y pretende analizar cuáles son las agrupaciones relevantes y "naturales" del mismo, más allá de las clases presentes en dicho dataset, y para ello desea utilizar el mencionado algoritmo.

REQUERIMIENTOS

Para la implementación se establecieron los siguientes requerimientos:

- Se puede escribir en el lenguaje de programación preferido por el grupo de trabajo.
- No se admiten implementaciones que posean la utilización de librerías o frameworks (o cualquier cosa parecida) que ya contengan el algoritmo solicitado.
- Debe realizarse una implementación del algoritmo sin "vectorizar" y otra "vectorizada".
- Está permitido utilizar librerías o frameworks (o cualquier cosa parecida) para soporte de los cálculos matemáticos, como pueden ser las operaciones de matrices, cálculos estadísticos, etc.
- Las implementaciones deben ser flexibles, es decir, que se debe poder determinar la cantidad de clústeres que se desean y además se debe poder manejar cualquier número de ejemplos y atributos del dataset correspondiente.
- Los tipos de datos que deben soportar las implementaciones propias de K-Medias, por supuesto que deben ser normalizados de alguna forma para evitar inconvenientes.
- Se debe realizar una comparativa entre las implementaciones propias y la implementación de alguna librería o framework disponible en el mercado (dicha librería o framework puede ser que se encuentre escrita en otro lenguaje de programación).

Teniendo en cuenta dichos requerimientos se debe lograr:

1. Construir las implementaciones de K-Medias solicitadas cumpliendo con lo establecido en los requerimientos.
2. Realizar una comparativa (puede ser gráfica) que contenga múltiples ejecuciones del dataset para diferentes números de clústeres utilizando tanto las propias implementaciones de K-Medias como las de terceros.
3. Se espera que las implementaciones propias de K-Medias posean una interfaz amigable y permitan “predecir” a qué grupo (o clúster) pertenecería un elemento (o registro) que se encuentre fuera del dataset .

OBJETIVOS GENERALES

1. Desarrollar implementación del algoritmo K-Medias: implementar el algoritmo K-Medias desde cero, cumpliendo con los requisitos establecidos, sin utilizar bibliotecas o frameworks que ya contengan el algoritmo solicitado.
2. Realizar implementaciones No Vectorizadas y Vectorizadas: crear dos versiones del algoritmo K-Medias para evaluar las diferencias en términos de eficiencia y rendimiento. 3. Soporte para diferentes números de clústeres y atributos: diseñar las implementaciones de tal manera que permitan configurar el número de clústeres y establecer los atributos a utilizar.
3. Comparación con implementaciones de terceros: realizar una comparativa entre la implementación propia y una implementación de terceros.
4. Interfaz gráfica: desarrollar una interfaz para poder visualizar los atributos del dataset, las estadísticas de cada atributo, controlar qué atributos se van a utilizar para la ejecución del algoritmo, establecer la cantidad de clústeres, visualizar resultados en forma de gráficos, etc.

DATASET

Para los propósitos del proyecto, ha sido provisto el dataset “Wine quality”, que contiene los registros de múltiples vinos, cada uno representado por sus características químicas, junto con una clasificación de calidad, asignada manualmente.

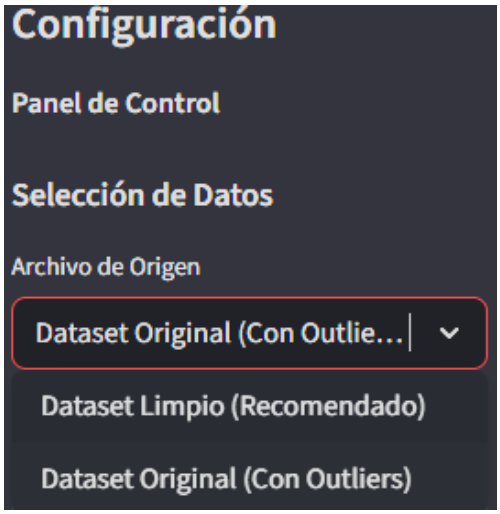
DESCRIPCIÓN DE LOS ATRIBUTOS

- **fixedacid (Ácido Fijo):** Representa los ácidos estructurales no volátiles.Es lo que le da al vino su acidez natural y estructura para que pueda envejecer bien en la botella.
- **volacid (Ácido Volátil):** Indica la cantidad de ácidos gaseosos. En exceso, hacen que el vino sepa y huela a vinagre.
- **citricacid (Ácido Cítrico):** Mide la pequeña cantidad de ácido cítrico presente que aporta frescura y potencia las notas frutales.
- **residualsugar (Azúcar Residual):** Es el nivel de azúcar natural de la uva que queda tras la fermentación, determinando si es un vino seco o dulce.
- **chlorides (Cloruros):** Refleja la concentración de sales disueltas provenientes del suelo, en exceso aporta un gusto salado indeseable .
- **freesulfur (Dióxido de Azufre Libre):** Mide el dióxido de azufre activo que actúa como conservante principal, protegiendo al vino de la oxidación y microbios.
- **totalsulfur (Dióxido de Azufre Total):** Es la suma de todo el dióxido de azufre (libre y combinado) que, si es demasiado, tapa los aromas ricos con un olor fuerte.
- **density (Densidad):** Indica la masa volumétrica o el "peso" del vino, la cual varía en función del equilibrio entre su contenido de alcohol y de azúcar, e indica qué tan pesado o aguado se siente en boca.
- **pH:** Mide qué tan puramente ácido es el vino, donde valores más bajos indican mayor frescura, viveza y estabilidad química.
- **sulphates (Sulfatos):** Representa los sulfitos que ayudan a mantener la frescura del vino y actúan como un conservante antimicrobiano adicional.
- **alcohol:** Mide el porcentaje de alcohol por volumen, lo que le da calor y cuerpo para equilibrar la acidez y el dulzor.

DESARROLLO DEL PROYECTO

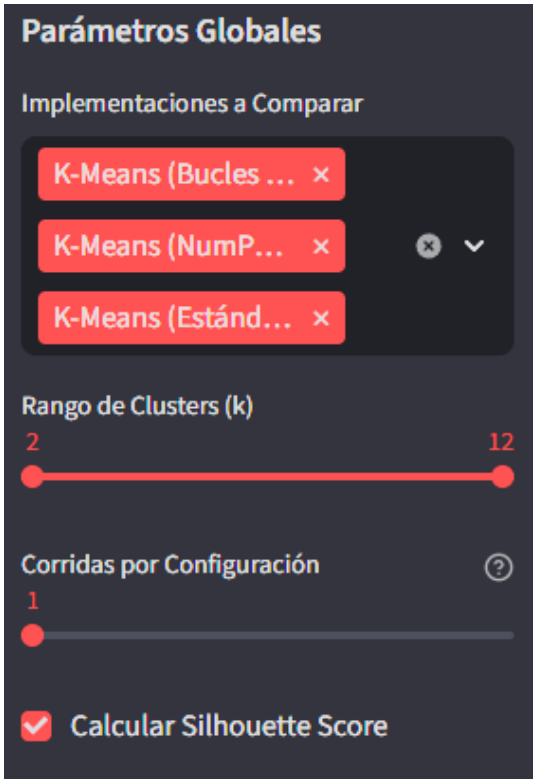
1.PANEL DE CONTROL

El panel de control permite al usuario seleccionar un dataset .arff. Se encuentran precargados en el sistema el dataset original y un dataset sanitizado en donde se han eliminado los outliers mediante el método del Rango Intercuartílico.



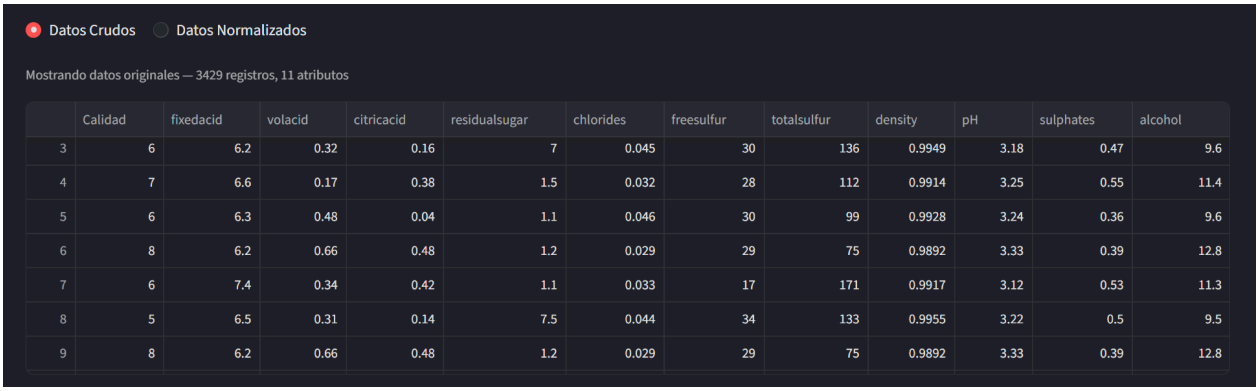
Al cargar el archivo, permite seleccionar qué datos del mismo se utilizarán para el análisis.

Adicionalmente, permite seleccionar los parámetros globales para el benchmark de comparación, entre ellos seleccionar las implementaciones que se va a comparar, el rango de k a utilizar y la cantidad de corridas por cada configuración que se realizará.



2.EXPLORADOR DE DATOS

Permite ver los datos crudos, normalizados y sus estadísticas descriptivas.
La normalización es realizada mediante el algoritmo z-score, dado por `Z = (X - Media) / Desviación_Estándar`



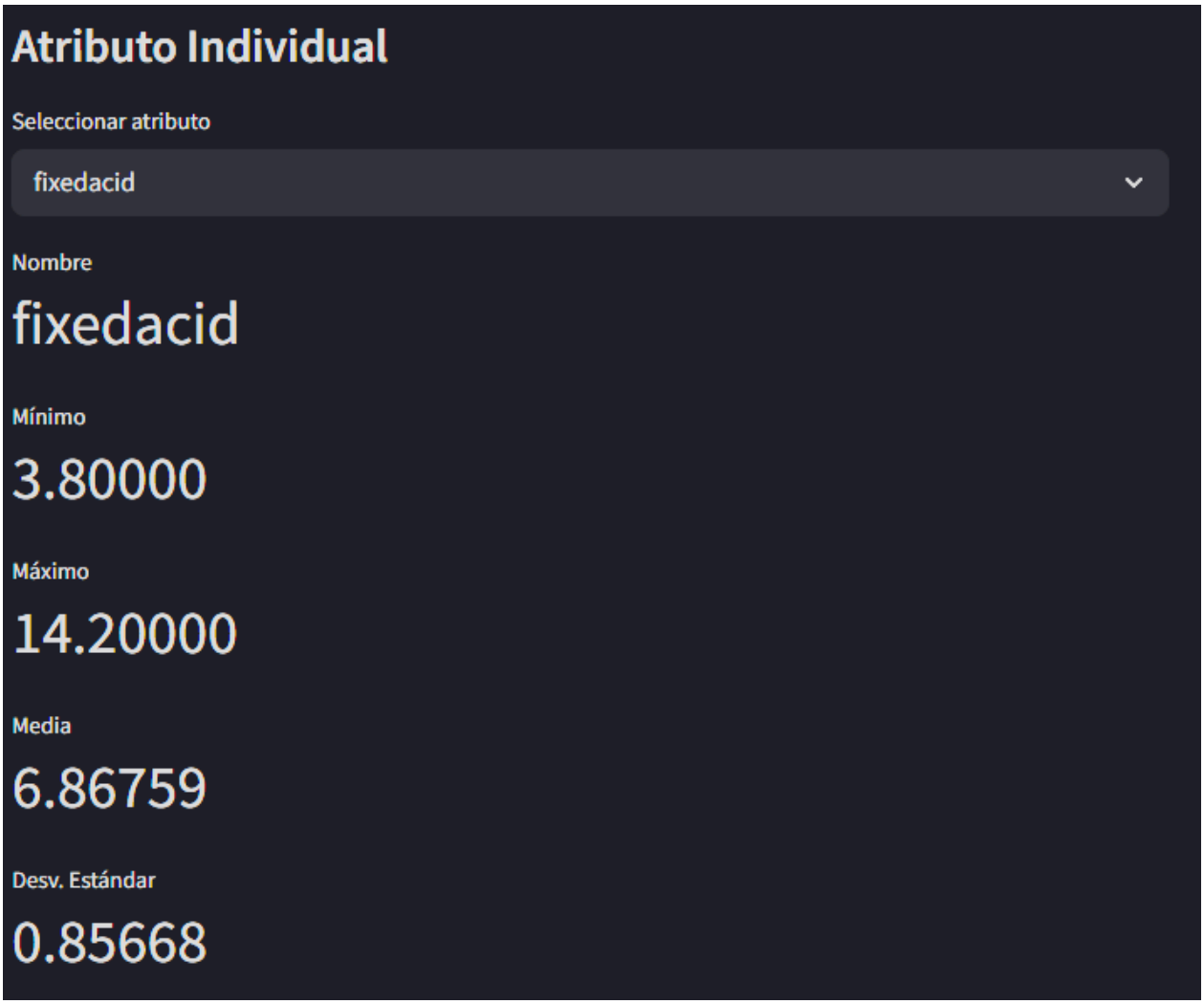
Datos Crudos Datos Normalizados

Mostrando datos originales — 3429 registros, 11 atributos

	Calidad	fixedacid	volacid	citricacid	residualsugar	chlorides	freesulfur	totalsulfur	density	pH	sulphates	alcohol
3	6	6.2	0.32	0.16	7	0.045	30	136	0.9949	3.18	0.47	9.6
4	7	6.6	0.17	0.38	1.5	0.032	28	112	0.9914	3.25	0.55	11.4
5	6	6.3	0.48	0.04	1.1	0.046	30	99	0.9928	3.24	0.36	9.6
6	8	6.2	0.66	0.48	1.2	0.029	29	75	0.9892	3.33	0.39	12.8
7	6	7.4	0.34	0.42	1.1	0.033	17	171	0.9917	3.12	0.53	11.3
8	5	6.5	0.31	0.14	7.5	0.044	34	133	0.9955	3.22	0.5	9.5
9	8	6.2	0.66	0.48	1.2	0.029	29	75	0.9892	3.33	0.39	12.8

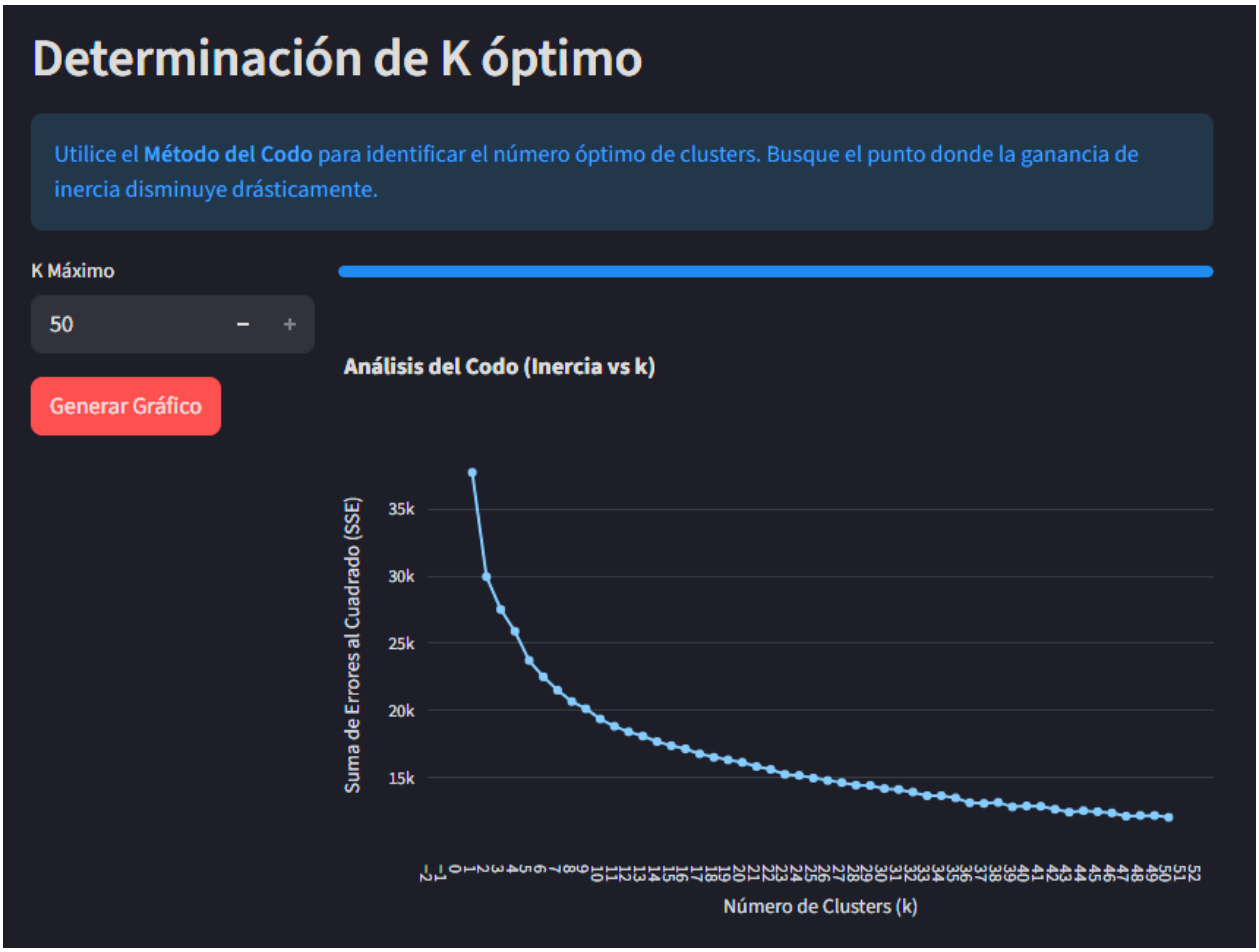
Estadísticas Descriptivas

	count	mean	std	min	25%	50%	75%	max
Calidad	3429.0000	5.8810	0.8890	3.0000	5.0000	6.0000	6.0000	9.0000
fixedacid	3429.0000	6.8676	0.8567	3.8000	6.3000	6.8000	7.3000	14.2000
volacid	3429.0000	0.2781	0.1004	0.0800	0.2100	0.2600	0.3200	0.9650
citricacid	3429.0000	0.3349	0.1215	0.0000	0.2700	0.3200	0.3900	1.6600
residualsugar	3429.0000	6.4058	5.1045	0.6000	1.7000	5.2000	10.0000	65.8000
chlorides	3429.0000	0.0456	0.0220	0.0090	0.0360	0.0430	0.0500	0.3460
freesulfur	3429.0000	35.3199	17.3562	2.0000	23.0000	34.0000	46.0000	289.0000
totalsulfur	3429.0000	138.1464	42.7031	9.0000	108.0000	134.0000	167.0000	440.0000
density	3429.0000	0.9940	0.0030	0.9871	0.9918	0.9938	0.9961	1.0390
pH	3429.0000	3.1852	0.1501	2.7200	3.0800	3.1800	3.2800	3.8100


Atributo Individual
Seleccionar atributo
fixedacid
Nombre
fixedacid
Mínimo
3.80000
Máximo
14.20000
Media
6.86759
Desv. Estándar
0.85668

2.DETERMINACIÓN DEL K ÓPTIMO

K-means es dependiente de encontrar la cantidad óptima de clusters para agrupar los datos. Para esto implementamos el método del codo, el cual es un método gráfico para encontrar el valor de K óptimo. Muestra los valores de la suma de cuadrados dentro del grupo (inercia o WCSS) con respecto a k. El valor óptimo es cuando la ganancia de inercia disminuye drásticamente.

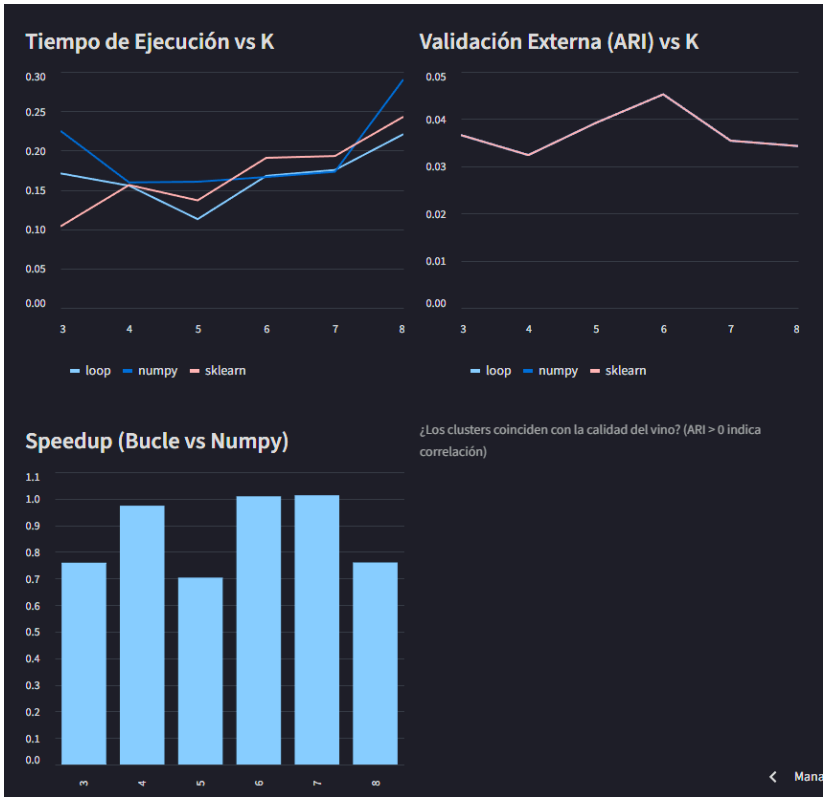


UBP
UNIVERSIDAD
Blas Pascal

3.COMPARATIVA DE RENDIMIENTO Y CALIDAD

Ejecuta benchmarks en base a las configuraciones definidas en el panel de control. Hace comparaciones entre las 3 implementaciones y los k seleccionados.

	Implementación	k	Inercia	Tiempo (s)	Iteraciones	Silhouette	ARI	NMI
0	loop	3	27518.87	0.170902	38.5	0.142	0.037	0.047
1	loop	4	25330.00	0.155210	26.5	0.156	0.032	0.045
2	loop	5	23719.52	0.112730	25.5	0.140	0.039	0.058
3	loop	6	22495.22	0.167789	52.0	0.143	0.045	0.077
4	loop	7	21502.80	0.175203	35.0	0.127	0.035	0.075
5	loop	8	20658.39	0.220788	51.5	0.124	0.034	0.074
6	numpy	3	27518.87	0.225055	38.5	0.142	0.037	0.047
7	numpy	4	25330.00	0.159401	26.5	0.156	0.032	0.045
8	numpy	5	23719.52	0.160288	25.5	0.140	0.039	0.058
9	numpy	6	22495.22	0.166278	52.0	0.143	0.045	0.077



UBP
UNIVERSIDAD
Blas Pascal

Implementaciones:

Se han implementado 3 versiones de K-means:

Versión No Vectorizada (KMeansLoop)

Esta versión itera explícitamente sobre cada punto y cada centroide. Es fácil de leer pero ineficiente en Python debido al GIL y overhead de interpretación. Complejidad: Efectúa N×K×D operaciones de distancia por iteración en bucles Python puros.

Versión Vectorizada (KMeansNumpy)

Utiliza Broadcasting de NumPy para calcular la matriz de distancias de todos los puntos contra todos los centroides en una sola operación de bajo nivel (C).

distancias = np.linalg.norm(X[:, np.newaxis] - centroides, axis=2)

labels = np.argmin(distancias, axis=1)

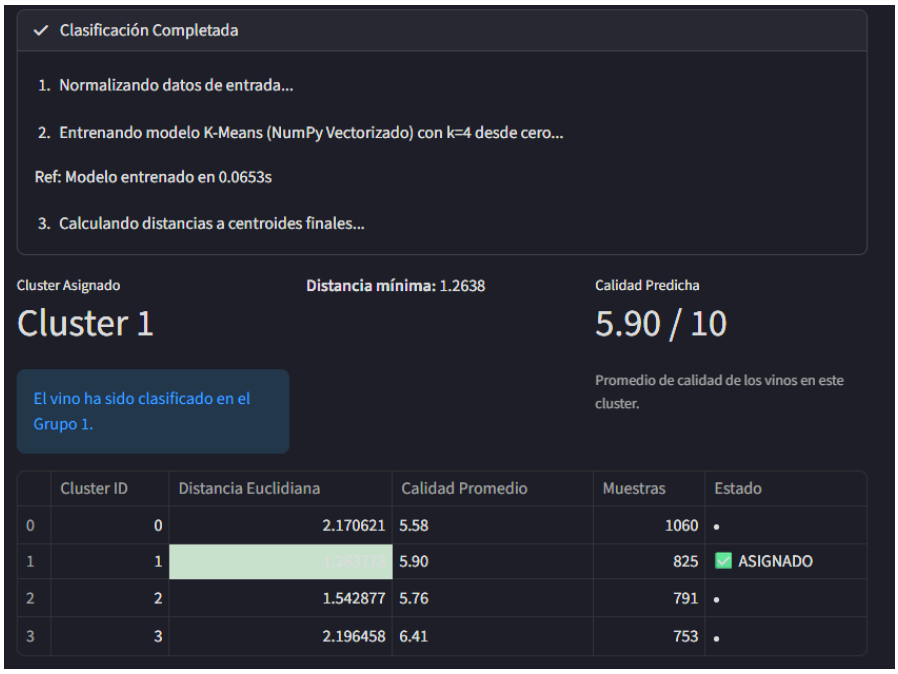
Esta implementación es entre 50 y 100 veces más rápida, demostrando la importancia de la vectorización en ciencia de datos.

Versión de Control (KMeansSklearn)

Ejecuta e instancia el algoritmo de K-means provisto por la librería Sklearn a modo de Patrón Oro para usar de referencia con las demás implementaciones.

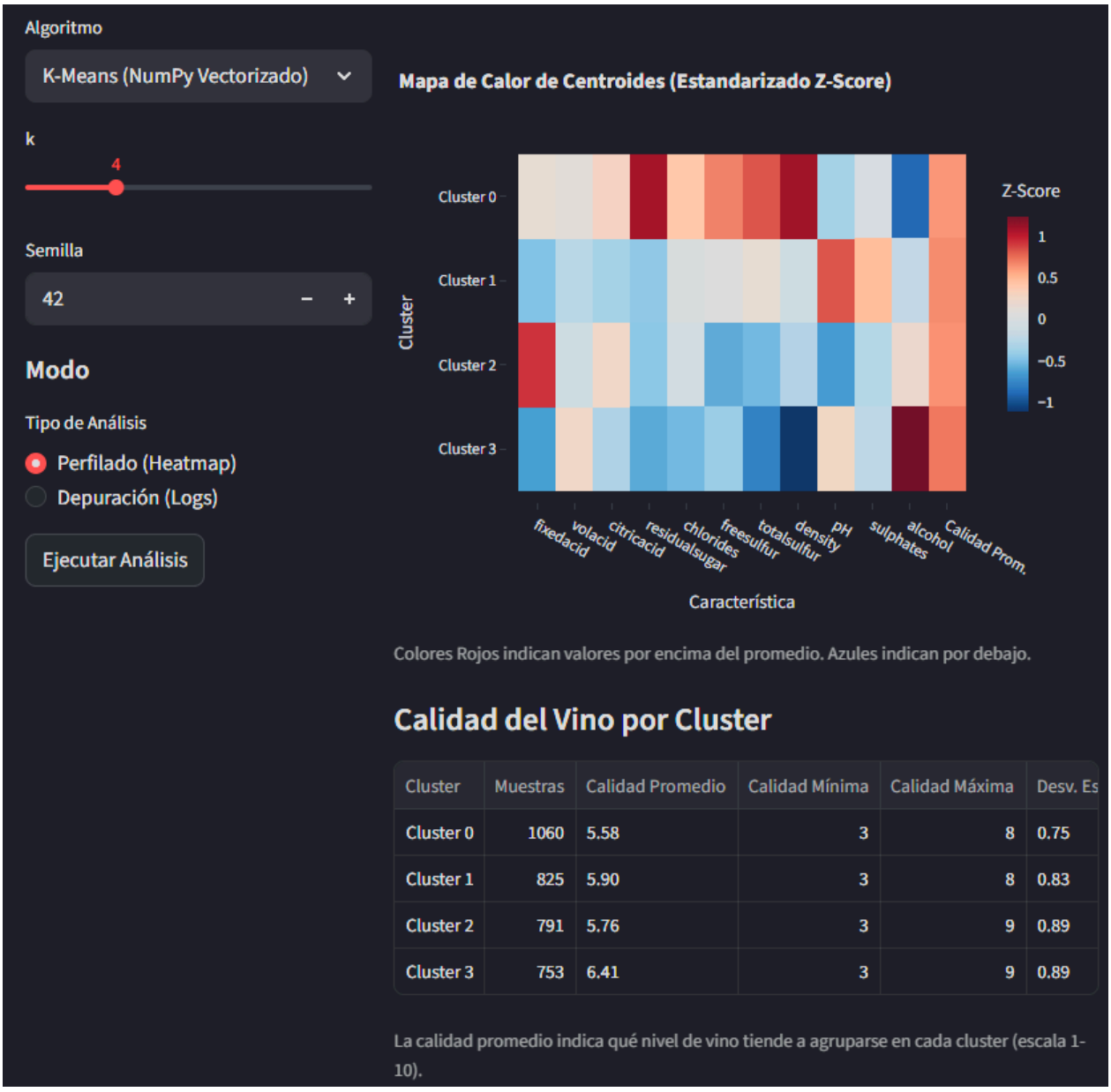
4.PREDICCIÓN

Simula la llegada de un nuevo vino. El sistema permite elegir la implementación a utilizar, el k a utilizar, y la semilla para generar los clusters. Al ejecutar, se entrena el modelo, se clasifica la muestra y se da el promedio de calidad del cluster.



5.ANÁLISIS DETALLADO

Panel cuya finalidad es depurar e interpretar el modelo entrenado. En su modo de perfilado, presenta un mapa de calor de los centroides, haciendo visible las características únicas de cada cluster generado. A su vez, ofrece una tabla con la calidad promedio, la mínima, la máxima, la desviación estándar y el total de muestras de cada cluster.



CONCLUSIONES

Este trabajo nos ha permitido conocer en profundidad el algoritmo K-Means desde sus cimientos matemáticos, comprobar empíricamente la gran diferencia de rendimiento entre un código nativo, uno vectorizado y una librería consumida, a la vez que se desarrolló una herramienta analítica completa e interactiva, que no solo ejecuta el algoritmo, sino que permite explorar los datos y validar los resultados visualmente. Se han aplicado las buenas prácticas de la Ingeniería de Software para asegurar la mantenibilidad a través de patrones de diseños como Data Transfer Object y Factory Method, y el desarrollo de una suite de tests automáticos para el TDD implementado a lo largo de la vida del proyecto.

BIBLIOGRAFÍA

- Russell, S. & Norvig, P. (2021). Artificial Intelligence: A Modern Approach. (Capítulos sobre Agentes y Aprendizaje Automático).
- Material de Cátedra SIA. Resumen de diapositivas (Clase 3: Clustering).
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. (Sección 9.1: K-Means Clustering).
- NumPy Documentation. Broadcasting rules y operaciones de álgebra lineal.