

# Sviluppo e confronto di algoritmi di fuzzy clustering

Andrea D'Amicis 869008

Relatore: Prof. Davide Elio Ciucci  
Correlatore: Dott. Andrea Campagner  
A.A. 2022/2023

# Obiettivi del tirocinio

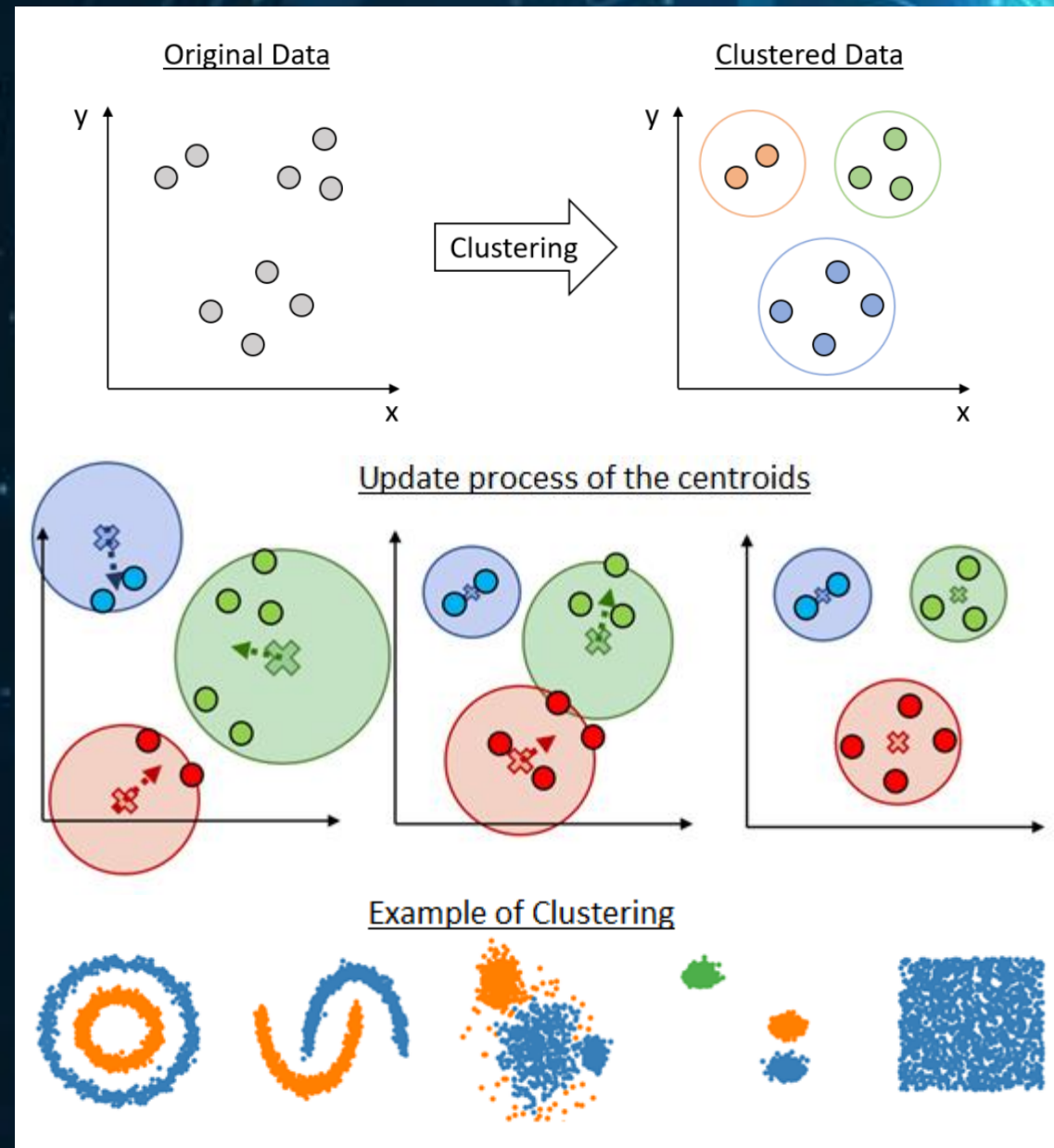
- Studiare il Clustering.
- Ricercare algoritmi di **Fuzzy Clustering** analizzando articoli su **Scopus**.
- Sviluppare gli algoritmi in Python.
- Effettuare dei confronti e **analizzare le prestazioni** di ognuno degli algoritmi implementati con dei Dataset di test.





# Cos'è il clustering

Il Clustering è un tipo **apprendimento automatico non supervisionato** che si pone l'obiettivo di definire delle tecniche di **selezione e raggruppamento** che consentano di apprendere in quali cluster è possibile dividere un dataset di dati descritto da **features**.



# Cos'è il fuzzy clustering

Il **Fuzzy Clustering** è un tipo di **Soft Clustering** in cui ogni punto dato può appartenere a più di un cluster con un valore di probabilità o un grado di appartenenza chiamato **membership degree**.

$$\sum_{j=1}^c u_{ij} = 1 \quad \forall i \in [1, n]$$

## Algorithm 1 Fuzzy C-Means

**Inputs:**  $X, c, m, \tau$

**for**  $t = 1$  **to**  $\tau$  **do**

$$u_{ij} = \left[ \sum_{k=1}^c \left( \frac{\|x_i - v_j\|^2}{\|x_i - v_k\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \quad \forall i \in [1, n], \forall j \in [1, c]$$

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m} \quad \forall j \in [1, c]$$

**end for**

**Outputs:**  $D$

Legenda:

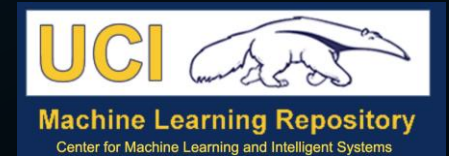
- $X$ : dataset
- $c$ : numero di cluster
- $m$ : grado di fuzziness (tipicamente è 2)
- $\tau$ : numero di iterazioni
- $v_j$ : centroide  $j$ -esimo
- $u_{ij}$ : grado di membership tra  $i$ -esimo elemento del dataset e  $j$ -esimo centroide
- $D$ : matrice dei gradi di membership

## Algoritmi implementati

- **Robust Fuzzy C-Means** (FCM- $\sigma$ ) utilizza una **funzione robusta** che riduce l'influenza dei valori anomali (outlier) sui risultati del clustering.
- **Kernelized Fuzzy C-Means** (KFCM) utilizza una **funzione di kernel** per mappare i dati in uno spazio di dimensione superiore rendendo più efficiente il processo di clustering.
- **Credibilistic Fuzzy C-Means** (CFCM) effettua una **valutazione spaziale** incorporando anche una nozione di credibilità.
- **Size-insensitive Fuzzy C-Means** (csiFCM) utilizza un **parametro di pesatura** che tiene conto della **dimensione dei cluster**, consentendo di creare cluster con dimensioni più equilibrate.



# Dataset

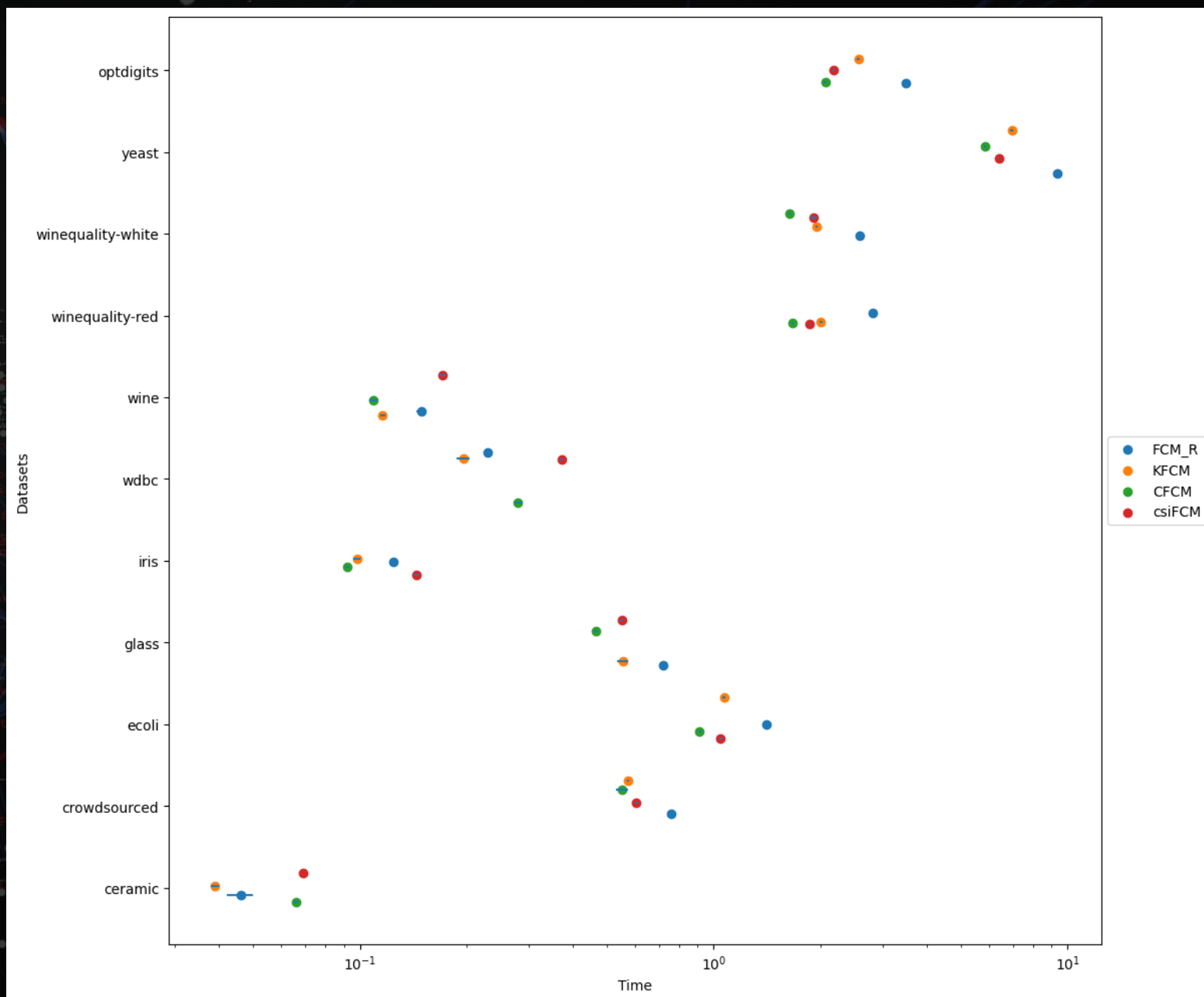


Tutti i dataset che sono stati usati come test per i 4 algoritmi sviluppati provengono dalla **UCI Machine Learning Repository** e di seguito vi sono riportati i nomi di ciascuno di essi con le relative dimensioni:

Nome dataset	N° di istanze	N° di features	N° di classi
Chemical Composition of Ceramic	88	17	2
Crowdsourced Mapping	300	28	6
Ecoli	336	7	8
Glass Identification	214	9	7
Iris	150	4	3
Optical Recognition	500	64	10
Breast Cancer Wisconsin	569	32	2
Wine	178	13	3
Wine-quality-red	500	11	9
Wine-quality-white	500	11	9
Yeast	1483	8	10

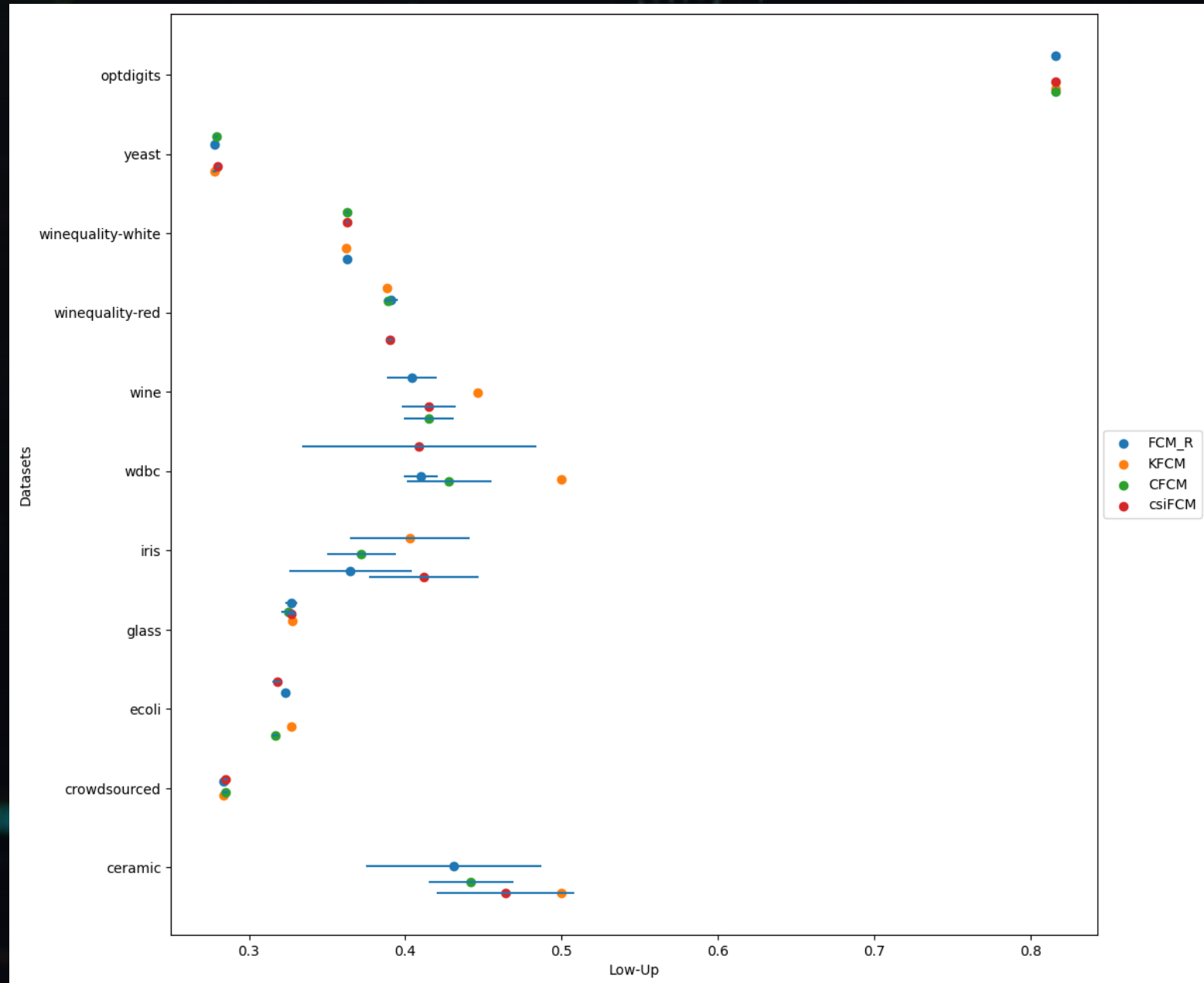
# Risultati – Plot tempo di esecuzione

- Nell'81% dei casi l'algoritmo **CFCM** è risultato essere il **più veloce**.
- Nel 63% dei casi l'algoritmo **FCM- $\sigma$**  è risultato essere il **più lento**.



# Risultati - Plot errore

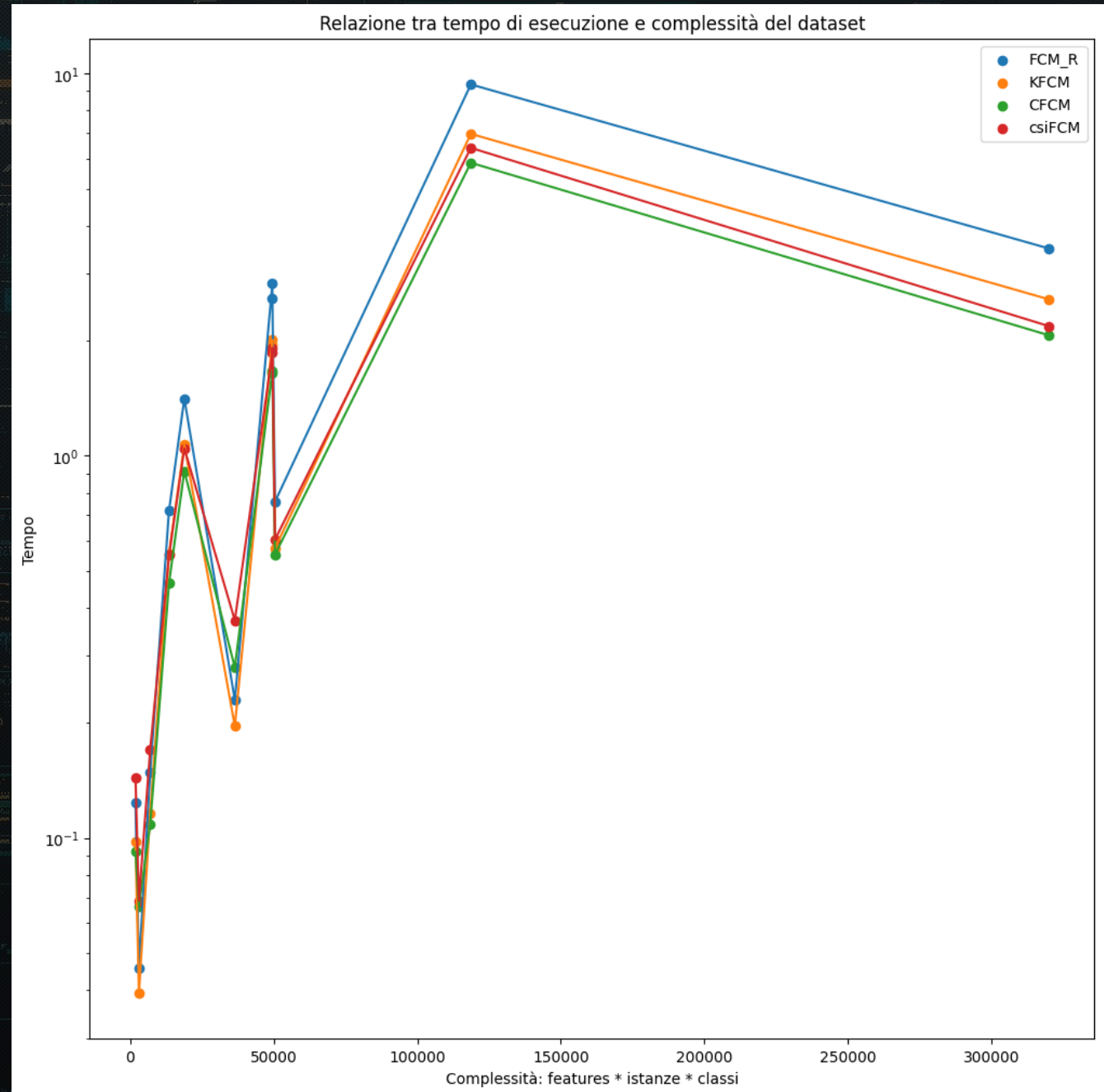
- Nella maggior parte dei casi è compreso tra 0,28 e 0,45.
- L'errore ottenuto per ogni algoritmo è pressoché **identico** e quindi conviene utilizzare gli algoritmi coi tempi di esecuzione più bassi.





# Risultati – Plot tempo/complessità

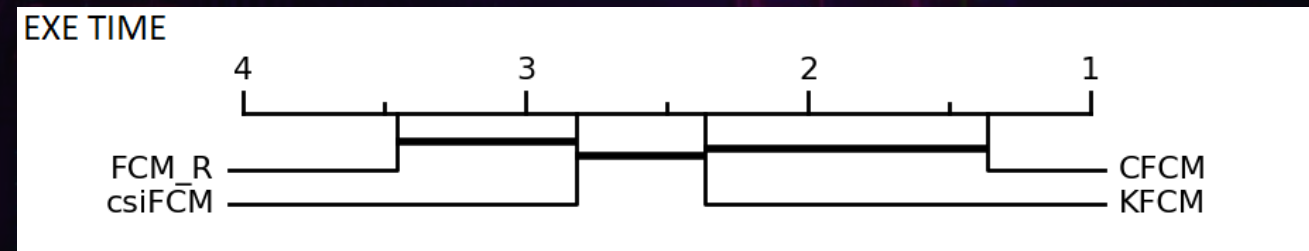
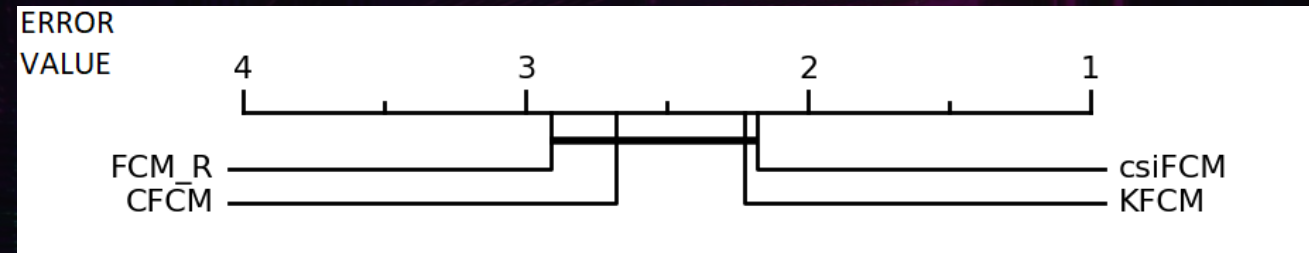
- I tempi di esecuzione degli algoritmi aumentano all'aumentare della complessità del dataset preso in esame.
- Il parametro che influisce maggiormente sui tempi di esecuzione è il **n° di features**.





## Risultati – Plot similarità

- Avendo un errore molto simile, possiamo concludere che non c'è un reale motivo per preferire l'utilizzo di un algoritmo piuttosto che dell'altro.
- Per i tempi di esecuzione è mostrata la classifica che sottolinea, a coppie di 2, la similarità dei tempi di esecuzione tra un algoritmo e il suo successivo in classifica.




# Conclusioni e sviluppi futuri

- Sono riuscito a sviluppare e analizzare degli algoritmi nell'ambito dell'**AI**.
- Sviluppare algoritmi **ibridi** che fondono diverse logiche potrebbe migliorare il clustering:
  - KFCM- $\sigma$  (Robust Kernel Fuzzy C-Mean)
  - CKFCM (Credibilistic Kernelized Fuzzy C-Means)
- Oppure implementare altri algoritmi con logiche differenti:
  - IFCM (Intuitionistic Fuzzy C-Means)
  - DPIFCM (Density Based Probabilistic Intuitionistic Fuzzy C-Means)
  - T2FCM (Type-2 Fuzzy C-Means)







# Grazie per l'attenzione

---

- Andrea D'Amicis 869008
- a.damicis1@campus.unimib.it