

# K-Means Clustering

Luca Alessi - 13286A

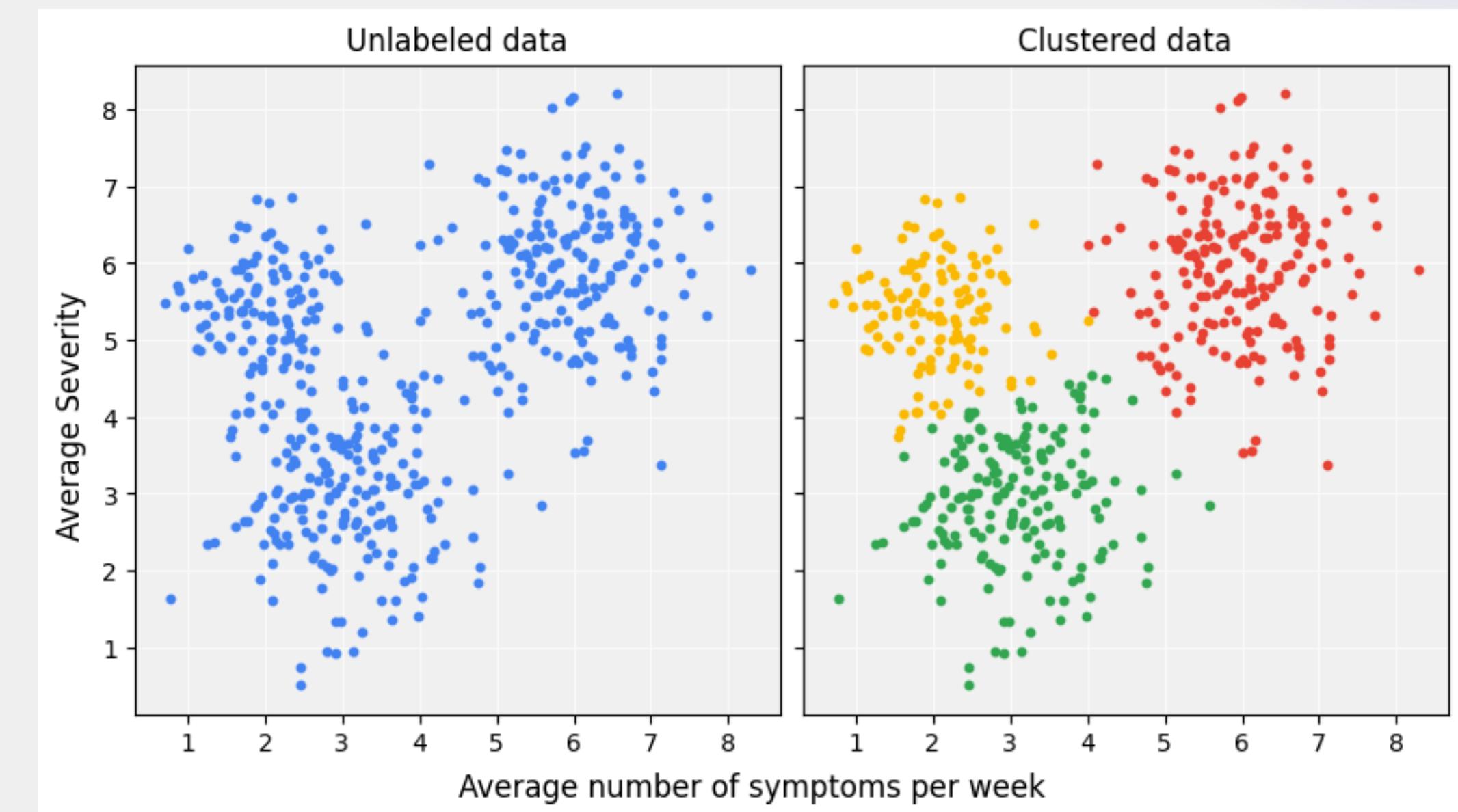
Enrico Dalla Stella - 10043A

Cristian Di Cillo - 12286A

# Clustering

Principi e Modelli della Percezione

K-Means Clustering



# Continuazione clustering

## Vantaggi:

- Organizzare dati complessi
- Scoprire pattern nascosti
- Flessibile

## Applicazioni pratiche:

- Segmentazione delle immagini
- Riconoscimento uditivo
- Percezione spaziale



# Dal caos all'ordine: il clustering nella percezione

## LEGGI DELLA GESTALT

- come il cervello percepisce e organizza stimoli in strutture coerenti
- prossimità: punti vicini vengono raggruppati
- similarità: oggetti simili percepiti come parte dello stesso gruppo
- chiusura: il cervello tende a completare figure incomplete

## STIMOLI PROSSIMALI E DISTALI

- stimoli prossimali: input grezzi che raggiungono i sensi
- stimoli distali: rappresentazione elaborata dal cervello
- nel k-means i dati grezzi diventano cluster finali (stimoli distali)
- da stimoli prossimali a distali: trasformazione e rappresentazione

## BOTTOM-UP E TOP-DOWN

- bottom-up: dai dati grezzi per costruire una rappresentazione del mondo
- top-down: cervello usa conoscenze per dare significato agli input sensoriali
- bottom-up: analisi iniziale dei dati grezzi.
- top-down: creazione di cluster finali come categorie organizzate.

3

# Supervisionato e non supervisionato

Il clustering può essere:

- Supervisionato: l'algoritmo è in grado di riconoscere le immagini
- Non supervisionato: si basa solo sulle caratteristiche delle immagini



# Introduzione a k-means

- Algoritmo non supervisionato
- Divide i dati in k cluster
- Ogni cluster ha un centroide

## Obiettivo:

- Diminuire distanza intra-classe
- Aumentare distanza inter-classe



# K-means clustering

## Implementazione dell'algoritmo

- Normalizzazione dei dati
- Inizializzazione dei centroidi
- Assegnazione dei punti ai cluster
- Ricalcolo dei centroidi
- Verifica della convergenza

# Normalizzazione

La normalizzazione è fondamentale per evitare che variabili con scale diverse dominino il clustering.

Formule principali:

- Min-Max Normalization:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- Z-score Standardization:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

Quando usarle:

- Min-Max: range uniforme (es. [0,1]).
- Z-score: distribuzioni con varianze diverse



# Inizializzazione dei centroidi

Il primo passo dell'algoritmo K-means è determinare i centroidi iniziali per i cluster, che sono i punti medi dei gruppi che l'algoritmo cercherà di formare.

- Scelta di k: la prima decisione importante è scegliere il numero di cluster, ossia il valore di k.
- Selezione dei centroidi: centroidi vengono scelti casualmente dal dataset, selezionando k punti che fungeranno da centri iniziali per i cluster.





# Assegnazione dei punti al cluster

Una volta scelti i centroidi iniziali, l'algoritmo assegna ogni punto del dataset al cluster il cui centroide è il più vicino.

- Calcolo della distanza: ogni punto è considerato in uno spazio multidimensionale.
- Assegnazione: dopo aver calcolato le distanze, ogni punto viene assegnato al cluster con il centroide più vicino. Questo passaggio crea una struttura iniziale che sarà migliorata nelle iterazioni successive.

# Ricalcolo dei centroidi

Dopo che ogni punto è stato assegnato al proprio cluster, il passo successivo è ricalcolare i centroidi. Ogni nuovo centroide viene determinato come la media delle coordinate di tutti i punti che appartengono al cluster.

Questo processo viene ripetuto iterativamente, affinché i centroidi si spostino verso le posizioni ottimali per rappresentare meglio il centro di ciascun cluster.

# Verifica della Convergenza

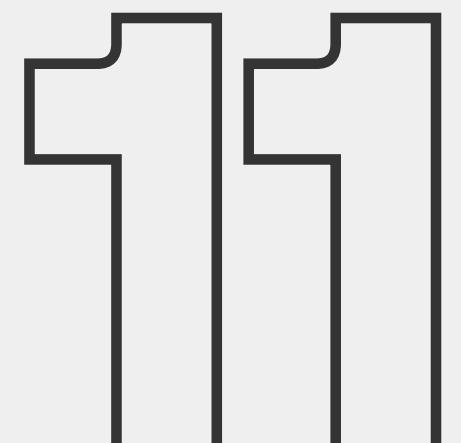
L'algoritmo K-means continua a iterare fino a quando non si verifica una delle seguenti condizioni di convergenza:

1. I centroidi non cambiano più o la loro variazione è inferiore a una soglia predefinita.
2. Si raggiunge il numero massimo di iterazioni.

Questa verifica garantisce che il processo si ferma quando i cluster sono stabili o per evitare loop infiniti in situazioni complesse.

Una volta terminato, è utile misurare la qualità del clustering utilizzando la somma delle distanze intra-cluster (WCSS).

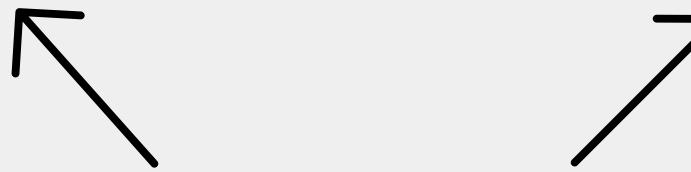
- Un valore basso di WCSS indica che i punti sono concentrati intorno al centroide, suggerendo un buon clustering.
- Un valore alto potrebbe significare che i cluster sono poco definiti.



# 01

## Scelta del numero di cluster (k)

K-means richiede la scelta a priori del numero di cluster, che può essere difficile da determinare. Tecniche come il metodo del gomito e l'indice di silhouette possono aiutare, ma rimane una questione soggettiva.



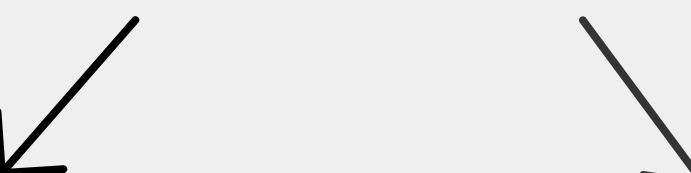
# 03

## Assunzioni sulla forma dei cluster

K-means presuppone cluster sferici e di dimensioni simili, risultando inefficiente per dati con cluster irregolari o di densità variabile. Fuzzy K-means può affrontare meglio situazioni con cluster sovrapposti.

# Limiti di K-Means

## E possibili soluzioni



# 02

## Sensibilità ai centroidi iniziali

La scelta iniziale dei centroidi influisce sul risultato finale. Tecniche come K-means++ migliorano l'inizializzazione, ma non garantiscono sempre il risultato ottimale. Spesso si esegue K-means più volte per ottenere il miglior output.



# 04

## Sensibilità agli outlier e alla metrica di distanza

K-means è influenzato dagli outlier, che possono distorcere i centroidi. Inoltre, l'uso della distanza euclidea come metrica limita l'algoritmo, soprattutto in presenza di dati con caratteristiche o di peso diverso.

# K-Means++

## K-means

Nel K-means classico, i centroidi iniziali sono scelti casualmente dal dataset. Questo approccio può portare a una cattiva distribuzione dei centroidi, il che può risultare in cluster non rappresentativi e aumentare la probabilità che l'algoritmo converga verso un minimo locale subottimale.

## K-means++

In K-means++, il primo centroide è scelto casualmente, mentre i successivi sono selezionati con una probabilità proporzionale alla distanza dal punto più vicino, garantendo una distribuzione più uniforme e migliorando la convergenza.

### Vantaggi Tecnici

K-means++ riduce la probabilità di convergere a minimi locali subottimali e porta a una convergenza più rapida. Inoltre, offre una qualità di clustering superiore e risultati più consistenti rispetto al K-means classico.

# Fuzzy K-Means

## K-means

Nel K-means classico, ogni punto è assegnato in modo rigido a un solo cluster. L'appartenenza di un punto è determinata dalla distanza euclidea dal centroide del cluster, e ogni punto appartiene esclusivamente al cluster più vicino.

## Fuzzy K-Means

Fuzzy K-means assegna a ogni punto un grado di appartenenza a ciascun cluster, compreso tra 0 e 1. I punti più vicini ai centroidi hanno gradi di appartenenza maggiori. Il parametro  $m$  regola la "fuzziness", controllando la sfumatura nell'appartenenza ai cluster.

### Vantaggi Tecnici

Fuzzy K-means consente una segmentazione più flessibile dei dati, particolarmente utile quando i confini tra i cluster sono sovrapposti o sfumati. Inoltre, risulta più robusto rispetto ai dati rumorosi o agli outlier, poiché un punto può appartenere parzialmente a più cluster.



# 01

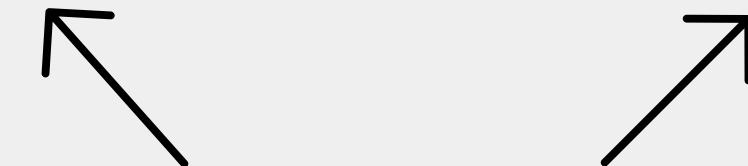
## Semplicità ed efficienza

La sua **efficienza computazionale** lo rende adatto a grandi insiemi di dati che quindi permettono una facile scalabilità

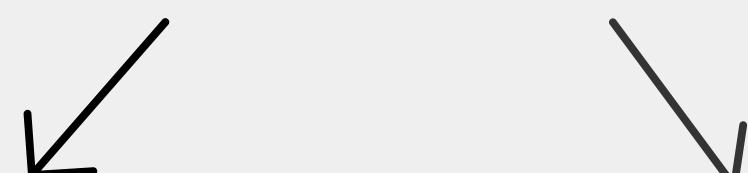
# 02

## Scalabilità

K-Means è in grado di gestire una **grande quantità di dati**, il che lo rende ideale per le applicazioni di **big data**.



# Ragioni per cui è popolare



# 15

# 03

## Versatilità

K-Means si rivela uno strumento versatile e utilizzabile in diversi campi dalla **segmentazione dei clienti** al **raggruppamento dei documenti**.

# 04

## Segmentazione delle immagini:

Nell'**analisi delle immagini**, K-Means è ampiamente utilizzato per compiti come la segmentazione delle immagini, dove l'obiettivo è quello di **suddividere un'immagine in regioni significative**.

# Segmentazione delle immagini

La segmentazione delle immagini è una delle applicazioni più comuni del clustering K-Means nell'analisi delle immagini. Consente di **scomporre** immagini complesse in **regioni gestibili**, rendendo più efficienti attività come il **rilevamento** e il **riconoscimento** di oggetti.

È particolarmente utile nelle immagini mediche, nelle immagini satellitari e nelle applicazioni di fotoritocco.

01.

## Estrazione dei dati dei pixel

Ogni pixel può essere rappresentato dai suoi valori di colore, come per esempio i valori RGB.

02.

## Raggruppamento

Si cerca di raggruppare i valori dei pixel così che **ogni cluster** rappresenterà un **diverso segmento dell'immagine**.

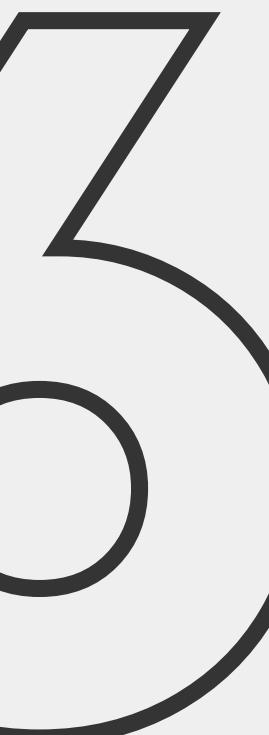
Ad esempio: in una scena naturale, i cluster potrebbero rappresentare oggetti diversi come il cielo, gli alberi e l'acqua.

03.

## Creazione di immagini segmentate

È possibile creare una nuova immagine in cui ogni **pixel** è **colorato** in base all'assegnazione del cluster.

In questo modo si evidenziano visivamente i diversi segmenti dell'immagine.



# Quantizzazione del colore

La quantizzazione del colore è un'altra affascinante applicazione di K-Means. Si tratta di **ridurre il numero di colori** in un'immagine mantenendo la massima qualità visiva possibile così da poter ridurre in modo significativo le dimensioni del file dell'immagine senza una notevole perdita di qualità visiva

Si tratta di un aspetto cruciale per la **compressione delle immagini** e l'archiviazione efficiente.

01.

## Estrazione del colore

Si incomincia estraendo tutti i colori unici dell'immagine. Ogni colore è rappresentato come un punto in uno spazio di colore ad esempio, RGB.

02.

## Raggruppamento

Si cerca ancora una volta di raggruppare i colori in K cluster. Il **centroide** di ogni cluster rappresenterà un **colore dominante** nell'immagine.

03.

## Ricostruire l'immagine

**Sostituendo il colore** di ogni pixel dell'immagine originale con il colore del centroide più vicino è possibile ricostruire immagine.  
In questo modo si riduce il numero complessivo di colori dell'immagine

# Rilevamento degli oggetti

Sebbene il K-Means Clustering da solo non sia tipicamente utilizzato per il rilevamento degli oggetti, svolge un ruolo di supporto nelle fasi di pre-elaborazione. **Aiuta a semplificare e organizzare i dati dell'immagine**, rendendo più efficaci gli algoritmi di rilevamento degli oggetti.

Organizzando l'immagine in cluster significativi, K-Means riduce la complessità delle successive fasi di rilevamento degli oggetti, rendendo il processo più efficiente e accurato.

01.

## Proposte di regioni

Si utilizza K-Means per segmentare l'immagine in **regioni che potrebbero contenere oggetti**.

02.

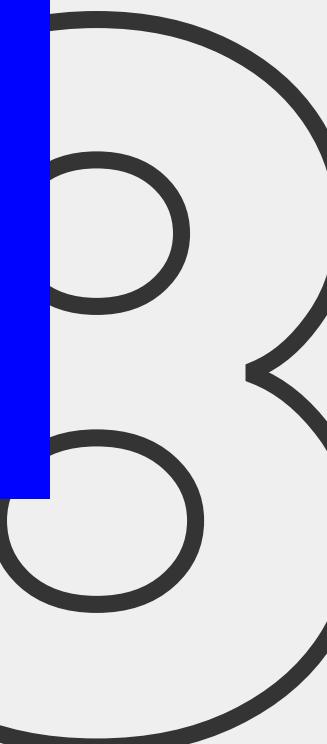
## Estrazione delle caratteristiche

Si estraggono poi le caratteristiche da queste regioni, come **bordi** e **texture**, utili per rilevare e **classificare** gli oggetti.

03.

## Classificazione degli oggetti

Si applica un **algoritmo di classificazione**, come una rete neurale, per identificare ed etichettare gli oggetti all'interno di queste regioni.



# Analisi della texture

L'analisi delle texture comporta l'esame dei modelli visivi all'interno di un'immagine per classificare o riconoscere le texture.

Questo metodo è particolarmente utile in settori come la **produzione tessile**, la **scienza dei materiali** e tutti i campi in cui la comprensione delle proprietà superficiali è fondamentale.

01.

## Estrazione di caratteristiche

Il primo passo da eseguire è sicuramente l'estrazione di caratteristiche della **texture** dall'immagine.

02.

## Raggruppamento

Si applica poi il K-Mean per raggruppare queste caratteristiche di texture.  
Ogni **cluster** rappresenta un tipo di texture distinta all'interno dell'immagine.

03.

## Classificazione delle texture

E per finire si assegnano delle **etichette** a questi cluster in base alla texture che rappresentano.

Ad esempio: liscia, ruvida, ripetitiva.

# Metodo del gomito

il metodo del gomito può essere utile per determinare il **numero ideale di cluster**.

1. Calcolo della **WCSS** (*Within Cluster Sum of Squares*) per ogni cluster (k).  
Si fa quindi la somma della distanza quadratica tra ciascun punto e il centroide di un cluster

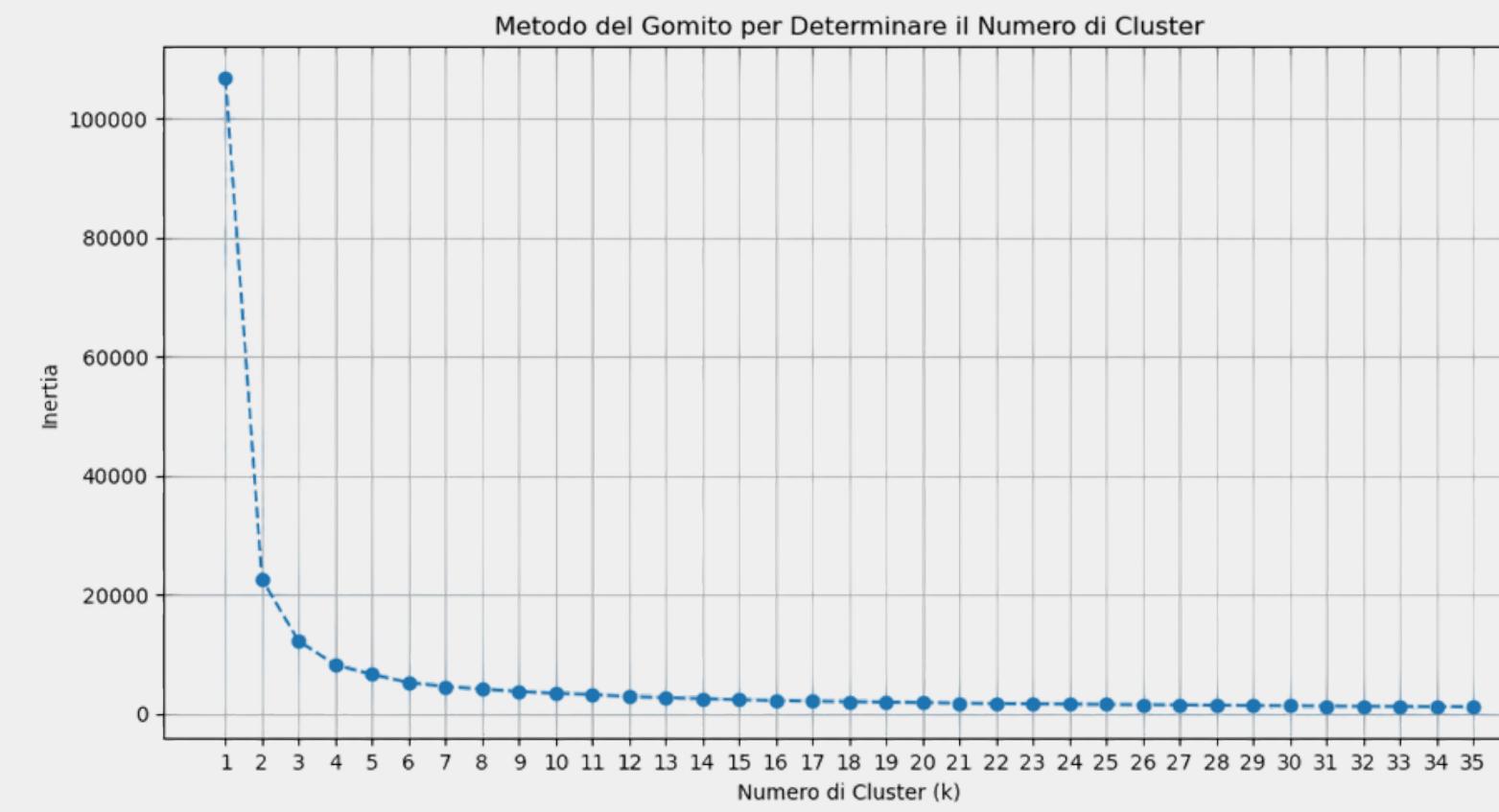
$$WCSS = \sum_{C_k}^{C_n} \left( \sum_{d_i \in C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

$C$  è il **centroide** del cluster e  $d$  è il **punto dati** in ciascun cluster.

2. Quindi, il valore **WCSS** viene tracciato lungo l'asse delle **ordinate** (y) e il numero di **cluster** sull'asse delle **ascisse** (x).  
All'aumentare del numero di cluster, il valore WCSS inizierà a diminuire sostanzialmente fino a raggiungere lo 0 quando ogni punto dati rappresenta un singolo cluster.



# Metodo del gomito



Come si può notare dal grafico, quando **k = 3** l'*inertia* (un'implementazione pratica della WCSS per scopi di ottimizzazione) smette di diminuire significativamente, indicando così il **numero ottimale di cluster**.



**Immagine Originale**



**Immagine Segmentata K= 10**



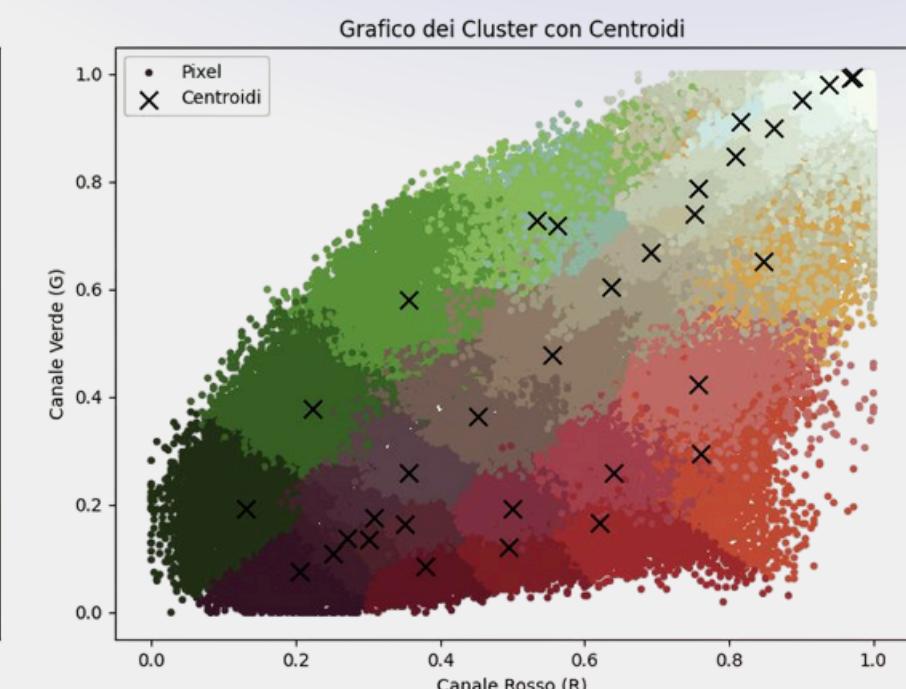
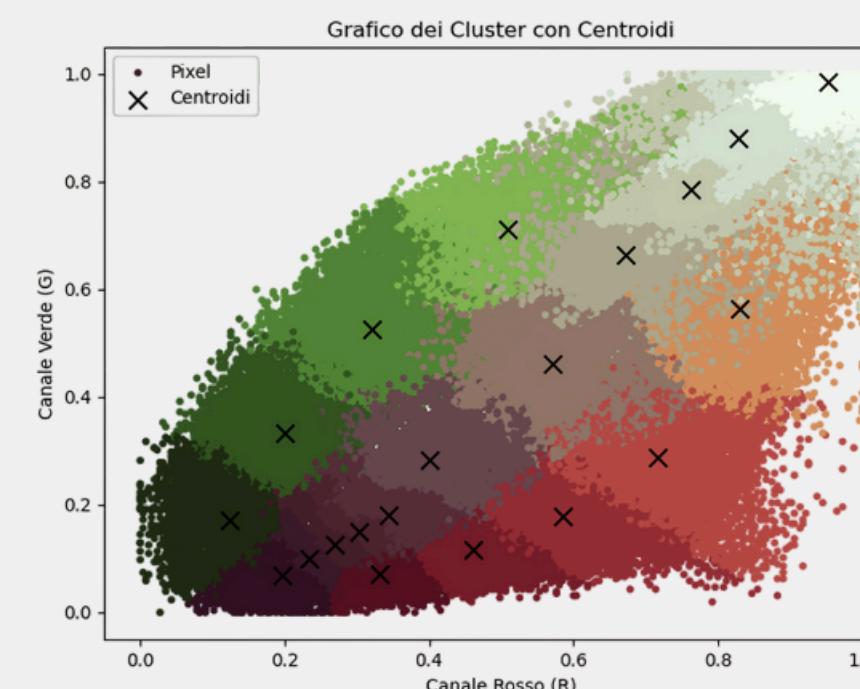
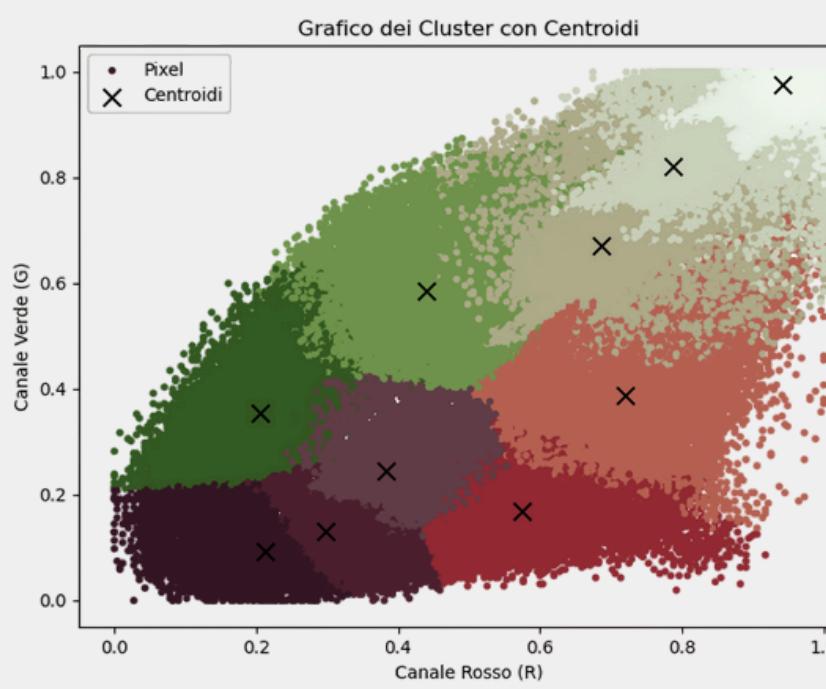
**Immagine Segmentata K= 20**



**Immagine Segmentata K= 33**



## Risultati ottenuti con K-Mean



**Immagine Originale**



**Immagine Segmentata K= 10**



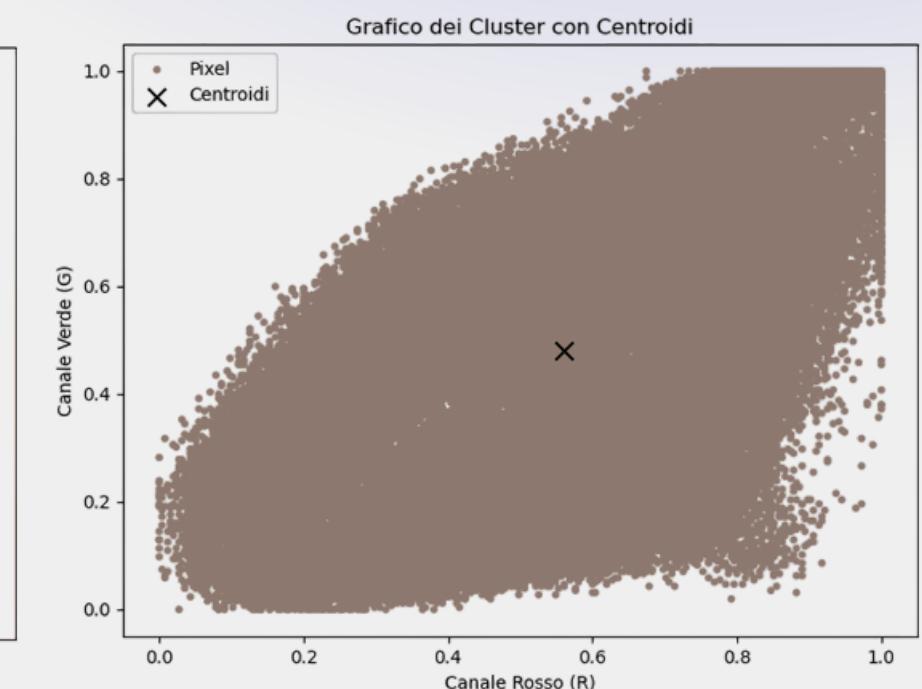
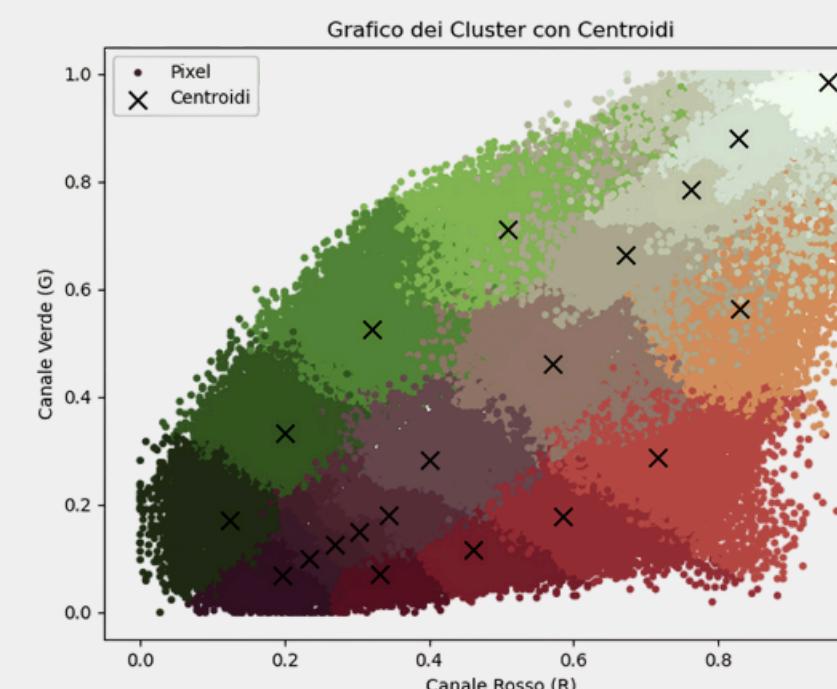
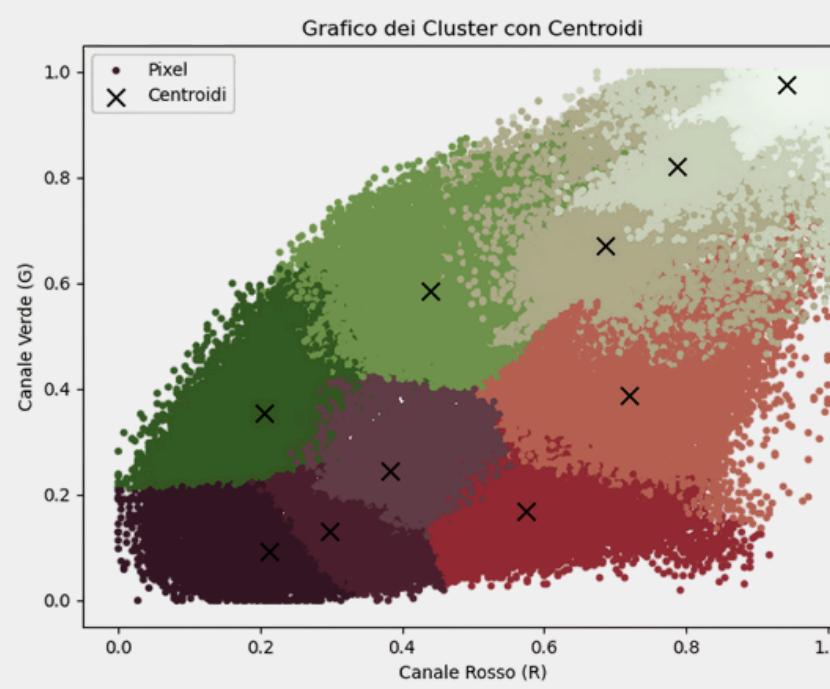
**Immagine Segmentata K= 20**



**Immagine Segmentata K= 33**



## Risultati ottenuti con K-Mean



**Immagine Originale**



**Immagine Segmentata K= 10**



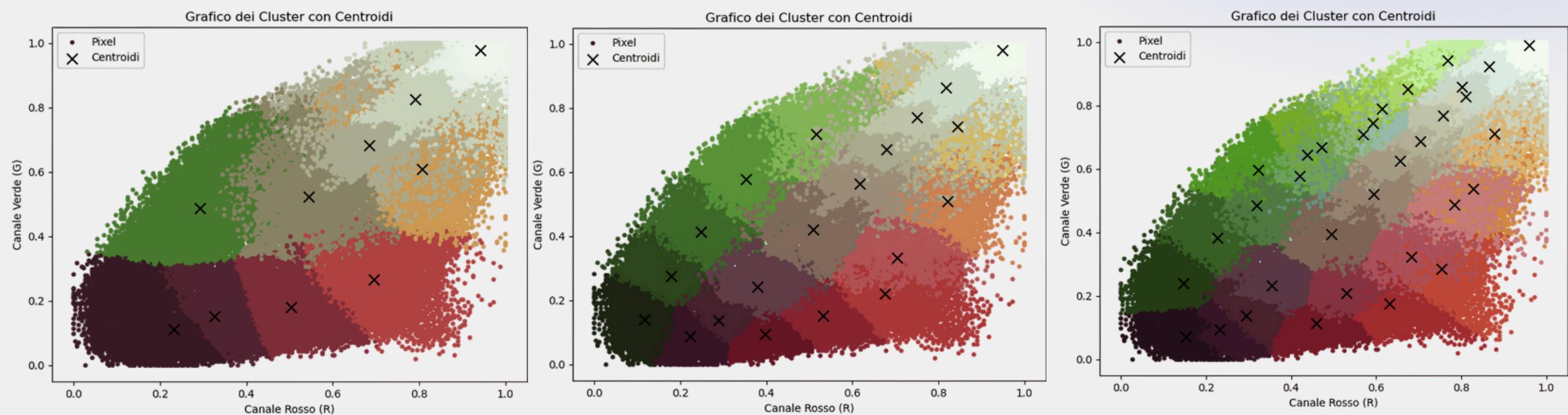
**Immagine Segmentata K= 20**



**Immagine Segmentata K= 33**



**Risultati ottenuti  
con K-Means++**



**Immagine Originale**



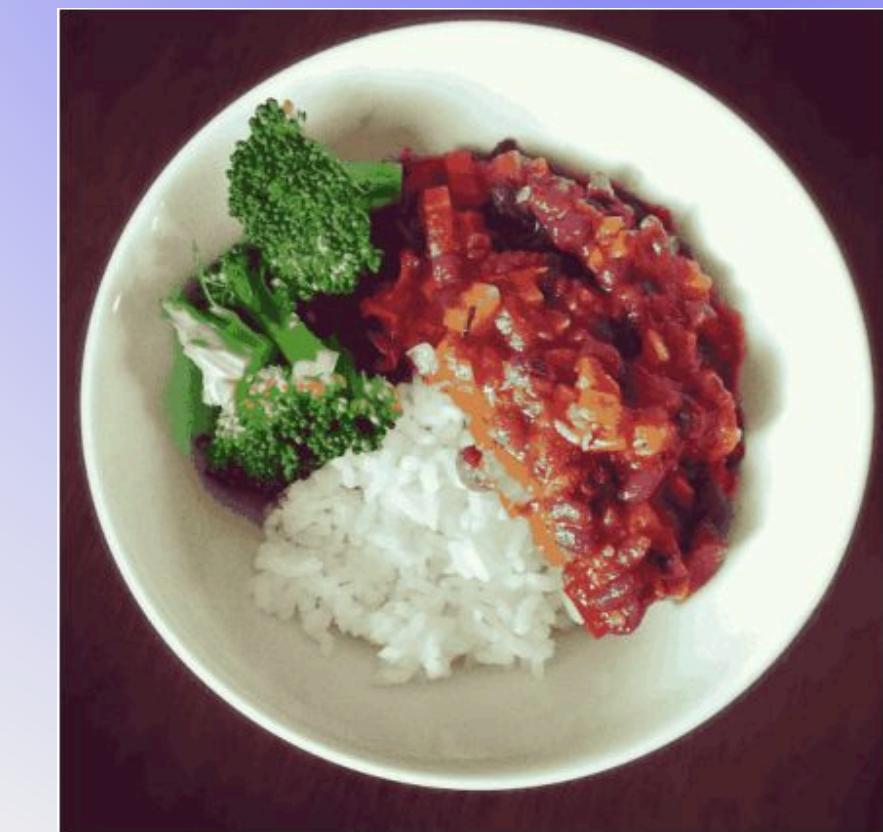
**Immagine Segmentata K= 10**



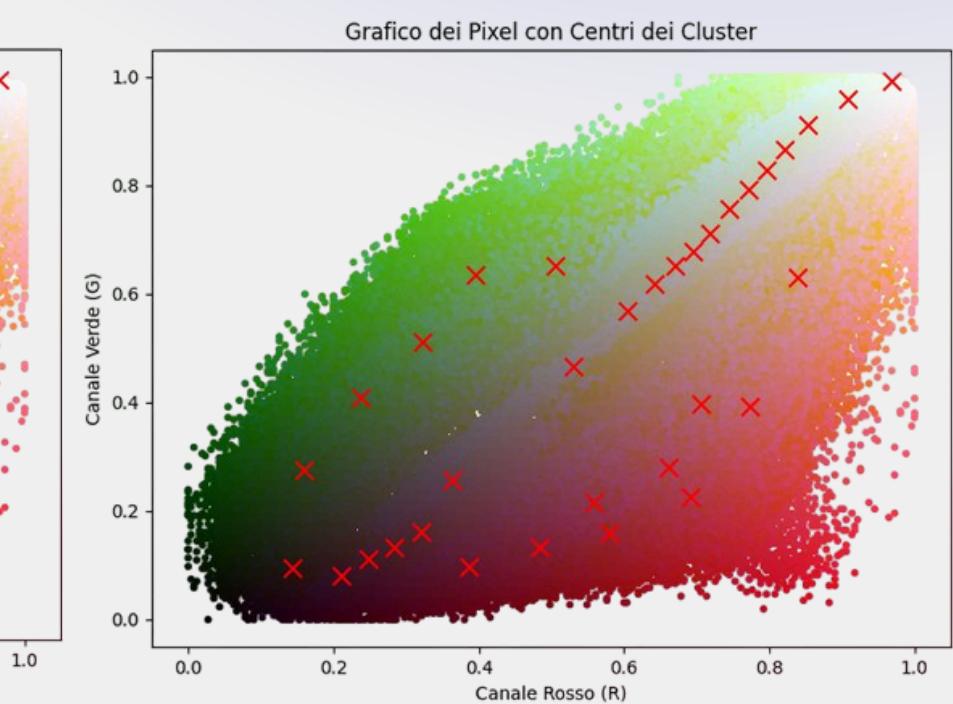
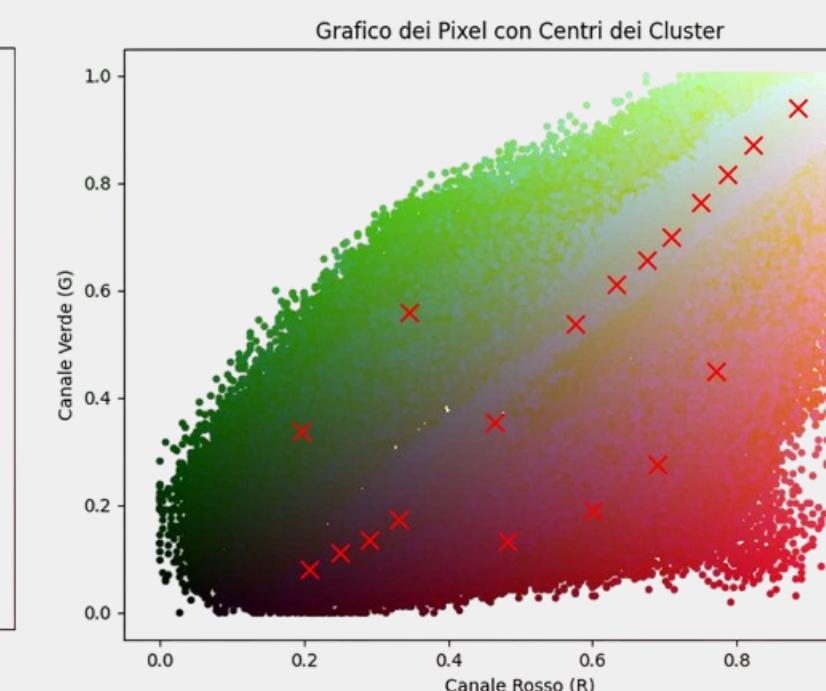
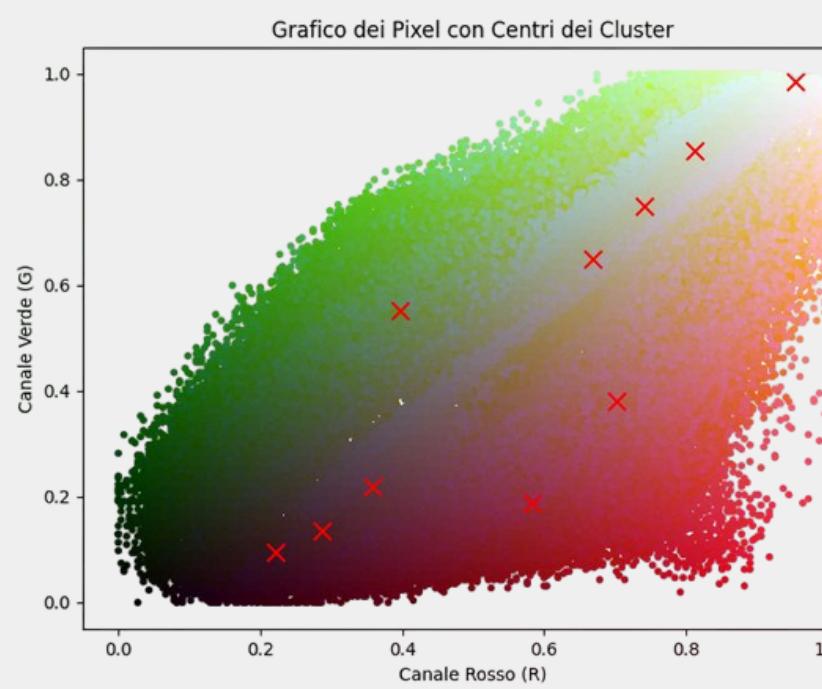
**Immagine Segmentata K= 20**



**Immagine Segmentata K= 33**



**Risultati ottenuti  
con K-Fuzzy**



**Grazie!**