Modello di Machine Learning

1. Introduzione

Questo documento ha lo scopo di spiegare i passaggi impiegati per costruire il modello di Machine Learning e ottenere la segmentazione della clientela.

L'attività per la creazione del modello prevede non solo la sua mera definizione ma anche dei passaggi a volte meno considerati come l'eliminazione degli outliers e lo scaling dei valori.

Verranno poi spiegati anche le metriche utilizzate per valutare il modello e quindi la precisione dei cluster formati.

2. Outliers

Prima di passare all'implementazione del modello ho voluto vedere tramite un grafico a scatola per valutare i valori outliers per Recency, Frequency e Total, questo viene fatto per non valutare valori eccessivamente sopra la media.

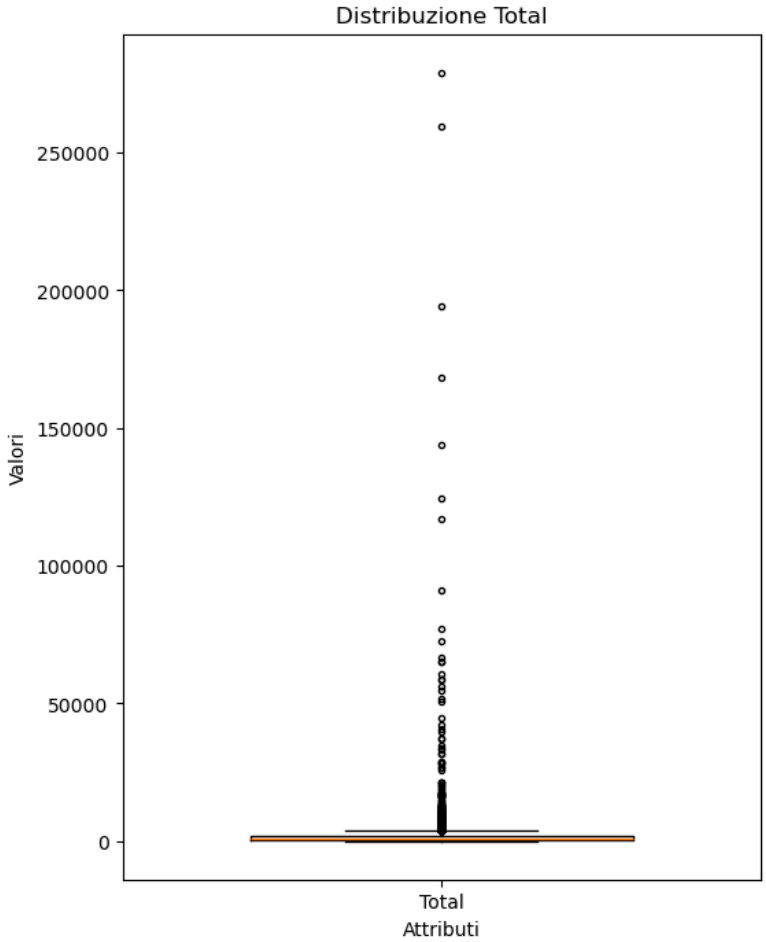

Figure 1: Distribuzione Total

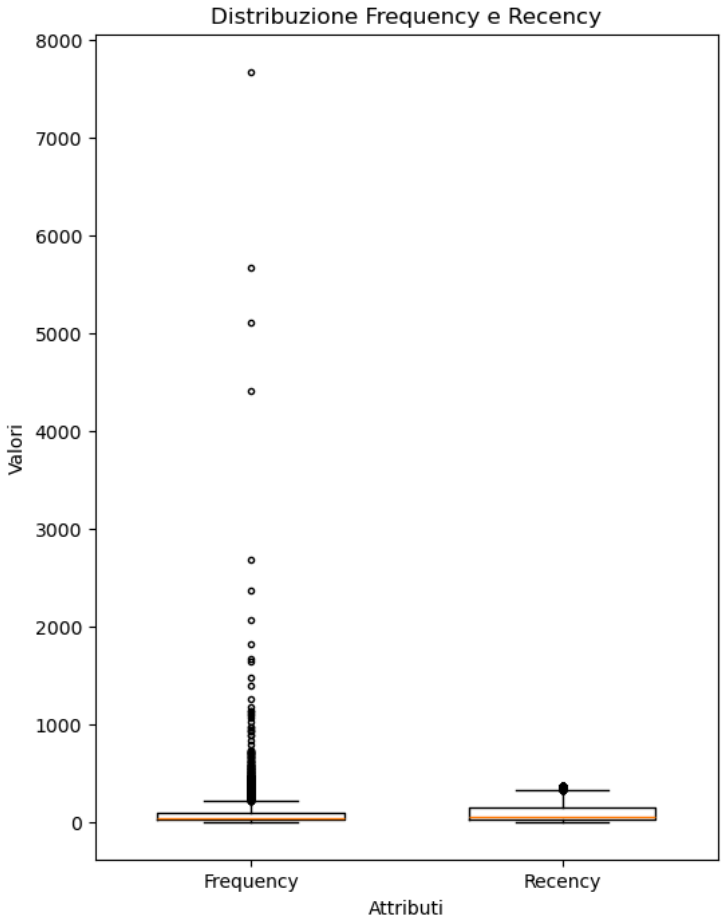

Figure 2: Distribuzione Frequency e Recency

Come si vede il grafico a box nemmeno si comprende in quanto molti valori superano l'estremo superiore o banalmente anche il 75° percentile della scatola.

Per questo motivo è stato eseguito il seguente codice:

```
Q1 = rfm.Total.quantile(0.05)
Q3 = rfm.Total.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Total >= Q1 - 1.5*IQR) & (rfm.Total <= Q3 + 1.5*IQR)]

Q1 = rfm.Recency.quantile(0.05)
Q3 = rfm.Recency.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Recency >= Q1 - 1.5*IQR) & (rfm.Recency <= Q3 + 1.5*IQR)]

Q1 = rfm.Frequency.quantile(0.05)
Q3 = rfm.Frequency.quantile(0.95)
IQR = Q3 - Q1
rfm = rfm[(rfm.Frequency >= Q1 - 1.5*IQR) & (rfm.Frequency <= Q3 + 1.5*IQR)]
```

Q1 e Q3 rappresentano rispettivamente il 5° precentile e il 95° percentile mentre IQR è l'intervallo interquartile calcolato come la differenza di Q3 - Q1. Successivamnete venivani quindi tolti i valori outliers prima da total poi da recency e infine da frequency.

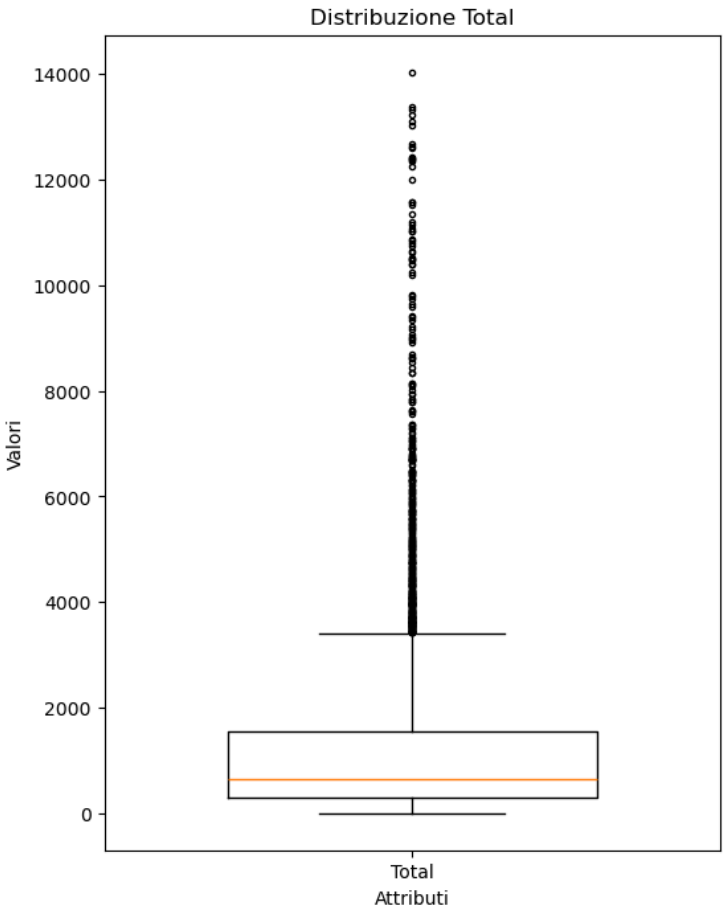

Figure 3: Distribuzione Total post eliminazione outliers

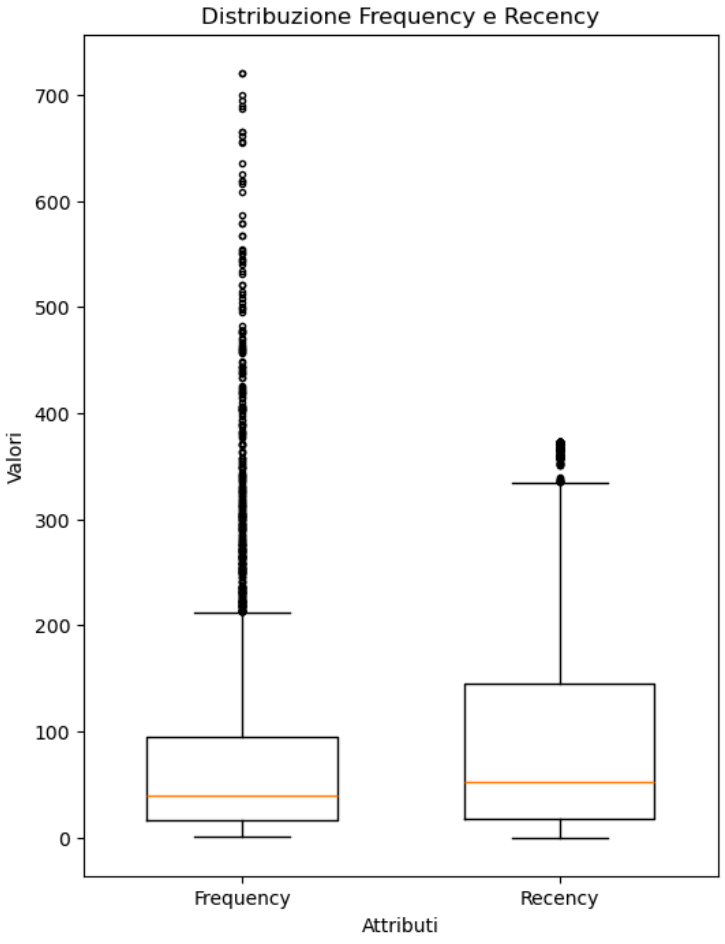

Figure 4: Distribuzione Frequency e Recency post eliminazione outliers

Come si può vedere una volta eliminati gli outliers i valori vengono molto più ridimensionati ed è anche possibile vedere il valore mediano delle varie distribuzioni.

3. Scaling dei valori

Lo scaling dei valori è anch'esso un passaggio fondamentale prima di dare in pasto al modello i dati. Questo viene fatto per avere dei valori simili tra di loro, facendo un esempio, Recency e Total sono su scale di valori totalmente diversi tra di loro, Total viaggia sulle decine di migliaia mentre Recency arriva alle centinaia.

Scalando i valori si ottiene una scala abbastanza simile per tutte le colonne, in questo caso è stato utilizzato `StandardScaler()` in sklearn il quale usa la media e la deviazione standard.

La formula è la seguente:

$$z = \frac{x - \mu}{\sigma}$$

dove:

- x = valore originale;
- μ = media dei valori per colonna;
- σ = deviazione standard per colonna;
- z = valore scalato.

Dovendo poi usare un algoritmo di KMeans che misurano le distanze tra i diversi valori, questi funzionano meglio e si ottengo valori migliori quando le features hanno scale simili.

	CustomerID	Monetary	Frequency	Recency
1	12347.0	1.689663	1.087371	-0.913418
2	12348.0	0.077064	-0.495713	-0.184819
3	12349.0	0.088464	-0.036108	-0.753725
4	12350.0	-0.564460	-0.608061	2.160669
5	12352.0	0.519850	0.045599	-0.574071

Figure 5: Dataframe con valori scalati.

4. Modello

Per il modello è stato scelto di usare l'algoritmo KMeans di sklearn.cluster in quanto un ottimo algoritmo per la segmentazione con la quantità di dati disponibili.

Prima di avere un modello finale bisogna capire bene quanti cluster è ottimale impostare come perametri dell'algoritmo, poichè una netta separazione delle distanze tra i diversi cluster risulta migliore.

Per affrontare questo problema ci vengono in aiuto due metodi (in realtà ce ne sono altri, ma ho deciso di utilizzare questi) l'elbow curve e lo silhouette score.

In entrambi i casi viene iterato il modello cambiando i parametri, ovvero facendo hyperparameter tuning per ottenere uno score adeguato al nostro problema.

4.1. Elbow curve

Per determinare un valore ottimale con il metodo della curva a gomito, bisogna valutare i valori della somma delle distanze al quadrato per i numero di cluster. Mostrando poi graficamente i risultati solitamente il cluster ottimale è quello in cui è presente appunto uno "spigolo a gomito".

Di seguito riporto il grafico dell'elbow curve con evidenziato tramite retta tratteggiata rossa il valore ottimale.

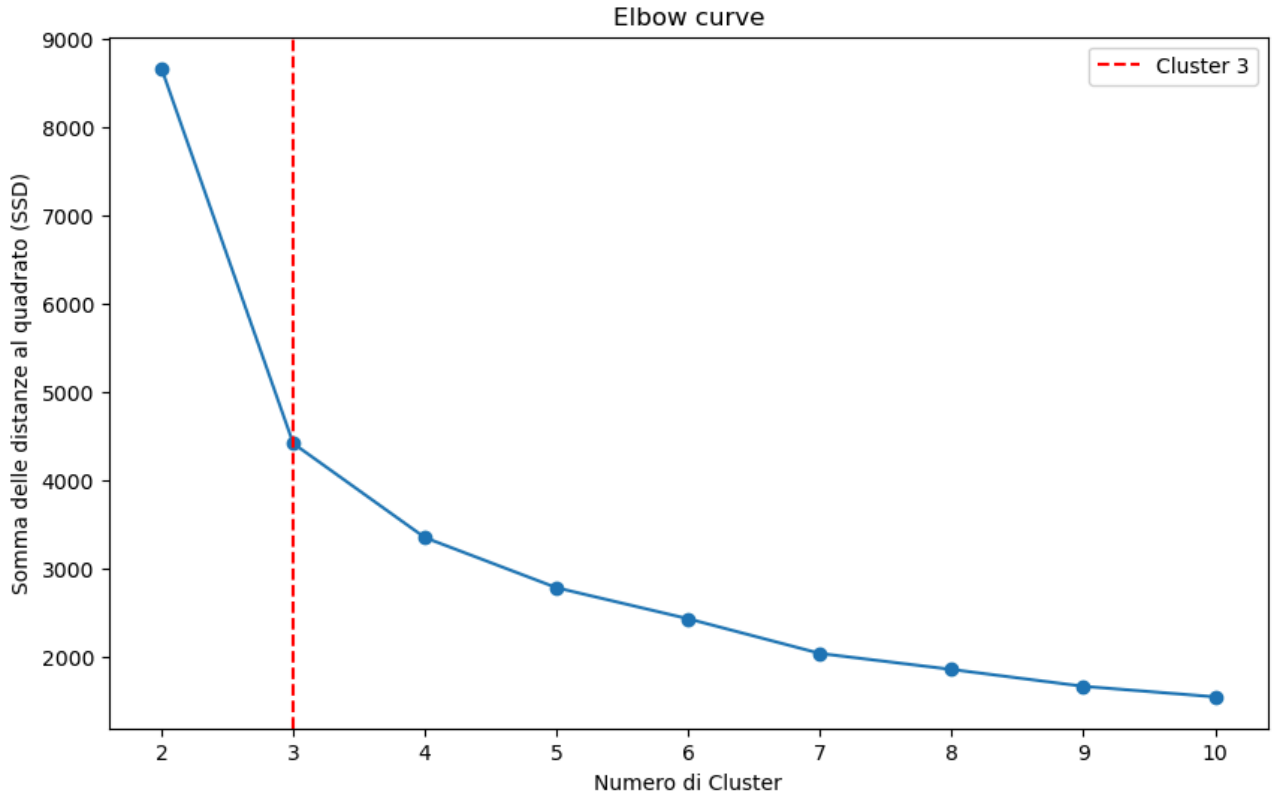

Figure 6: Elbow curve.

Non sempre è facile capire dove si trovi il valore ottimale poichè non è detto che la curva sia cosi visibile, in ogni caso ho deciso di confrontarlo con un silhouette score per confermare la scelta di tre cluster.

4.2. Shilhouette score

Il Shilhouette score invece misura la distanza media tra il punto x e tutti gli altri dello stesso cluster e la distanza media tra il punto x e i punti del cluster più vicino. Più vicino a 1 è il valore dello score più i cluster sono ben definiti e distinti.

Anche in questo caso tramite iterazione si è andato a modificare gli iperparametri dell'algoritmo andando a stampare i diversi punteggi per numero di cluster, ne sono poi usciti due grafici.

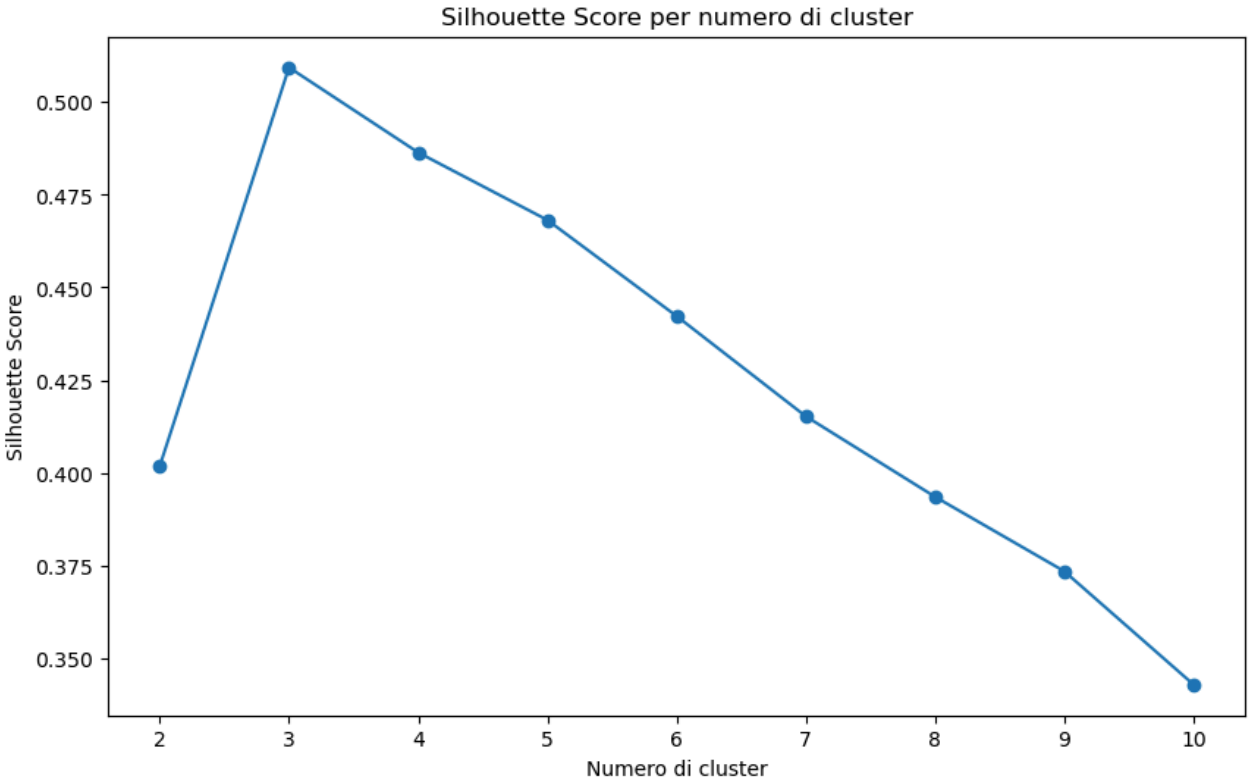

Figure 7: Shilhouette score

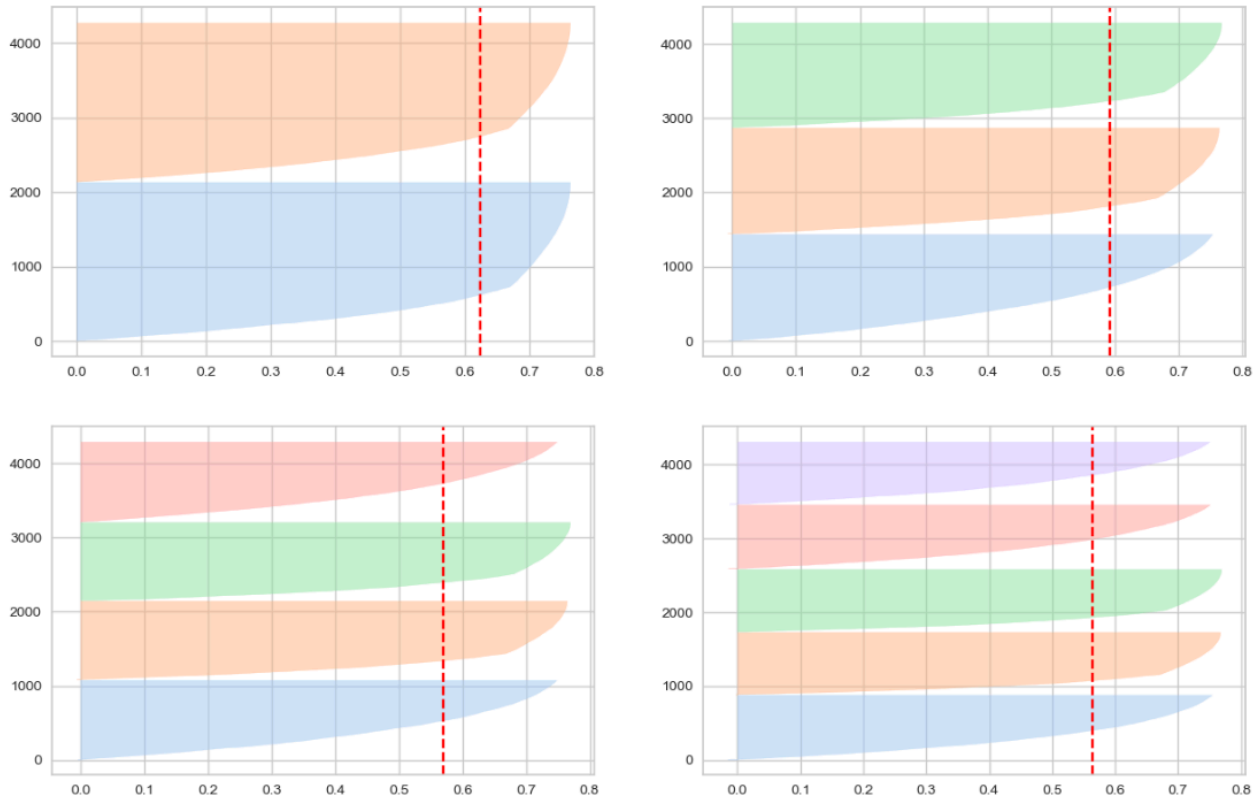

Figure 8: Shilhouette score per 2,3,4 e 5 cluster.

Con il punteggio di questa metrica si è confermato il numero di cluster a 3, procedendo quindi ha implementare i il modello finale.

4.3. Modello finale

Una volta implementato il modello finale è stato aggiunto al dataframe una nuova colonna che assegnasse ad ogni utente il suo cluster di appartenenza.

Risulato quindi esserci 2713 clienti assegnati al cluster 0, 1053 clienti assegnati al cluster 1 e 491 clienti asseganti al cluster 2.

A modello concluso si sono inoltre volute ottenere delle metriche per valutarne l'efficenza ottenendo:

- Silhouette Score: 0.5771029681348092, indica che, complessivamente, i punti sono ben raggruppati nei loro cluster e i cluster sono ragionevolmente separati;
- Calinski Harabasz Score: 5008.770628595688 indica che, la separazione tra i cluster è buona e che i punti all'interno dei cluster sono abbastanza compatti;
- Davies Bouldin Score: 0.6373852191635594 indica che, i cluster sono distinti e ben separati l'uno dall'altro.

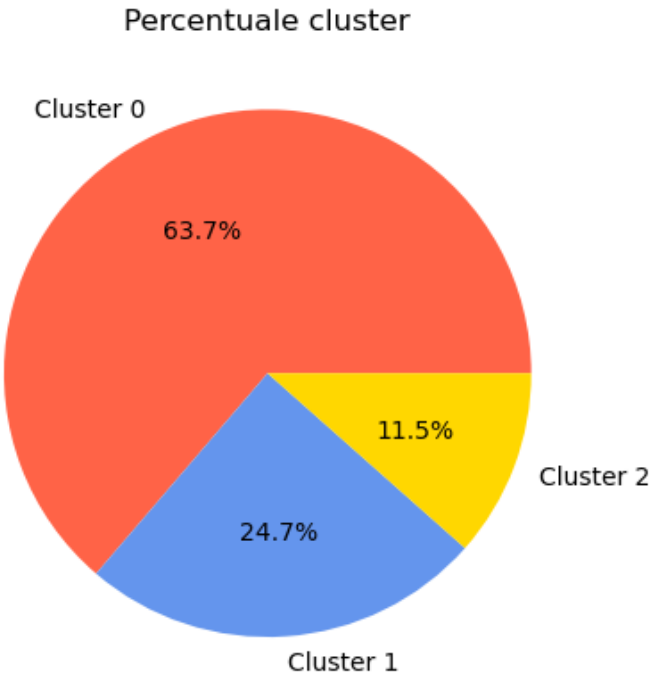

Figure 9: Suddivisione dei cluster.