Michael Cavicchioli
Sofia Dami
Claudia Landi
Anno Accademico: 2023-2024

Progetto Data Mining - Customer Segmentation con Clustering e Classificazione

1 Introduzione al caso di studio

La customer segmentation è una strategia di marketing che consiste nel dividere un insieme di clienti in gruppi di individui con caratteristiche comuni, come età, sesso, interessi e abitudini di spesa. La segmentazione consente alle aziende di creare campagne di marketing mirate, adattando messaggi e offerte specifiche per ogni gruppo di clienti. Questo permette di aumentarne l'efficacia e di migliorare il tasso di risposta e la soddisfazione del cliente, andando ad aumentare le vendite e la fedeltà degli acquirenti.
Nel presente caso di studio è stato preso in analisi un dataset contenente alcune informazioni su un gruppo di 200 clienti, con l'obiettivo di eseguire la customer segmentation. Per raggiungere questo obiettivo, sono stati applicati due algoritmi di clustering, K-Means e DBScan, al fine di identificare gruppi distinti di clienti basati su caratteristiche comuni. Successivamente si è scelto di addestrare modelli di alberi decisionali, in particolare Decision Tree, al fine di predire diverse categorie di spending score. Il lavoro è stato dunque suddiviso in:

- una fase di analisi del dataset e preprocessing;
- una di applicazione degli algoritmi di clustering con la ricerca dei migliori parametri;

- una fase di addestramento e valutazione del modello Decision Tree, al fine di predire una suddivisione in due, tre o quattro classi di spending score;
- una fase di analisi dei risultati ottenuti.

Nella presente relazione si andranno quindi a descrivere in modo più approfondito i passaggi sopra descritti.

2 Data understandig e preprocessing

In una prima fase si è reso necessario comprendere la struttura del dataset, a tale fine è stato utilizzato il software Weka, che permette di visualizzare informazioni sulle variabili, la loro composizione e le relazioni che intercorrono tra di esse.
Il dataset considerato è composto da 200 osservazioni, ciascuna delle quali corrisponde ad un cliente, a cui sono associate le seguenti variabili:

- Customer ID, ovvero un identificativo univoco;
- Genre, che indica il sesso (si hanno 88 maschi e 112 femmine);
- Age, che indica l'età dei clienti e va da 18 a 70 anni;
- Annual Income, un numero che indica il reddito annuale in termini di migliaia di dollari (da un minimo di 15000 ad un massimo di 137000);
- Spending Score, un valore nel range da 1 a 100 che corrisponde ad un punteggio assegnato in base alle abitudini di spesa.

Visualizzando dei plot che mettano in relazione tra loro le variabili, si possono già notare alcune relazioni interessanti. Ad esempio, se si visualizza la distribuzione dei clienti in base ad età e spending score, è possibile vedere che la maggioranza delle persone con meno di 40 anni tende ad avere uno spending score superiore a 50, mentre sopra ai 40 anni è distribuito su valori più bassi (figura 1).

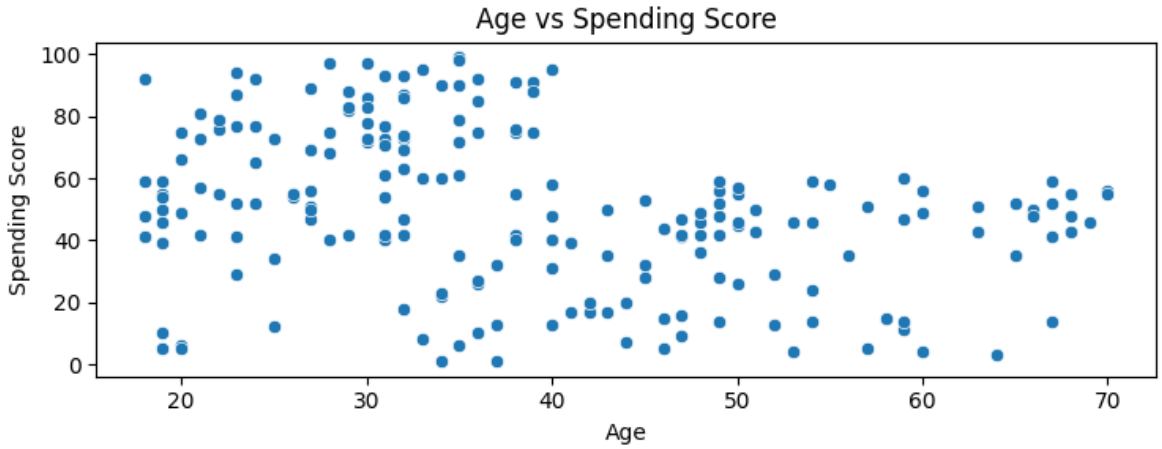

Figura 1: Distribuzione dei clienti in base ad età e spending score.

Un altro interessante grafico è quello che mostra la relazione tra lo stipendio annuale e lo spending score. Si può notare che i clienti tendono a raggrupparsi in cinque cluster piuttosto evidenti: uno con spending score tra 0 e 40 e stipendio inferiore a 50, uno con lo stesso range di spending score ma stipendi al di sopra di 70, uno con spending score tra 40 e 60 e stipendi compresi tra 40 e 80, uno con spending score al di sopra di 60 e stipendi inferiori a 50 e infine uno con lo stessso range di spending score ma stipendi superiori a 80 (figura 2).

Figura 2: Distribuzione dei clienti in base a stipendio annuale e spending score.

Una volta compresa la struttura del dataset si è reso necessario prepararlo al fine di applicare gli algoritmi di machine learning desiderati. Per le operazioni di preprocessing e addestramento dei modelli è stato utilizzato un ambiente Python.
Le funzioni per la gestione del dataset sono state implementate nella classe

MallCustomersDatasetController: come prima cosa è stato caricato il file *.csv* nell'ambiente di lavoro, dopodiché, tramite il metodo *print_column_info()* si sono visualizzate informazioni sui tipi delle variabili e sulla presenza di eventuali valori nulli. Per eventuali righe con valori nulli è stata predisposta l'eliminazione, ma queste non sono risultate presenti nel dataset. Infine è stato effettuato l'encoding della variabile categorica "Genre", facendo uso della funzione di *label_encoding()* del pacchetto *Scikit-Learn*.

3 Algoritmi di clustering

Terminata la fase di preprocessing, è stato possibile andare ad applicare diversi algoritmi di clustering al fine di selezionare i migliori risultati ed ottenere informazione sulla segmentazione dei clienti.

3.1 K-Means

Il primo modello che si è scelto di applicare è K-Means, una tecnica di clustering che suddivide un insieme di dati nel numero di gruppi specificato dal parametro k. Durante l'esecuzione dell'algoritmo vengono scelti casualmente k punti dal dataset, detti centroidi iniziali, che rappresenteranno i centri di partenza per i cluster, dopoidiché ogni punto del dataset viene assegnato al centroide più vicino, formando k cluster. La vicinanza viene solitamente misurata utilizzando la metrica della distanza euclidea. Per ogni cluster viene poi calcolato il nuovo centroide come il punto medio di tutti i punti assegnati a quel cluster. I passaggi di assegnazione e calcolo dei centroidi vengono ripetuti: i punti vengono riassegnati ai nuovi centroidi calcolati e si calcolano nuovi centroidi per i cluster aggiornati. L'algoritmo continua termina quando i centroidi non cambiano più in modo significativo e quindi i cluster hanno raggiunto una stabilità.
Per utilizzare K-Means nel caso di studio, come prima cosa è stato necessario definire il numero di cluster ottimale. Per determinarlo è stata implementata la funzione *best_kmeans()* che esegue l'algoritmo con diversi valori di k e sceglie il migliore in base al miglior SSE e silhouette score. Il numero ottimale di cluster determinato dal metodo è 6, sia in base all'elbow method (figura 3) che al grafico del silhouette score ottenuto (figura 4).

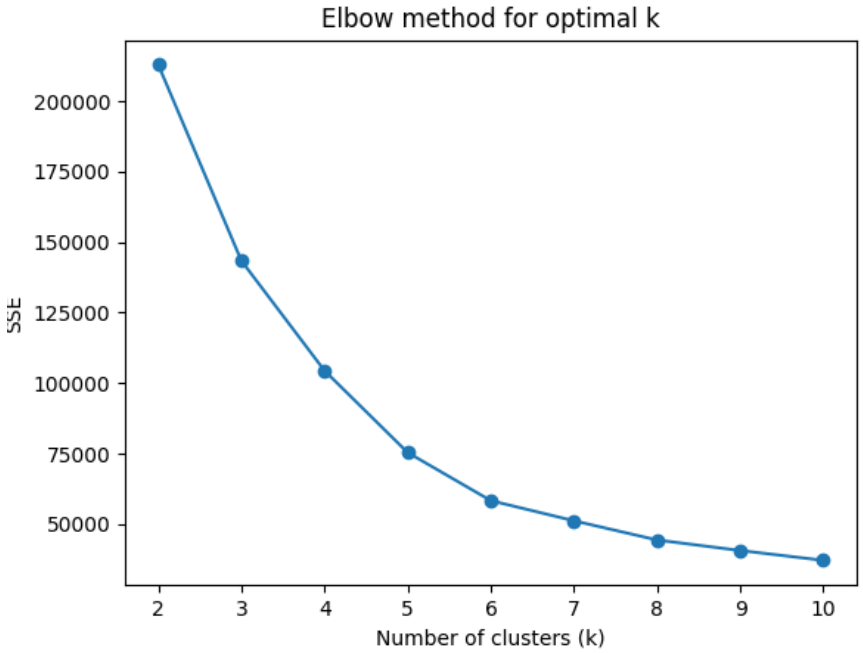

Figura 3: Grafico che mostra l'andamento dell'SSE al variare del numero di cluster di K-Means.

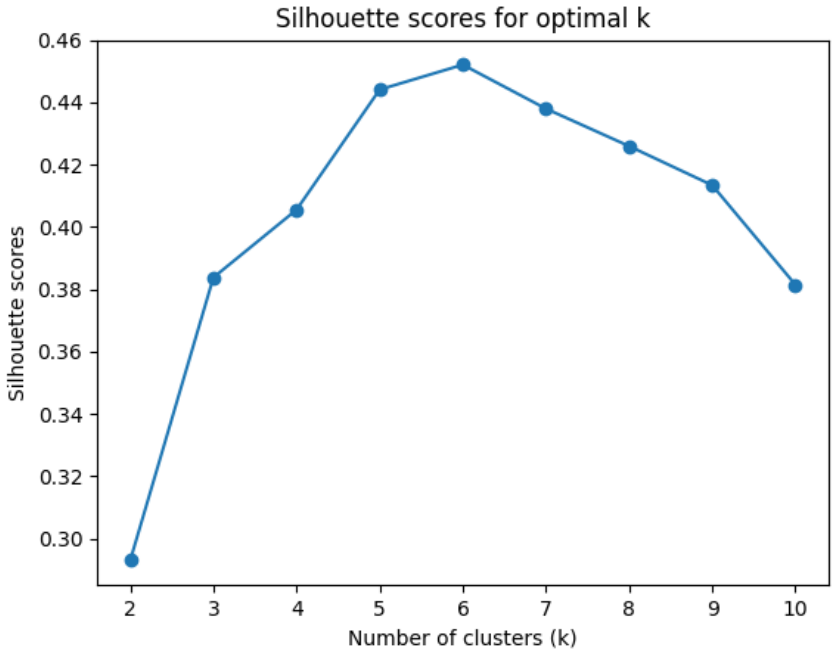

Figura 4: Grafico che mostra l'andamento del silhouette score al variare del numero di cluster di K-Means.

Dato che la scelta dei centroidi nel K-Means è casuale, si potrebbero ottenere risultati diversi a seconda di come questi vengono selezionati, pertanto è stato implementato un metodo che ripete più volte l'algoritmo

con diversi centroidi iniziali e che sceglie quelli che permettono di ottenere il miglior SSE score.
In figura 5 si mostrano i risultati del clustering ottimale selezionato, quindi con 6 cluster e i migliori centroidi iniziali.

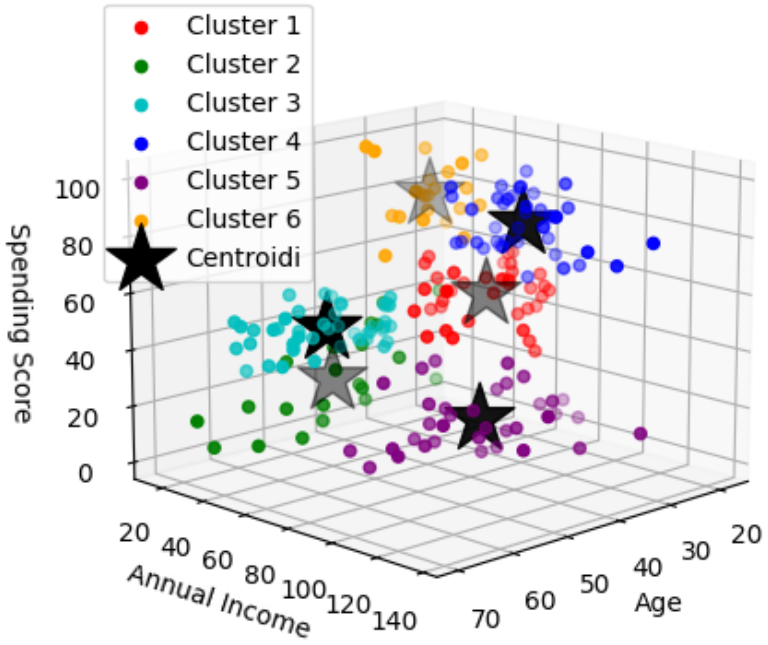

Figura 5: Cluster risultanti dall'applicazione dell'algoritmo K-Means. Nell'immagine il plot è eseguito solo rispetto a tre variabili, ma per la selezione dei cluster sono state considerate tutte.

3.2 DBScan

DBScan è un algoritmo di clustering in grado di identificare gruppi di punti basandosi sulla loro densità, individua quindi i cluster che si presentano come regioni più dense, separate da regioni con una minore densità, che costituiranno i punti di rumore. Utilizza due parametri:

- Epsilon, che indica il raggio entro il quale cercare i vicini di un punto,
- MinPts, che indica il numero minimo di punti per identificare un punto come core.

Tutti i punti del dataset sono quindi suddivisi in:

- punti di core: se il numero di vicini all'interno del raggio epsilon è almeno MinPts,
- punti di border o di rumore: se il numero di vicini è inferiore a MinPts, ma il punto si trova in prossimità di un punto di core, allora è classificato come di border, se non è né di border né di core allora è rumore.

A partire dal core point, l'algoritmo espande i cluster includendo tutti i punti densamente connessi. I core points vengono inseriti in un cluster insieme ai propri vicini e i punti border sono assegnati al core point più vicino. Alla fine dell'algoritmo tutti i punti saranno classificati come rumore oppure all'interno di un cluster.
Per applicare l'algoritmo al dataset in analisi, è stato necessario ricercare i migliori parametri epsilon e MinPts (eps e min_samples in Python). Come prima cosa è stato lanciato più volte l'algoritmo con diversi valori e sono stati selezionati quelli che permettevano di ottenere il miglior silhouette score, questo ha permesso di determinare il parametro min_pts e di stimare un eps. Il numero di min_pts migliore è risultato essere 3. A questo punto è stato utilizzato l'algoritmo K-Nearest-Neighbor per determinare la distanza di ogni punto dai suoi 3 vicini più prossimi e le distanze sono state ordinate e visualizzate in un grafico. Questo ha permesso di determinare il miglior valore di epsilon, andando a scegliere quello che corrisponde ad una crescita più rapida delle distanze nel grafico, ovvero $eps = 14$ (figura 6).

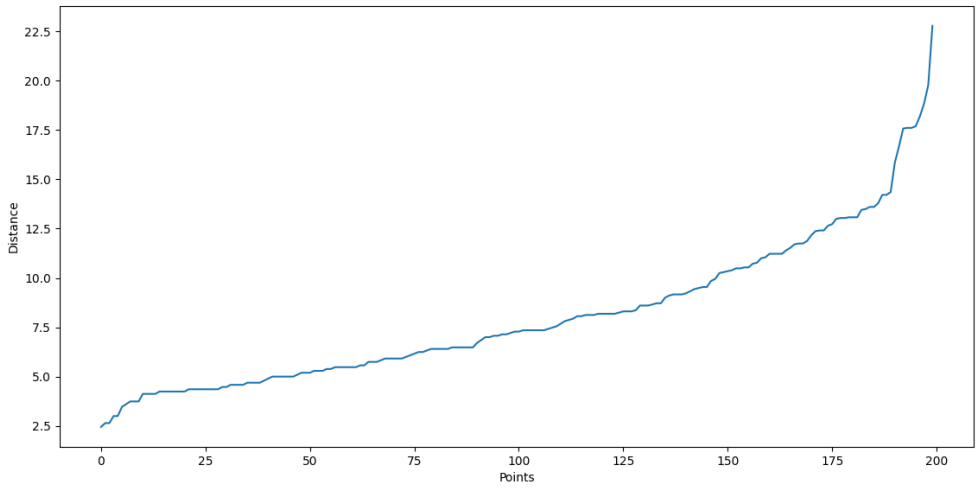

Figura 6: Plot delle distanze di ogni punto dai suoi vicini più prossimi. Si vede che il grafico sale più velocemente intorno alla distanza 14.

Andando ad applicare l'algoritmo DBScan con i valori trovati, si ottiene il risultato mostrato in figura 7.

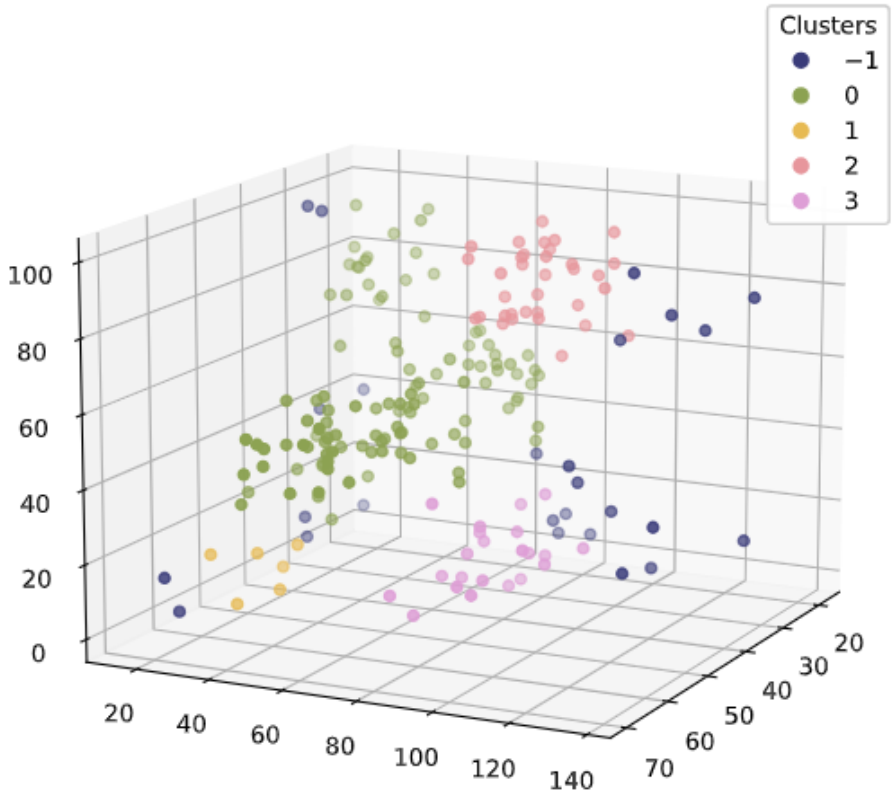

Figura 7: Cluster ottenuti dall'algoritmo DBScan.

Si osserva che i risultati restituiti da questo algoritmo risultano leggermente meno soddisfacenti, questo può essere dovuto al fatto che è presente una certa variabilità di densità nel dataset, per cui alcune istanze vengono riconosciute come outliers e i gruppi risultano meno definiti che nel caso del K-Means.

3.3 Clustering agglomerativo

Il clustering agglomerativo è una tipologia di clustering gerarchico, che quindi non richiede di specificare in anticipo il numero di gruppi che si vogliono ottenere, ma raggruppa insieme dati simili secondo una gerarchia. Ogni istanza dei dati è inizialmente considerata come un proprio cluster, dopodiché i vari cluster vengono fusi in modo iterativo sulla base della loro somiglianza fino a quando tutte le istanze appartengono a un unico cluster. La fusione dei cluster simili avviene calcolando le distanze tra tutti i possibili cluster, il cui calcolo dipende dal tipo di algoritmo usato:

- Con il *single link* la distanza tra i cluster è definita come la distanza minima tra qualsiasi coppia di punti, dove ogni punto è preso da uno dei due cluster;
- Nel *complete linkage* la distanza tra due cluster è data dalla massima distanza tra gli elementi dei cluster;
- Nel *group average* la distanza è data dalla distanza media tra le coppie di punti dei due cluster.

Nel caso di studio è stato applicato ciascuno di questi algoritmi e i risultati sono stati visualizzati tramite alcuni dendrogrammi, che mostrano come i punti del dataset sono raggruppati in cluster e come quest'ultimi si uniscono a diversi livelli di similarità (figure 8, 9, 10).

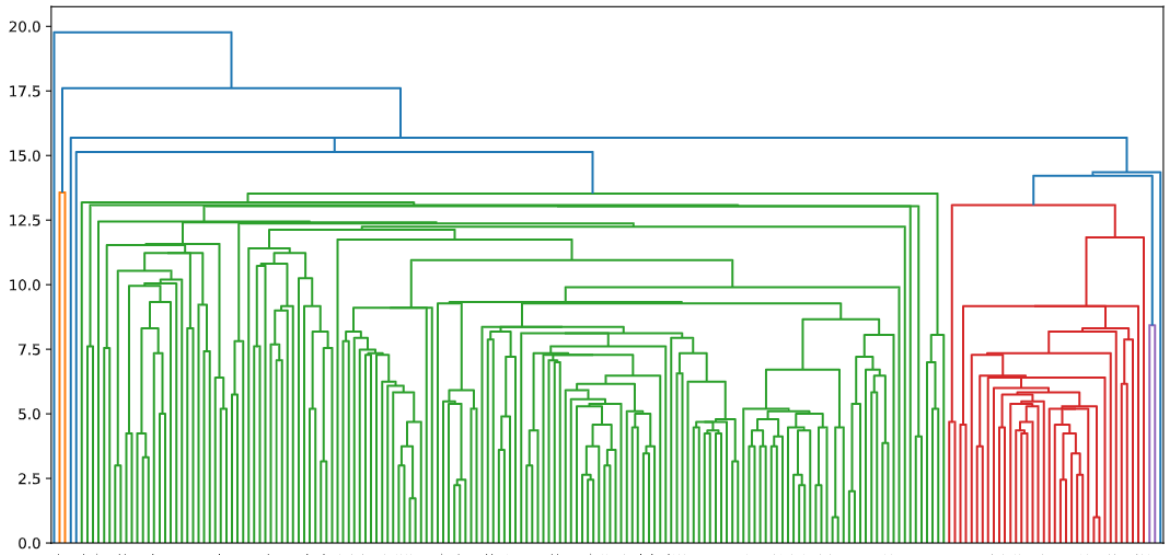

Figura 8: Dendrogramma ottenuto dall'algoritmo single link.

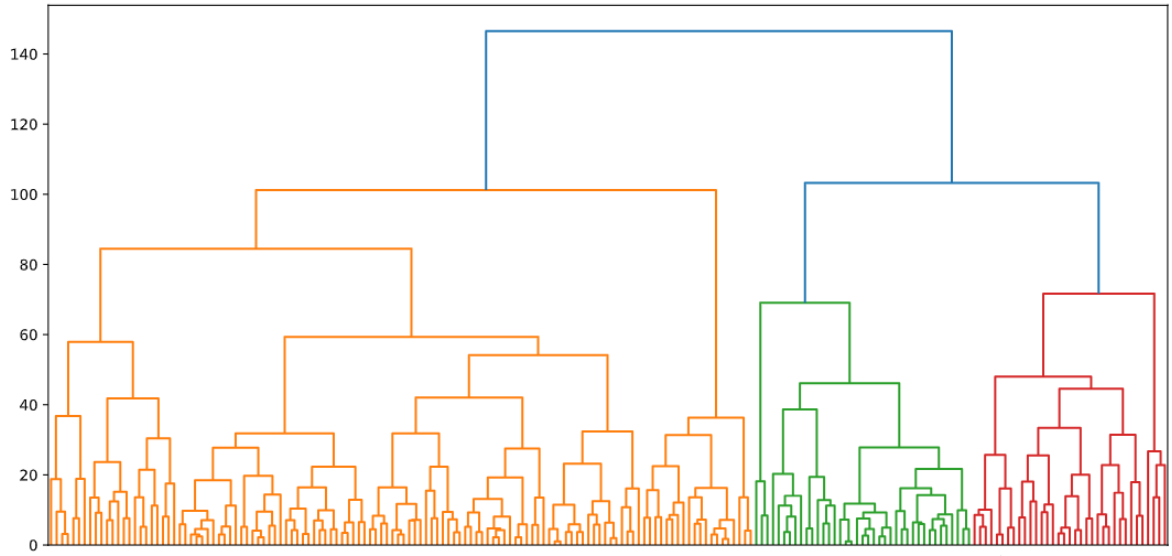

Figura 9: Dendrogramma ottenuto dall'algoritmo complete linkage.

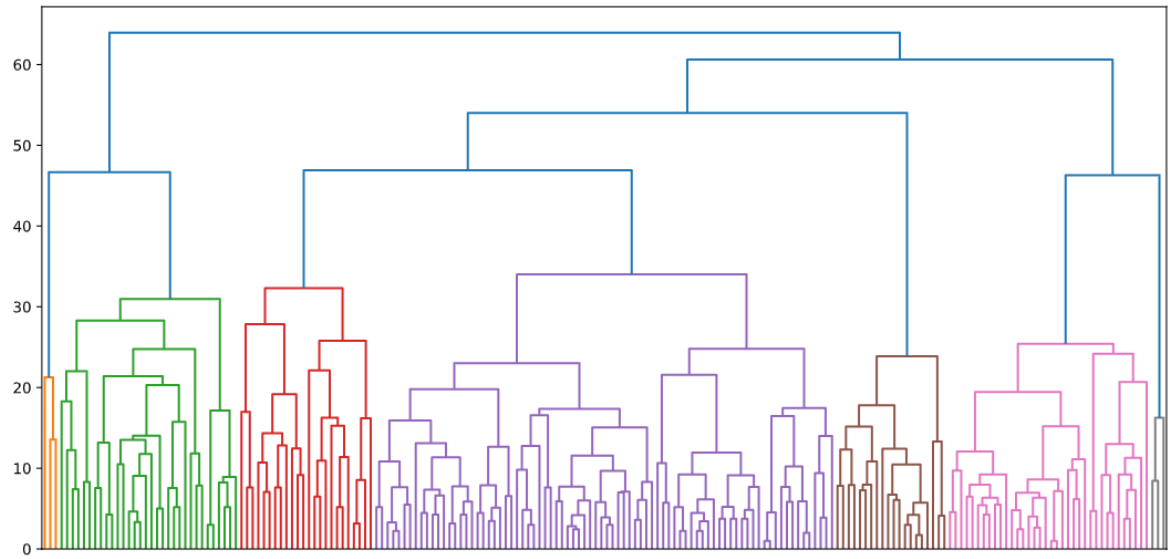

Figura 10: Dendrogramma ottenuto dall'algoritmo group average.

Una volta ottenuti i dendrogrammi, è stato scelto per ciascuno il taglio che permetteva di ottenere 6 cluster, così da poter confrontare i risultati con quelli di K-Means e per ciascuna tipologia di clustering sono stati salvati i risultati in file csv. Infine si è scelto di confrontare tra loro le tre metodologie utilizzate attraverso la metrica del silhouette score, il quale misura quanto i punti sono ben raggruppati all'interno del proprio cluster. La strategia migliore in base al silhouette score è quindi risultato essere il complete linkage, seguita dal group average e per ultimo il single link.

3.4 Analisi dei cluster ottenuti

Avendo utilizzato diverse strategie di clustering si è scelto di andare a confrontare i risultati più interessanti, in particolare quelli dell'algoritmo K-Means e del clustering agglomerativo con complete linkage.
Per visualizzare le informazioni di ciascun cluster ottenuto con K-Means, sono stati salvati dei file csv distinti per ognuno, che sono accessibili nella cartella *dataset_cluster* una volta eseguito il codice. Inoltre sono state create anche delle tabelle che per ciascun gruppo mostrano la media, il minimo e il massimo di ciascuna feature in ogni cluster.
Come prima cosa si osserva che il sesso non è una variabile discriminante nel formare i cluster, infatti, sebbene maschi e femmine non siano sempre nello stesso numero, non si evidenziano particolari disparità tra i due sessi nei raggruppamenti. Di seguito si riportano le caratteristiche osservate in ciascuno dei cluster selezionati:

- **Cluster 0**: comprende persone dai 43 ai 60 anni con uno stipendio in fascia media che risultano avere anche uno spending score in fascia media, ovvero tra 35 e 60.
- **Cluster 1**: è composto da persone in una fascia di età ampia, dai 19 ai 59 anni, quindi in questo caso l'età non sembra un fattore discriminante per il raggruppamento. La caratteristica curiosa di questo cluster è che, sebbene gli stipendi siano alti e al di sopra della media, lo spending score risulta basso, ovvero tra 1 e 39 con una media di 17.
- **Cluster 2**: composto da individui di età inferiore ai 40 anni, con un reddito intorno alla media che presentano anche uno spending score in una fascia media (con un minimo di 29, un massimo di 61 e una media di 49).
- **Cluster 3**: questo gruppo è composto da persone di età inferiore ai 40 anni, con un reddito alto e uno spending score alto, con una media di 82.
- **Cluster 4**: è costituito da giovani di età media 25 anni, con uno stipendio basso, che però presentano uno spending score medio-alto, che quindi in questo caso non sembra tener conto del reddito.
- **Cluster 5**: anche in questo gruppo rientrano individui con una fascia di età ampia, che presentano uno stipendio basso che si riflette anche su un basso spending score.

Per quanto riguarda il clustering agglomerativo, si è scelto di analizzare i cluster ottenuti dalla migliore strategia, ovvero il complete linkage. Anche in questo caso è possibile osservare che la variabile sesso non è discriminante nella distinzione dei gruppi; di seguito si riportano le altre caratteristiche di ognuno dei cluster:

- **Cluster 0**: comprende persone in una fascia di età ampia, che presentano redditi bassi e uno spending score medio-basso.
- **Cluster 1**: composto da persone in una fascia di età ampia, con stipendi e spending score intorno a valori intermedi.
- **Cluster 2**: include persone giovani, tra i 18 e i 35 anni, con un reddito annuo basso, ma con un alto spending score.
- **Cluster 3**: comprende persone dai 27 ai 42 anni, con sia stipendio che spending score alto.
- **Cluster 4**: le persone in questo gruppo hanno un'età media di 42 anni, presentano stipendi sopra la media, ma spending score bassi.
- **Cluster 5**: persone tra 32 e 47 anni, con un alto reddito ma spending score molto al di sotto della media.

Questa analisi permette di osservare che, sebbene i cluster restituiti dai due algoritmi non siano esattamente gli stessi, emergono alcune interessanti informazioni, come il fatto che il sesso non ha influenza sul raggruppamento e anche il fatto che non sempre a bassi redditi siano associati spending score bassi o viceversa. In particolari le persone giovani sembrano avere una predisposizione ad uno spending score alto, nonostante gli stipendi al di sotto della media, ed emerge anche un insieme di persone che, nonostante gli alti redditi, presentano uno spending score molto basso.

4 Classificazione

Una volta definite le caratteristiche dei cluster, si è ritenuto opportuno addestrare un modello di classificazione in grado di predire lo spending score dei clienti. L'algoritmo scelto per la classificazione è il Decision Tree, un modello di machine learning utilizzato per prendere decisioni basate su una serie di regole derivate dai dati. A ogni livello dell'albero, viene selezionata la feature che meglio divide il set di dati in sottogruppi, basandosi su metriche

come l'indice Gini o l'entropia. Una volta selezionata la caratteristica, i dati vengono divisi in sottogruppi in base ai valori di quella feature. Questo processo continua ricorsivamente per ogni sottoinsieme di dati fino a quando non viene raggiunto un criterio di arresto.
Per applicare l'algoritmo Decision Tree sul dataset in analisi si è scelto di dividere lo spending score in un diverso numero di classi, per poi valutare su ciascuna suddivisione le prestazioni dell'algoritmo:

- Due classi: spending score minore o uguale a 50 e maggiore di 50;
- Tre classi, una per ciascuno dei seguenti intervalli di spending score: [0, 33], [34, 66], [67, 100];
- Quattro classi, una per ciascuno dei seguenti intervalli di spending score: [0, 25], [26, 50], [51, 75], [76, 100].

Per ciascuna delle suddivisioni sopra descritte è stato addestrato un modello di Decision Tree. Al fine di definire i migliori iperparametri per il modello è stata inoltre utilizzata la procedura della grid search, la quale permette di specificare diversi valori per gli iperparametri, testare automaticamente ogni combinazione e selezionare quella che consente di ottenere i migliori risultati in termini di cross validation accuracy. I parametri da testare sono contenuti nella variabile *param_grid*, in particolare sono stati confrontati i due criteri "gini" ed "entropy" e diversi valori di profondità dell'albero, ovvero 3, 5, 7, 10 e "None" (l'ultimo termina lo splitting quando ogni nodo è puro).
Il miglior modello di previsione è risultato essere quello che suddivide lo spending score in 3 classi, con migliori iperparametri l'utilizzo dell'entropia e profondità massima 3 (figura 11). La cross validation accuracy ottenuta con tale modello è 0.79, inoltre è stata visualizzata anche la matrice di confusione ottenuta dal modello, utilizzando l'80% dei dati per il training e il 20% per il testing (figura 12).

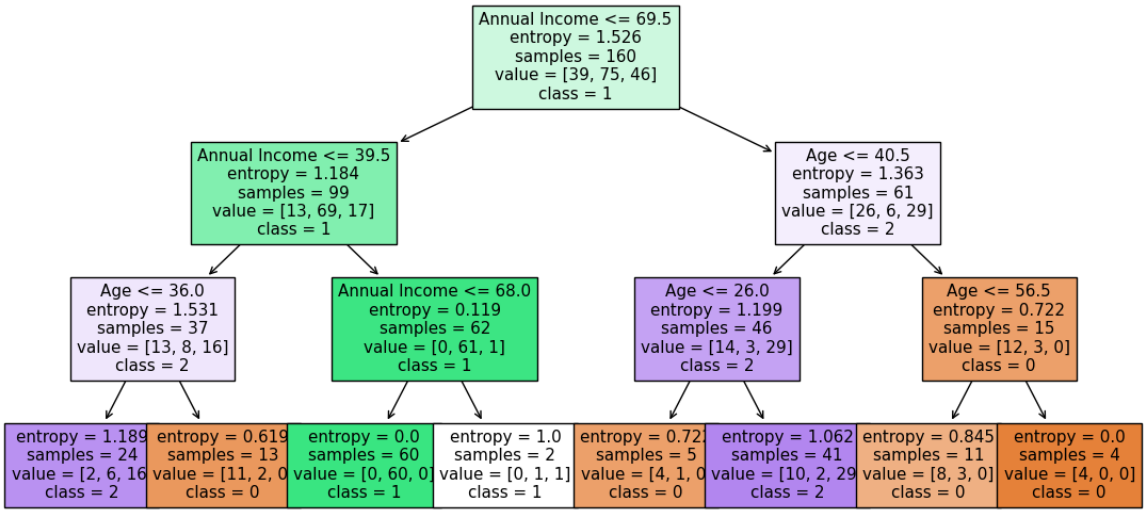

Figura 11: Decision Tree ottenuto con 3 classi di spending score, entropia e massima profondità = 3.

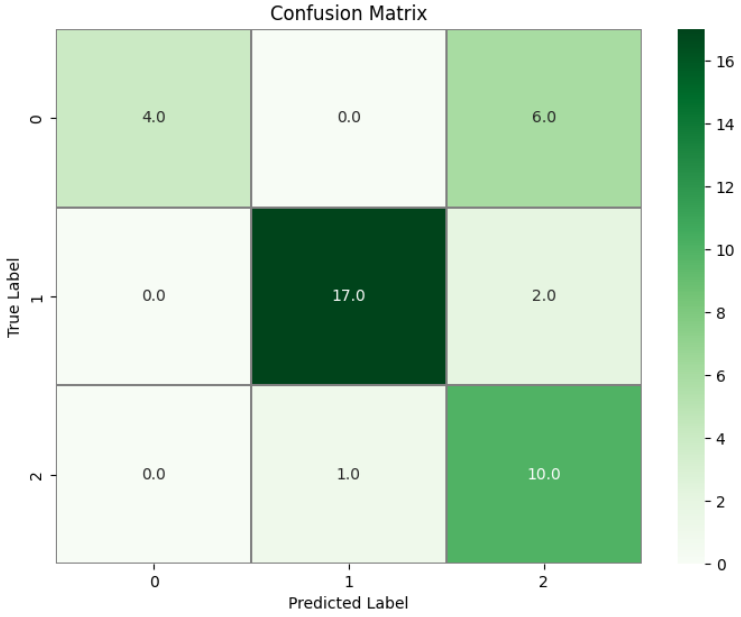

Figura 12: Confusion matrix del modello Decision Tree addestrato con 80% dati di training e 20% dati di test.

Dalla matrice di confusione è possibile osservare che le classi 1 e 2 sono predette correttamente la maggior parte delle volte, mentre la classe 0 viene classificata più spesso come 2 piuttosto che nel modo corretto. In generale anche i risultati di accuracy sono buoni, ma potrebbero essere migliorati.

5 Conclusioni

Il lavoro svolto aveva come obiettivo individuare categorie di clienti sulla base delle informazioni fornite, ovvero età, sesso, stipendio annuale e spending score. Per tale scopo sono stati impiegati diversi algoritmi di clustering, tra cui si è scelto di approfondire i risultati di K-Means e del clustering agglomerativo con strategia group average. I raggruppamenti ottenuti dai due algoritmi non sono risultati gli stessi, ma le caratteristiche osservate sono risultate simili. Sono emersi sei principali cluster, ciascuno indipendente dal sesso, ma con caratteristiche variabili per quanto riguarda le altre features. È stato interessante osservare come in alcuni casi lo stipendio non si rifletta sullo spending score, come nel caso di un gruppo di persone giovani che presentano uno stipendio annuo basso, ma uno spending score al di sopra della media, o il caso di un gruppo di individui che sebbene abbiano uno stipendio annuo al di sopra della media, hanno uno spending score piuttosto basso. In altri cluster invece è stata notata una proporzionalità tra stipendio e spending score.
Una volta analizzate le caratteristiche dei cluster si è scelto di impiegare dei modelli di Decision Tree al fine di cercare di predire lo spending score in base alle altre informazioni fornite. Sono state esplorate diverse combinazioni di parametri e diverse suddivisioni in classi basate sulla variabile target. Il modello migliore ottenuto è risultato essere quello addestrato per distinguere tre diverse categorie di spending score, utilizzando il Decision Tree con 'entropy' e profondità massima pari a 3. Il modello è stato in grado di ottenere un'accuracy di 0.79, anche se è risultato meno preciso nella classificazione della classe di spending score tra 0 e 33.
Possibili approcci futuri potrebbero prevedere l'impiego di altri modelli di machine learning per classificazione, al fine di confrontare i risultati e selezionare il modello più adatto. Potrebbe inoltre essere utile aumentare la dimensione del dataset con un maggior numero di osservazioni, così da ottenere risultati più precisi.