Data Mining Project
A. A. 2020/21

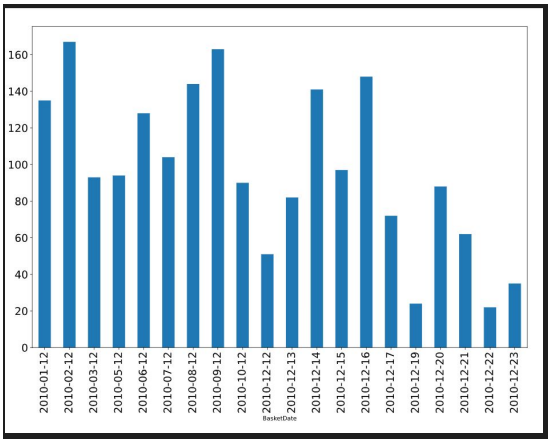
Matteo D’Onofrio - Giacomo Mariani

Data Semantics

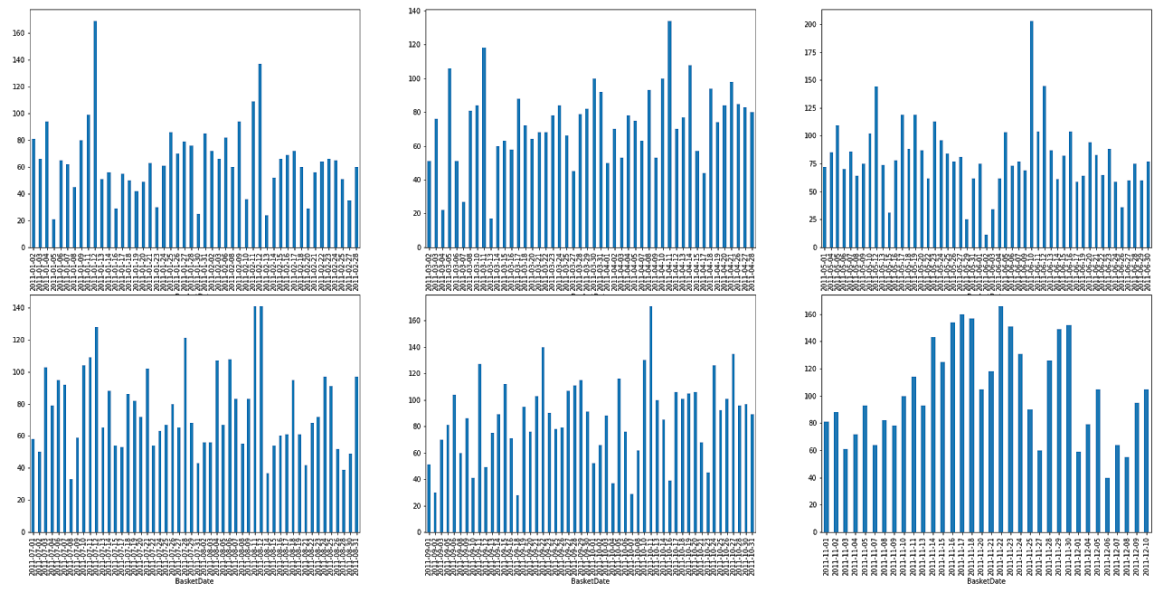
Nel dataset si descrivono le sessioni di acquisto di un certo numero di clienti. Una sessione di acquisto, detta basket (`BasketID`) e si svolge in una data (`BasketDate`), e viene effettuata da un cliente (`CustomerID`) residente in uno stato (`CustomerCountry`).
Ogni sessione rappresenta una serie di prodotti acquistati da un cliente, e in un record è possibile ritrovare, per il singolo prodotto, l'identificativo (`ProdID`), la descrizione (`ProdDescr`), e la quantità (`Qta`). Infine viene riportato il costo unitario del prodotto (`Sale`).

Distribution of variables and statistics

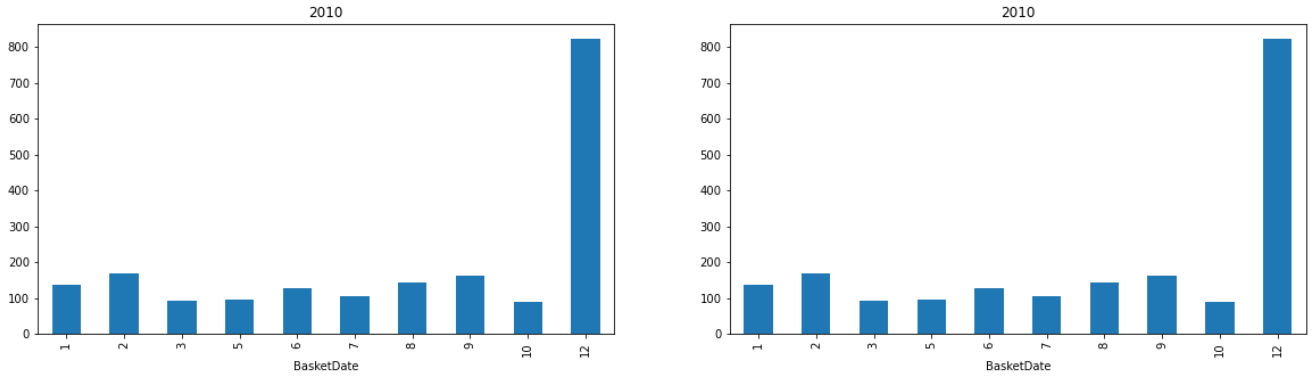
Il dataset contiene le informazioni su 24627 diversi basket, che modellano sessioni di acquisto eseguite in 22428 distinte date, concernenti un piccolo numero di record (34393) registrati in 1732 diverse date del 2010 e un numero più consistente (437517) risalenti a 20696 date del 2011.
Per quanto riguarda il 2010, il numero di baskets è distribuito nel seguente modo:



Per il 2011, per rendere i plot più leggibili, si è scelto di dividere il plot in slot di 2 mesi:

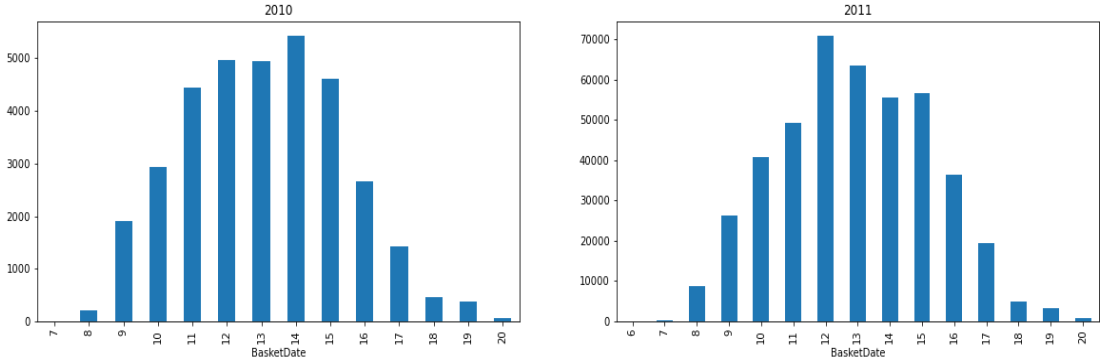


Dato che è difficile estrarre informazioni da questi grafici, si allega anche il numero di `BasketID` divisi per mesi:



In entrambi i casi non è possibile ipotizzare una distribuzione del numero di accessi rispetto alle date, ma si osserva un incremento del numero di accessi nei mesi di novembre e dicembre, a ridosso delle feste di Natale, così come ipotizzato dal senso comune.

Per quanto riguarda le ore di maggiore frequentazione, i plot per il 2010 e il 2011 sono i seguenti:



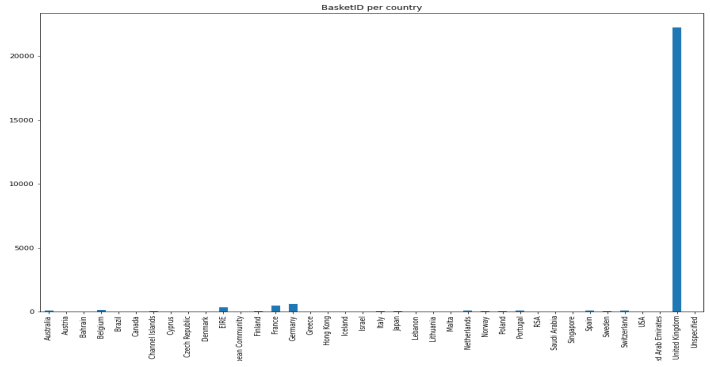
La distribuzione è riconducibile ad una gaussiana centrata nelle ore del primo pomeriggio.

I prodotti considerati sono 3953, associati a 4097 descrizioni (alcuni prodotti hanno descrizioni diverse, mentre ad alcuni non è associata una descrizione) e a 1146 diversi prezzi.

Matteo D'Onofrio - Giacomo Mariani

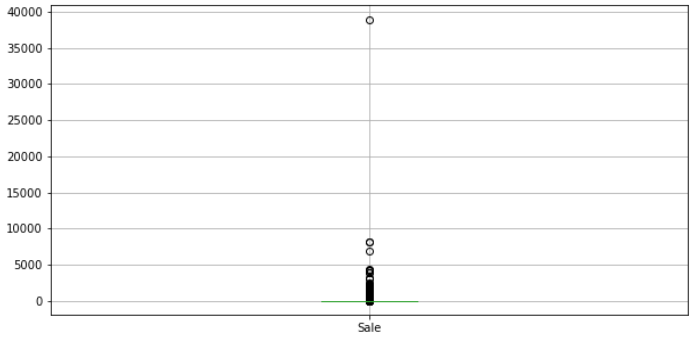
Nel dataset si considerano 4372 distinti clienti, ma 65080 record relativi a 2437 basket non hanno `CustomerID`.

I punti vendita considerati nel dataset sono locati in 38 diversi paesi, con una netta prevalenza di record registrati nel Regno Unito:

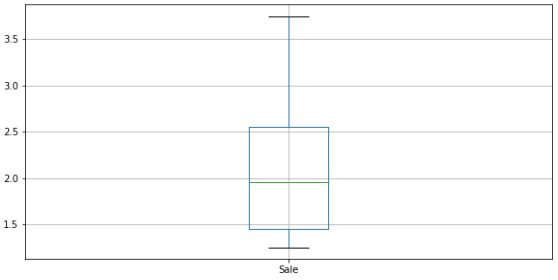


Assessing Data Quality

Dato che la finalità del progetto è la classificazione dei clienti, si decide di rimuovere dal dataset tutti i record in cui non è presente un `CustomerID`. Dopodiché si analizza la distribuzione dei prezzi:

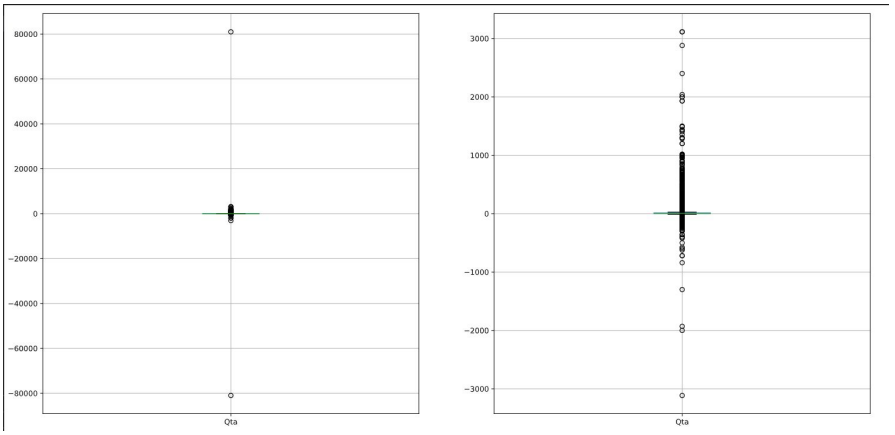


Dato che si vogliono normalizzare i record considerando i prezzi, si sceglie di eliminare i record i cui prezzi non appartengono all'intervallo centrato nell'InterQuantile Range e di raggio 1,5.
Si ottiene quindi una distribuzione normale dei prezzi su 220833 record, con media 2.089 e varianza 0.764357716180097



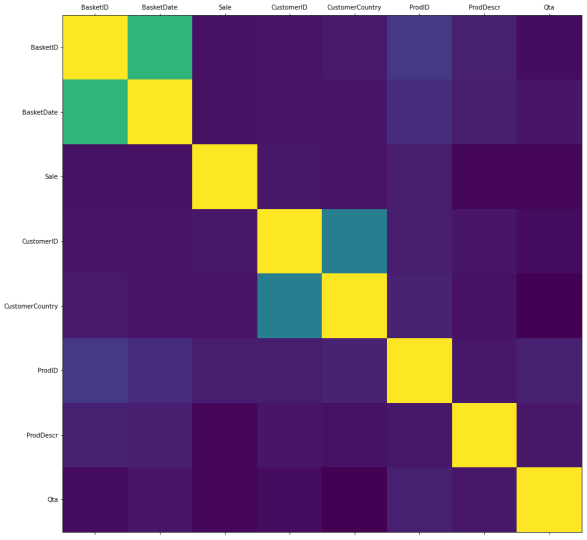
:

Non si ritiene necessario normalizzare allo stesso modo i record anche in base al campo quantità, in quanto in questo caso si eliminerebbero tutti i record il cui prezzo è negativo, perdendo le informazioni su eventuali resi. Andando comunque ad analizzare la distribuzione dell'attributo `Qta` ci si rende conto dell'esistenza di due attributi dalle quantità estremamente alte e contrapposte, che devono essere eliminati. Il boxplot prima e dopo la rimozione di questi due outliers è mostrato di seguito:



Pairwise Correlation

La matrice di correlazione degli attributi del dataset dopo il preprocessing è la seguente:



Dato che non ci sono attributi altamente correlati (più dell'80%) si sceglie di non cancellare nessuna colonna, per cui si definiranno attributi per il customer utilizzando tutti e 8 gli attributi selezionati.

Data Preparation

Per il loro calcolo, essendo indicatori specifici di ogni singolo cliente, è stato effettuato prima un raggruppamento di tutti i record di uno stesso cliente in un nuovo dataset, successivamente questo è stato diviso in 2 sotto-dataset:

1. **dataset positivo** (contenente solo record in cui abbiamo Quantità positive)
2. **dataset negativo** (contenente solo record in cui abbiamo Quantità negative)

Questo ci ha permesso di effettuare un'analisi più specifica e dettagliata.

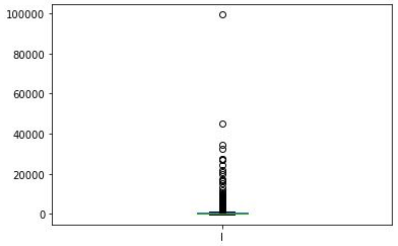
Di seguito sono elencati tali indicatori (vengono elencati quelli supplementari ai 4 suggeriti : I, Iu, Imax, E) :

- **Spending**: somma del denaro emesso da un cliente (considerando solo suoi record con quantità positive)
- **Returning:** somma del denaro restituito ad un cliente (considerando solo suoi record con quantità negative)
- **Maximum_items:** numero massimo di item acquistati in una sessione (tenendo conto anche delle quantità negative, ovvero è definito come la massima differenza tra item comprati e restituiti in una sessione)
- **Total_returned_items:** numero totale di prodotti restituiti
- **Max_cost:** massimo costo unitario, di un prodotto acquistato
- **Min_cost:** massimo costo unitario, di un prodotto restituito
- **Avg_bought:** costo medio dei prodotti acquistati (considerando solo record con quatità positive)
- **Avg_returned:** costo medio dei prodotti restituiti (considerando solo record con quatità negative)
- **Most_bought_cost:** costo del prodotto più acquistato
- **Most_returned_cost:** costo del prodotto più restituito
- **Baskets:** numero di sessioni di acquisto distinte
- **Favorite_country:** paese in cui ha effettuato più acquisti
- **Hour:** orario in cui ha effettuato più acquisti
- **Month:** mese in cui ha effettuato più acquisti

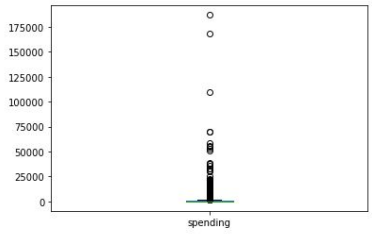
Tramite questi indicatori è stato definito un nuovo dataset: **cutomer_indicators**. Ogni record è identificato dall'attributo **Customer-ID**, lo stesso usato nel dataset inziale.
Nel Notebook è possibile trovare la sua matrice di correlazione, notando che al di là dell'attributo **Spending** non ci sono correlazioni troppo elevate (>= 0.8)

Da un'analisi di questo nuovo dataset sono emersi i degli outliers per i seguenti attributi:

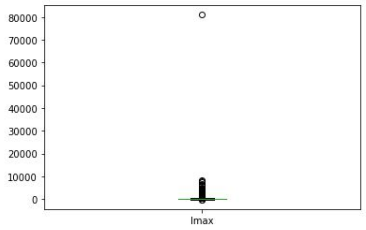
- **I**



- **Spending**



- **Imax**



Di conseguenza si è deciso di rimuovere tali valori:

- record con attributo **I** avente valore = 100000
- record con attributo **Spending** > 150000
- record con attributo **Imax** = 80000

Clustering analysis

Per questo task si è scelto di realizzare anche il subtask opzionale, utilizzando la tecnica dell'**X-Means**.

La dimensione del dataset utilizzato per effettuare il clustering, **customer_indicators**, è di 4180 x 17; tale numero di attributi subito ha sollevato, durante i primi esperimenti, **il problema della dimensionalità (curse of dimensionality)**.
In seguito è stato rilevato anche un problema dato dall'**alta densità dei record nello spazio**, che ha reso difficile la classificazione in cluster.

Di conseguenza si è scelto di individuare **due sottoinsiemi di attributi**, e di applicare ogni algoritmo ad entrambi, per poi valutare i risultati ottenuti.
Si consideri che in ogni sottoinsieme, non sono presenti coppie di attributi con una correlazione > 0.8 .

La scelta di tali attributi è stata fatta in base alla significatività di essi, al fine e il tipo di segmentazione che volevamo effettuare e al valore della correlazione tra loro; di conseguenza abbiamo preso attributi che possono meglio descrivere e distinguere il comportamento dei clienti.

1) **Sottoinsieme-1** di 8 attributi : { I, Iu, Imax, Returning, Baskets, E, Hour, Month } e la relativa matrice di correlazione:

	I	Iu	Imax	returning	baskets	E	hour	month
I	1.000000	0.530615	0.707347	0.244277	0.693377	0.423775	-0.061254	-0.012969
Iu	0.530615	1.000000	0.296171	0.088731	0.598218	0.778808	0.063965	0.013881
Imax	0.707347	0.296171	1.000000	0.256644	0.243636	0.307470	-0.058209	0.009419
returning	0.244277	0.088731	0.256644	1.000000	0.229044	0.043564	-0.016313	0.005911
baskets	0.693377	0.598218	0.243636	0.229044	1.000000	0.444196	-0.040210	-0.032778
E	0.423775	0.778808	0.307470	0.043564	0.444196	1.000000	0.042977	0.041858
hour	-0.061254	0.063965	-0.058209	-0.016313	-0.040210	0.042977	1.000000	0.048298
month	-0.012969	0.013881	0.009419	0.005911	-0.032778	0.041858	0.048298	1.000000

2) **Sottoinsieme-2** di 6 attributi : { I, Iu, Imax, Most_bought_cost, Avg_bought, E } e la relativa matrice di correlazione

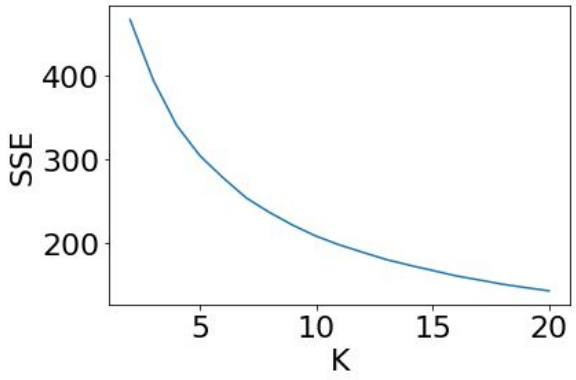
	I	Iu	Imax	most_bought_cost	avg_bought	E
I	1.000000	0.530615	0.707347	-0.047928	-0.030157	0.423775
Iu	0.530615	1.000000	0.296171	-0.132364	-0.035532	0.778808
Imax	0.707347	0.296171	1.000000	-0.074933	-0.052056	0.307470
most_bought_cost	-0.047928	-0.132364	-0.074933	1.000000	0.668207	-0.116143
avg_bought	-0.030157	-0.035532	-0.052056	0.668207	1.000000	-0.013734
E	0.423775	0.778808	0.307470	-0.116143	-0.013734	1.000000

Per quanto riguarda lo **Scaling** dei dati, si è applicato il ridimensionamento **Min-Max**.
Viene di seguito presentato ogni algoritmo, prima con il Sottoinsieme-1 e successivamente con il Sottoinsieme-2.

1) K-Means (Sottoinsieme-1)

1.1) Identificazione del parametro K

Abbiamo misurato il valore SSE, Silhouette e Separation, in diverse esecuzioni al variare del parametro K: da 2,...,20; ottenendo il seguente risultato e grafico:



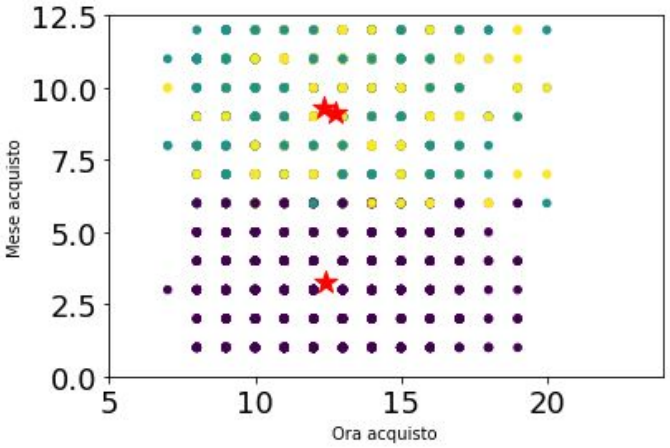
guardando la curva dell'SSE al variare di K, vediamo che il valore smette di decrescere rapidamente dopo K>5. Si sceglie però K=3, perchè negli esperimenti fatti con K=4,5 si ottenevano centroidi quasi del tutto sovrapposti, che non riportavano quindi alcuna informazione utile alla classificazione, cosa che non accade invece con K=3.

Con K=3 abbiamo ottenuto le seguenti metriche:

SSE 394.74589677038114
Silhouette 0.2651611365701626
Separation 1.3449420538617698

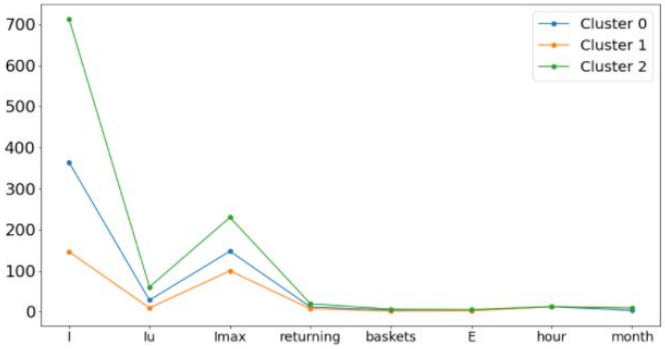
1.2/1.3) Descrizione dei centroidi e della distribuzione dei record nei cluster

Il valore della Silhouette= 0,265 è abbastanza basso, ad indicare la presenza di cluster non nettamente separati e distanti.
Tale intuizione viene confermata guardando alcuni dei plot presenti nel notebook:

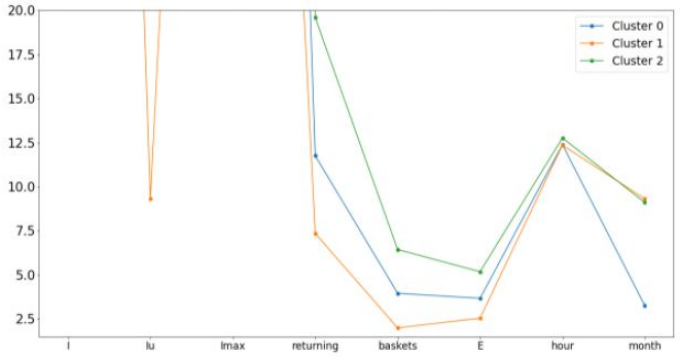


Nell'ultimo riusciamo a vedere una classificazione più netta che nell'altro, ma con due centroidi molto vicini tra loro.

Di seguito riportiamo la visualizzazione dei valori medi dei centroidi:



Qui abbiamo uno Zoom sulla parte in basso a dx:



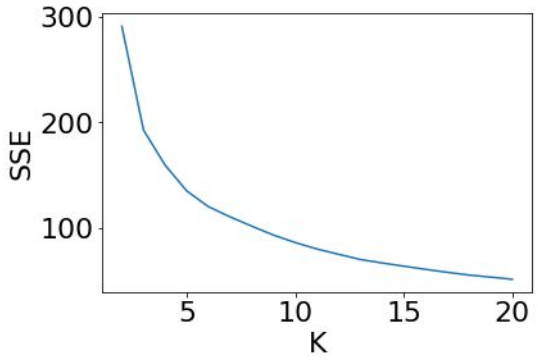
In base a quanto visto dalla distribuzione dei record nei cluster, e dalla media dei valori dei centroidi, individuiamo 3 categorie di clienti:

1) **Cluster-0 (viola nel plot della distribuzione):** clienti "nella media", coprono la fetta maggiore della clientela con tutti i valori degli attributi intermedi, con propensione all'acquisto verso i primi mesi dell'anno.
2) **Cluster-1 (azzurro nel plot della distribuzione):** clienti che hanno effettuato meno acquisti di tutti, pochi item totali, pochi item distinti (acquisti simili e ripetuti), poche sessioni di acquisto ed entropia bassa (oggetti sempre molto simili), con propensione di acquisto negli ultimi mesi dell'anno.
3) **Cluster-2 (gialli nel plot della distribuzione):** clienti che hanno effettuato più acquisti di tutti, in più sessioni distinte e con un'entropia maggiore, di conseguenza molte sessioni con acquisti molto variegati.

K-Means (Sottoinsieme-2)

1.4) Identificazione del parametro K

Abbiamo misurato il valore SSE, Silhouette e Separation, in diverse esecuzioni al variare del parametro K: da 2,...,20; ottenendo il seguente risultato e grafico:



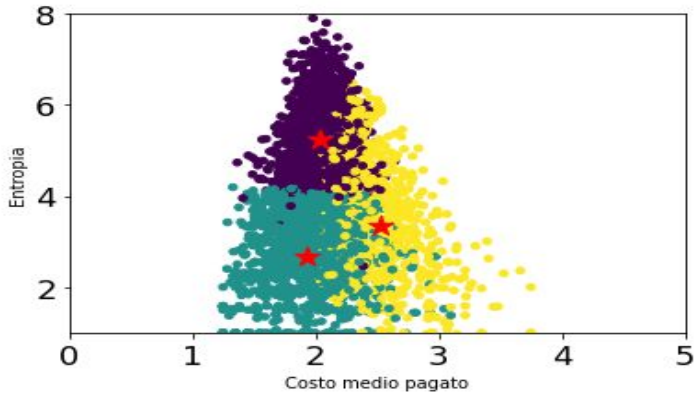
Guardando la curva dell'SSE al variare di K, vediamo che il valore smette di decrescere rapidamente dopo K>5. Si sceglie però K=3, perche negli esperimenti fatti con K=4,5 si ottenevano centroidi quasi del tutto sovrapposti, che non riportavano quindi alcuna informazione di classificazione, cosa che non accade invece con K=3.
(Considerazioni analoghe agli esperimenti fatti con il Sottoinsieme-1, ma con valori differenti)

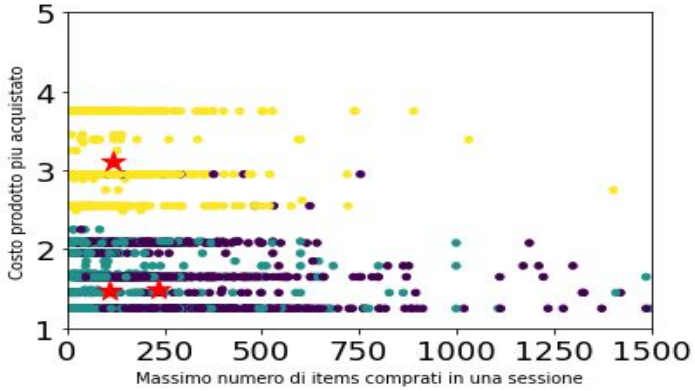
Con K=3 abbiamo ottenuto le seguenti metriche:

SSE 192.73544272800152
Silhouette 0.36370536331047687
Separation 0.9529479245704291

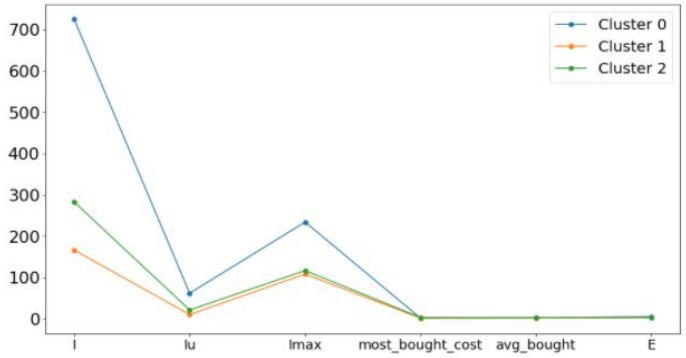
1.5/1.6) Descrizione dei centroidi e della distribuzione dei record nei cluster

Qui si registra un valore delle metriche SSE e Separation minore, con conseguente valore di Silhouette maggiore, denotando una migliore classificazione; ciò è possibile riscontrarlo in alcuni plot:

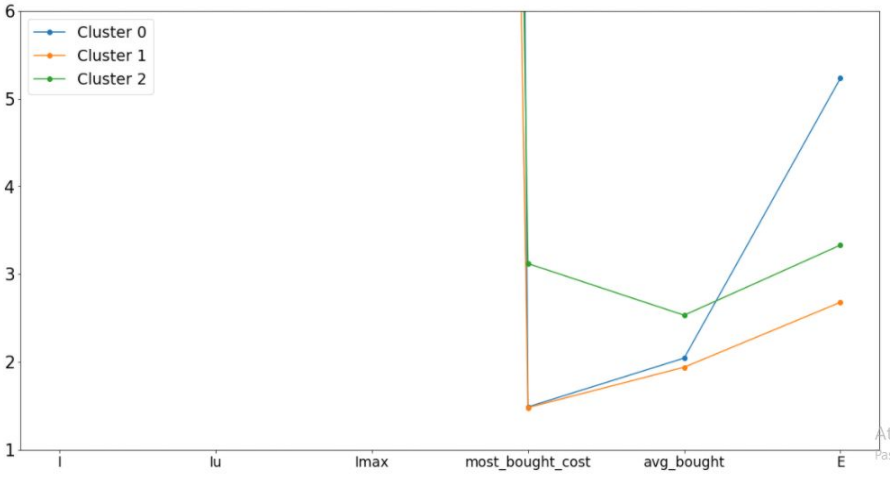




Di seguito riportiamo la visualizzazione dei valori medi dei centroidi:



applichiamo uno zoom in basso a dx:



In base a quanto visto dalla distribuzione dei record nei cluster, e dalla media dei valori dei centroidi, individuiamo anche qui 3 categorie di clienti, ma in questo caso si riesce a fare una distinzione più dettagliata:

4) **Cluster-0 (viola nel plot della distribuzione):** clienti che hanno effettuato più acquisti di tutti, in più sessioni distinte, con il costo del prodotto più acquistato minore e un'entropia maggiore. Una profilazione di un tipo di cliente che compra molto, tanti oggetti diversi e tutti a basso costo.
5) **Cluster-1 (colore azzurro nel plot della distribuzione):** clienti che hanno effettuato meno acquisti di tutti, pochi item totali, pochi item distinti (acquisti simili e ripetuti), poche sessioni di acquisto ed entropia bassa (oggetti sempre molto simili), costo del prodotto più acquistato minore e spesa media minore. Una profilazione di un tipo di cliente che compra poco, spese basse e sempre gli stessi prodotti.
6) **Cluster-2 (gialli nel plot della distribuzione):** clienti nella media per quanto riguarda il numero di prodotti acquistati, il numero di sessioni fatte e l'entropia; ma a differenza degli altri hanno il costo del prodotto più acquistato e costo del prodotto medio più alto di tutti. Una profilazione di un tipo di cliente che compra un numero di prodotti medio ma su cui spende di più.

Complessivamente l'applicazione del K-Means sul Sottoinsieme-2, si è rivelata migliore permettendoci di ottenere classificazioni più distinte.

2) Density based clustering (Sottoinsieme-1)

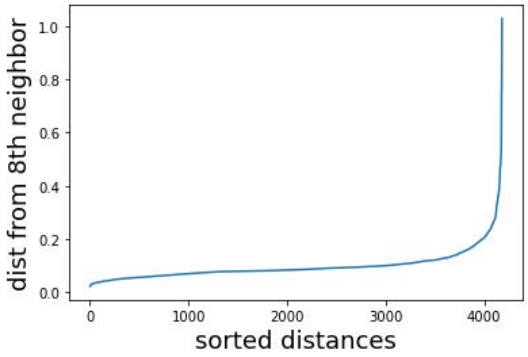
Come descritto precedentemente, anche per questo algoritmo è stato applicato lo Scaling con il ridimensionamento **Min-Max.**

2.1) identificazione e studio dei parametri

Per l'identificazione dei parametri **Eps** e **Min_sample,** abbiamo:

1) calcolato la distanza da ogni punto (record) verso tutti gli altri
2) effettuato diversi plot al variare del valore K tra [2,...,20]
3) tali plot sono cosi' realizzati:
    a) sull'asse x, mettiamo tutti i punti ordinati, in ordine crescente, in base alla distanza dal loro k-esimo punto più vicino
    b) sull'asse y, mettiamo le distanze da tale k-esimo elemento più vicino

Di seguito sono riportati alcuni dei plot:



distanza del punto 4000 dal 8-vicino: 0.2055305772533702

Notiamo che ovviamente, aumentando il valore di k, aumenta la distanza del k-esimo punto più vicino. Di conseguenza abbiamo effettuato vari test, con differenti applicazioni dell'algoritmo DBSCAN, al variare di questi parametri, ottenendo come miglior risultato tale configurazione:

Eps = 0.209
Min_samples = 8

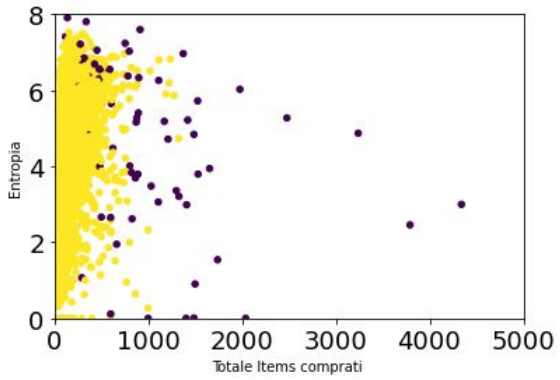
Con metrica Silhouette = 0.34

2.2) caratterizzazione e interpretazione dei cluster

I cluster individuati sono 2, ma con un estremo sbilanciamento:

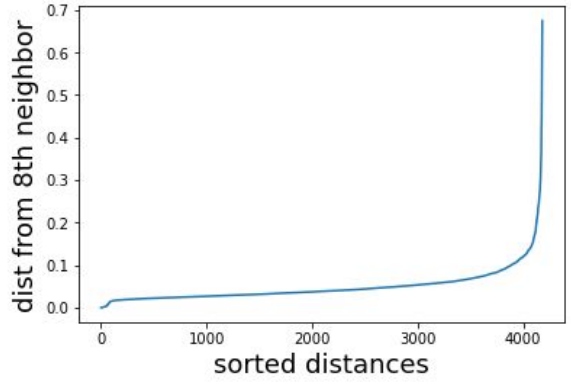
- 77 record nel Cluster-0
- 4013 record nel Cluster-1

Data l'alta dimensionalità e l'alta densità dei dati, tale classificazione tende a concentrarsi tutta su uno specifico cluster, com'è anche evidente dal seguente plot:



Density based clustering (Sottoinsieme-2)

2.3) identificazione e studio dei parametri



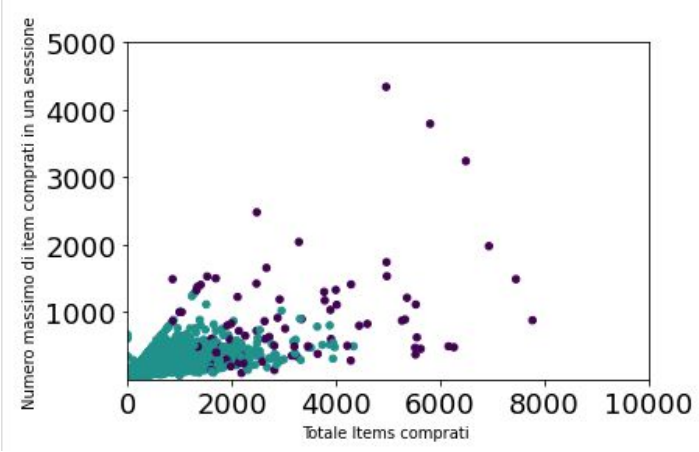
Procedendo come descritto prima, qui troviamo distanze più basse.
Dopo numero esperimenti anche qui abbiamo trovato che i parametri che meglio classificano sono:
Eps = 0.1209
Min_samples = 8

Con metrica Silhouette = 0.38

2.4) caratterizzazione e interpretazione dei cluster

I cluster individuati sono 3; tuttavia nonostante la diminuzione della dimensionalità data dal minore sottoinsieme di attributi, abbiamo ancora un estremo sbilanciamento:

- 86 record nel Cluster-0
- 4071 record nel Cluster-1
- 23 record nel Cluster-2



Da un'analisi sommaria di entrambe le applicazioni di DBSCAN è possibile evincere che:

- Cluster-0 e Cluster-2: rappresentano i clienti agli estremi della distribuzione
- Cluster-1: rappresentano i clienti con valori intorno o uguali alla media

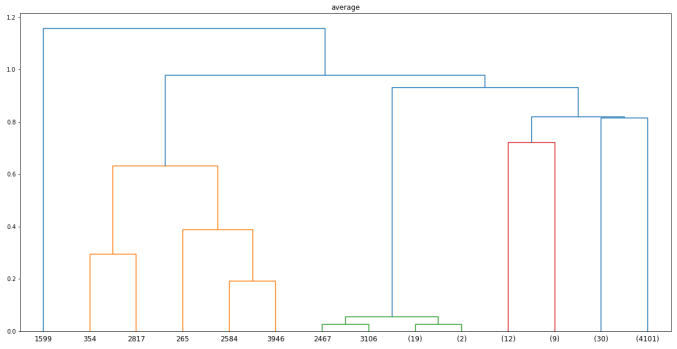
3) Clustering Gerarchico

Per eseguire una suddivisione gerarchica dei cluster è stato usato l'algoritmo di Agglomerative Clustering implementato all'interno di SciKit - Learn. Si analizza l'esecuzione dell'algoritmo utilizzando la distanza **euclidea** al variare delle metriche di **linkage** utilizzate, considerando in sequenza i vari sottoinsiemi. Dato che, per la struttura stessa dell'implementazione, non è possibile specificare in partenza il numero di clusters quando si vuole visualizzare il dendrogramma, si fissa semplicemente a 0 la massima distanza entro cui fermare lo splitting, generando in questo modo tutto l'albero della gerarchia. Nei dendrogrammi si visualizzano comunque i primi 4 livelli di splitting, in quanto si vuole apprezzare la distribuzione delle istanze rispetto alla distanza.

1. Sottoinsieme 1

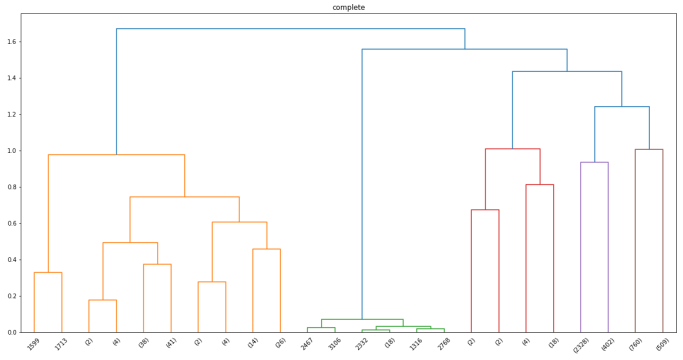
1.1. Average linkage

La distanza tra due cluster è la media delle distanze tra i punti nei due cluster diversi.



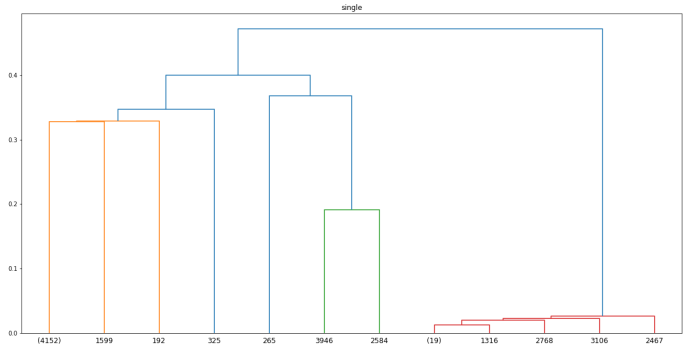
1.2. Complete linkage

La distanza tra due cluster è la massima distanza tra due punti.



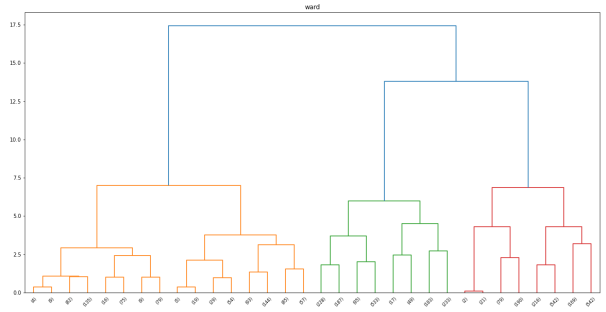
1.3. Single linkage

La distanza tra due cluster è la minima distanza tra due punti nei cluster.



1.4. Ward linkage

A differenza delle altre tecniche basate sulla distanza, Ward minimizza la varianza dei punti all'interno dei cluster.

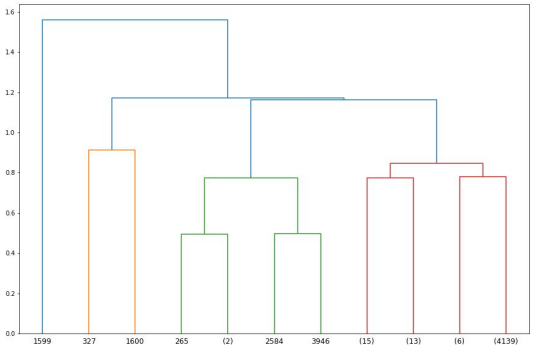


Analisi del clustering

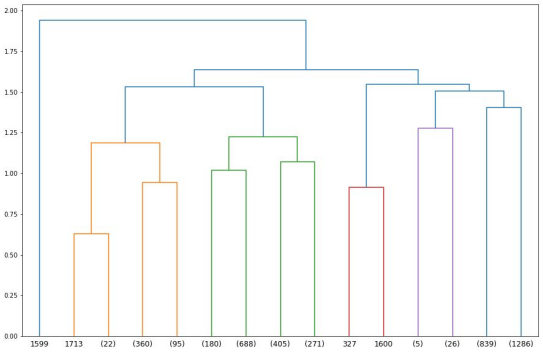
Si riconferma la tendenza già notata con K-Means, per cui i clusters di dimensione considerevole (nel grafico apprezzabili in quanto hanno colore diverso dal blu) sono generalmente 3. Unica eccezione il Complete Linkage, che genera 5 cluster, in quanto si suppone che tenda a clusterizzare tra loro i record più isolati dai clusters centrali.

2. Sottoinsieme 2

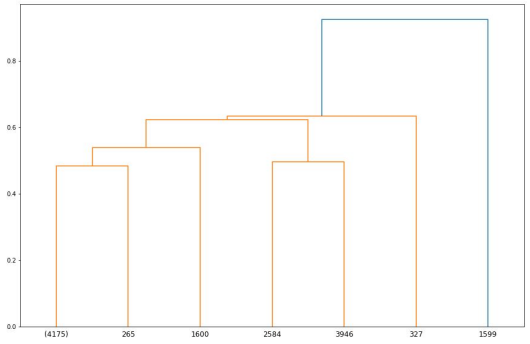
2.1. Average linkage



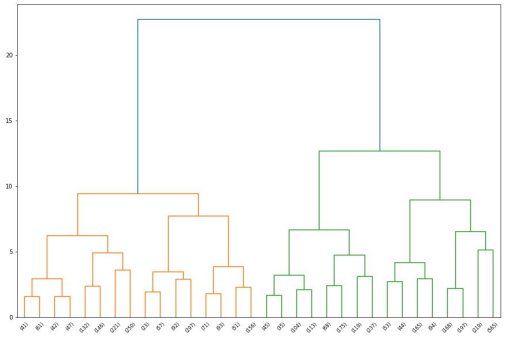
2.2. Complete linkage



2.3. Single linkage



2.4. Ward linkage



Analisi del clustering

Questo clustering, essendo basato su una matrice contenente meno attributi, genera meno clusters la cui distanza inter-cluster è molto più uniformata del caso precedente. Si vengono comunque a creare 3 cluster

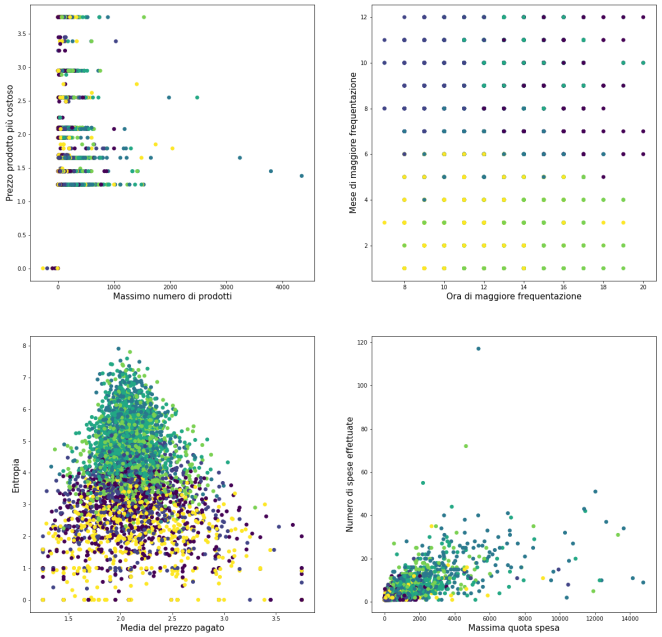
4) X-Means

L'algoritmo X-Means permette di non specificare il numero di clusters che si desidera produrre, ma è possibile in pratica specificare un range di clusters entro cui l'algoritmo può procedere. Per questo si è deciso di utilizzare X-Means con un numero minimo di 2 e massimo di 6 cluster, calcolando i centroidi di partenza con l'algoritmo K-Means++.

Sottoinsieme 1

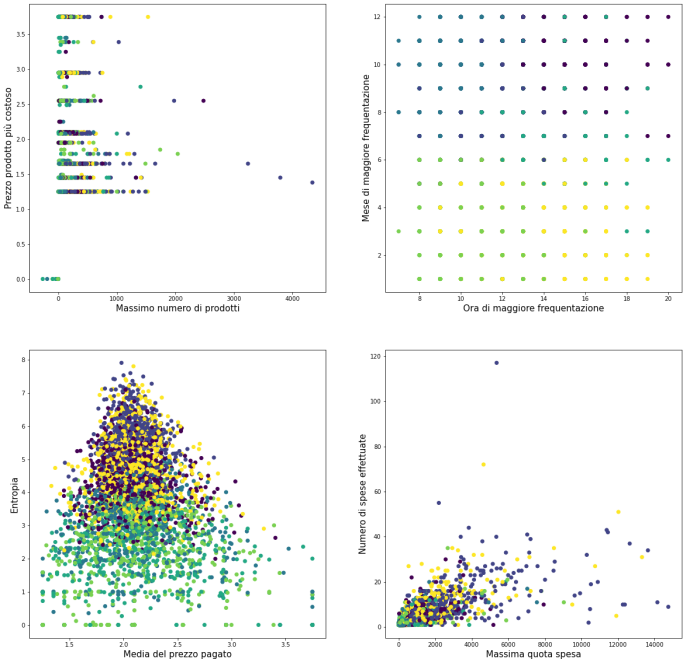
Il risultato dell'esecuzione ha portato all'identificazione di 6 clusters, con un SSE di 279.4519178963032 e silhouette score di 0.3440533568692403, ad indicare una situazione

di clustering estremamente poco separabile. Di seguito una visualizzazione dei clusters, che insieme alle suddette metriche può essere ricondotto ai risultati già evidenziati da K-Means.



Sottoinsieme 2

Come sopra, il clustering ha portato ad un SSE di 283.33319458800355 con silhouette score di 0.3385475239296091, la cui visualizzazione è la seguente:



Conclusioni

I dati contenuti nel dataset sono estremamente vari e distribuiti in maniera non omogenea. La nostra scelta delle features ha evidenziato inoltre un problema di dimensionalità che ha reso necessario eseguire l'analisi su due sottoinsiemi parzialmente sovrapposti delle features per validare il risultato.
Quello che ne è conseguito è però chiaramente una suddivisione in circa 3 clusters, riconducibili al tipo di cliente della catena di negozi. I clusters non sono nettamente separati e anzi, presentano numerose sovrapposizioni.

Il clustering gerarchico ha mostrato però la prevalenza, al variare degli algoritmi di linkage utilizzati, di 3 clusters equidistanti rispetto alla distanza euclidea, che suggeriscono che questa sia la suddivisione ottimale anche per l'esecuzione di altri algoritmi. Per contro, X-Means ha mostrato un eccessivo livello di dettaglio, non fornendo informazioni illustrative più di quanto già fatto con l'algoritmo K-Means.

Il clustering partizionale, con K-Means, ha evidenziato analogamente la presenza di 3 clusters, concentrati e densi, ma che riescono a definire 3 tipi differenti di clientela. L'algoritmo DBSCAN invece, data l'alta densità dei record, non ha trovato nessuna clusterizzazione con contenuto informativo utile.