Relazione Progetto Data Mining

Matteo Ceregini e Sara Montemaggi

Gruppo 13

1 Data Understanding e Data Preparation

Il dataset contiene informazioni riguardanti gli acquisti fatti in un negozio, presumibilmente uno store online, tra il 2010 e il 2011. In questa prima parte vengono analizzati i dati, descrivendo la semantica degli attributi del dataset. Inoltre, verrà discussa la qualità dei dati, provvedendo a risolvere eventuali problemi riscontrati durante l'analisi, come ad esempio valori mancanti o errati. Infine, verranno indagati alcuni aspetti statistici come la distribuzione degli attributi e la correlazione tra coppie di attributi.

1.1 Semantica degli attributi

Ogni record del dataset *customer_supermarket* rappresenta un acquisto di un determinato prodotto fatto da un certo utente. Ogni record è composto dai seguenti attributi:

- **BasketID**: identifica univocamente il carrello di cui l'acquisto fa parte. Ogni carrello può contenere uno o più acquisti e ogni acquisto fa parte di un solo carrello.
- **BasketDate**: indica la data in cui è stato fatto l'acquisto. Tutti gli acquisti appartenenti a un determinato carrello sono fatti nella stessa data.
- **Sale**: indica il prezzo per unità del prodotto acquistato.
- **CustomerID**: identifica univocamente il cliente che ha effettuato l'acquisto.
- **CustomerCountry**: indica la nazione o zona geografica in cui il cliente risiede.
- **ProdID**: identifica univocamente il prodotto acquistato.
- **ProdDescr**: consiste in una breve descrizione del prodotto acquistato.
- **Qta**: indica quante unità di prodotto sono state acquistate.

1.2 Qualità dei dati

Record duplicati Nel dataset sono presenti dei record duplicati. Dato che il dataset rappresenta lo storico degli acquisti, si suppone che i record duplicati siano dovuti a qualche errore nel sistema e quindi possano essere eliminati. Inoltre, viene esclusa la possibilità che l'utente possa aggiungere al carrello lo stesso prodotto più volte (come acquisti separati). Questa assunzione viene fatta poiché nei maggiori store online (ad esempio amazon.com), se l'utente prova ad aggiungere al carrello lo stesso oggetto più volte, viene incrementato il valore della quantità relativo a quell'oggetto e non viene aggiunto un nuovo acquisto.

Miglioramento della qualità dei dati In seguito, per ogni attributo vengono descritte le eventuali operazioni effettuate per migliorarne la qualità così da facilitarne una futura analisi.

- **CustomerID**: alcuni record hanno CustomerID = NaN. Dato che la futura analisi si concentrerà sui clienti, se non sarà possibile recuperare i CustomerID mancanti allora i rispettivi record verranno eliminati. Una strategia per cercare di recupare i CustomerID mancanti è la seguente: per ogni record r_1 con CustomerID = NaN, si controlla nel dataset se esiste un record r_2 tale che r_2.BasketID è uguale a r_1.BasketID e r_2. CustomerID $\neq$ NaN. Se il record r_2 esiste, allora è assegnato al record r_1 il CustomerID di r_2. Infatti, se degli acquisti appartengono allo stesso carrello allora essi sono fatti dallo stesso cliente. Tuttavia, applicando questa strategia non è stato possibile recuperare alcun CustomerID, quindi tutti i record aventi CustumerID = NaN sono stati eliminati.

I CustomerID sono salvati come dei float, ma nessuno di essi ha parte decimale uguale a zero, quindi sono stati convertiti in integer in quanto è un formato più adatto a rappresentare degli ID.

- **ProdID**: alcuni record del dataset non rappresentano degli acquisti e quindi possono essere eliminati. Questi record si distinguono dal fatto che hanno come ProdID una delle seguenti stringhe: *POST, M, D, DOT, CRUK, PADS, C2, BANK CHARGES*.
- **ProdDescr**: nel dataset sono presenti dei record aventi ProdDescr = NaN. Tali record non creano problemi ai fini dell'analisi. Inoltre, si è notato che essi vengono eliminati dopo aver apportato le modifiche al dataset descritte nei punti precedenti.
- **BasketDate**: è stato controllato che tutti i valori dell'attributo rappresentino delle date valide e che esse siano scritte secondo lo stesso formato.
- **Sale**: i valori di questo attributo sono stati salvati come delle stringhe sebbene questi valori rappresentino il costo di un'unità di uno specifico prodotto. Infatti, è utilizzata la virgola per separare la parte intera dalla parte decimale. Le virgole sono state quindi sostituite con dei punti e i valori sono stati trasformati in float. Nel dataset sono presenti alcuni record con Sale = 0. Si è ipotizzato che questi record rappresentano dei prodotti omaggio e quindi è stato deciso di eliminarli in quanto non interessanti per l'analisi.
- **CustomerCountry**: all'interno del dataset sono presenti alcuni record aventi valore dell'attributo CustomerCountry uguale a "Unspecified". È stato deciso di considerare tale valore corretto, in quanto il cliente in fase di acquisto potrebbe non aver voluto specificare la nazione o zona geografica dove risiede, ad esempio per motivi di privacy.
- **Qta e BasketID**: all'interno del dataset sono presenti dei record aventi valore dell'attributo Qta < 0. Ciascuno di questi record ha il valore di BasketID che inizia con la lettera C. Si è ipotizzato che tali record rappresentano dei rimborsi richiesti dai clienti in caso di oggetti danneggiati. Si è quindi deciso di applicare i rimborsi agli acquisti, facendo le seguenti assunzioni:

  1. Per ogni ordine è possibile richiedere al massimo un rimborso.
  2. Un rimborso può essere applicato ad un ordine solo se la data dell'ordine è antecedente a quella del rimborso, il ProdID è lo stesso, il CustomerID è lo stesso e la quantità di oggetti presente nell'ordine è maggiore o uguale alla quantità in valore assoluto degli oggetti indicata nel rimborso.
  3. Dato un ordine, se ci sono più rimborsi applicabili a questo ordine, si applica quello con la quantità di oggetti in valore assoluto maggiore. Se ci sono due o più rimborsi con la stessa quantità scegliamo uno tra questi.

  Dopo aver applicato i rimborsi, alcuni record hanno Qta = 0, ciò significa che il cliente ha annullato l'intero ordine. Quindi, tali record possono essere eliminati. Inoltre, dopo che i rimborsi sono stati eliminati dal dataset, tutti i valori dell'attributo BasketID sono delle stringhe che rappresentano dei numeri interi e quindi sono stati convertiti in integer.

Aggiunta dell'attributo Spending Si è aggiunto al dataset un nuovo attributo chiamato *Spending* che rappresenta la spesa totale relativa ad un acquisto. Quindi, per ogni record del dataset si ha che Spending = Sale $\times$ Qta.

1.3 Distribuzione dei dati

In seguito sono discusse alcune considerazioni generali riguardanti il dataset e le distribuzioni di alcuni attributi.

Periodo di osservazione Il dataset rappresenta uno storico degli acquisti effettuati tra il 1 dicembre 2010 e il 9 dicembre 2011.

Prodotti più acquistati Per quanto riguarda gli attributi ProdDescr, è interessante capire quali sono i dieci prodotti che sono stati acquistati maggiormente durante il periodo di osservazione. Sono riportati solo i primi dieci prodotti più acquistati così da ottenere un istogramma più comprensibile. In Figura 1 è riportato l'istogramma ottenuto.

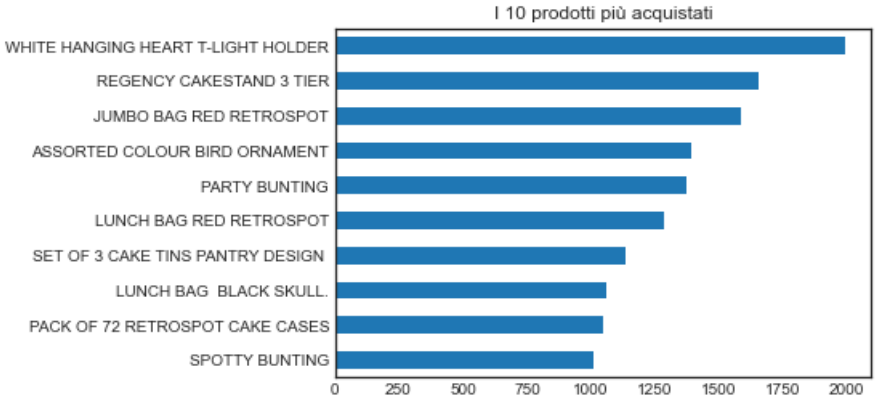
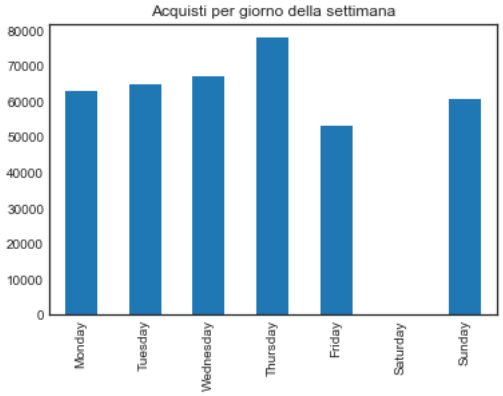


Figura 1: I 10 prodotti più acquistati.

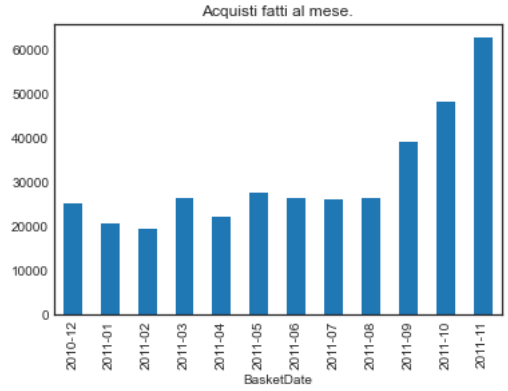
Mesi in cui sono stati fatti più acquisti Utilizzando l'attributo BasketDate è possibile determinare la distribuzione degli acquisti durante il periodo di osservazione. In Figura 2b è rappresentato il numero di acquisti effettuatati mensilmente tra dicembre 2010 e novembre 2011. Sono stati esclusi gli acquisti fatti nel mese di dicembre 2011 in quanto il mese sembra essere incompleto, ossia sono presenti solo gli acquisti fatti nei primi nove giorni del mese. Si nota che i mesi in cui sono stati fatti più acquisti sono novembre 2011, ottobre 2011 e settembre 2011.

Giorni della settimana in cui sono stati fatti più acquisti Utilizzando l'attributo BasketDate è possibile determinare la distribuzione degli acquisti rispetto ai giorni della settimana. In Figura 2a sono rappresentati il numero di acquisti che sono stati effettuati per ogni giorno della settimana, durante il periodo di osservazione. Si nota che il giovedì è il giorno in cui sono stati effettuati più acquisti.

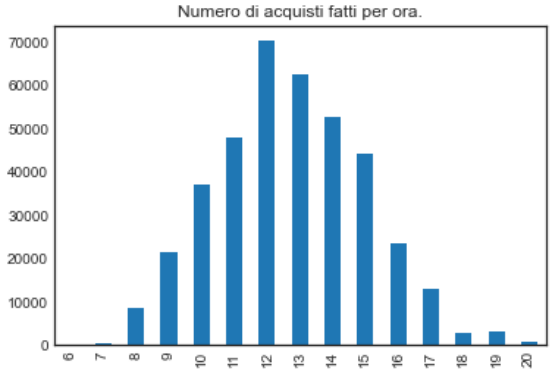
Ore del giorno in cui sono stati fatti più acquisti Utilizzando l'attributo BasketDate è possibile determinare in quali ore del giorno sono stati fatti più acquisti. In Figura 2c sono rappresentati il numero di acquisti che sono stati effettuati per ogni ora del giorno, durante il periodo di osservazione. Se una determinata ora non compare nell'istogramma allora significa che non sono stati fatti acquisti durante quella specifica ora. Si nota che la maggior parte degli acquisti è stata effettuata tra la ore 11,00 e le ore 14,00.



(a) Distribuzione settimanale degli acquisti.



(b) Distribuzione mensile degli acquisti.



(c) Distribuzione degli acquisti rispetto alle ore del giorno.

Figura 2: Grafici relativi alla distribuzione degli acquisti nei diversi periodi temporali.

Paesi o zone geografiche da cui sono stati fatti più acquisti Utilizzando l'attributo CustomerCountry è possibile determinare quali sono i Paesi o zone geografiche da cui sono stati fatti più acquisti. Nel seguente grafico l'area totale del rettangolo rappresenta la totalità degli acquisti. L'area di ogni sotto-rettangolo è proporzionale al numero di acquisti fatti da clienti residenti nel Paese o zona geografica che il sotto-rettangolo rappresenta.

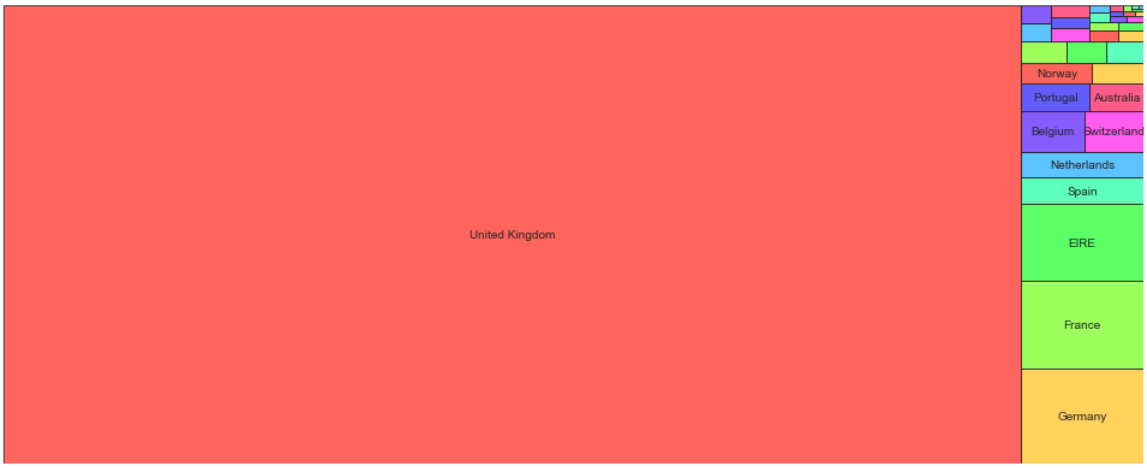
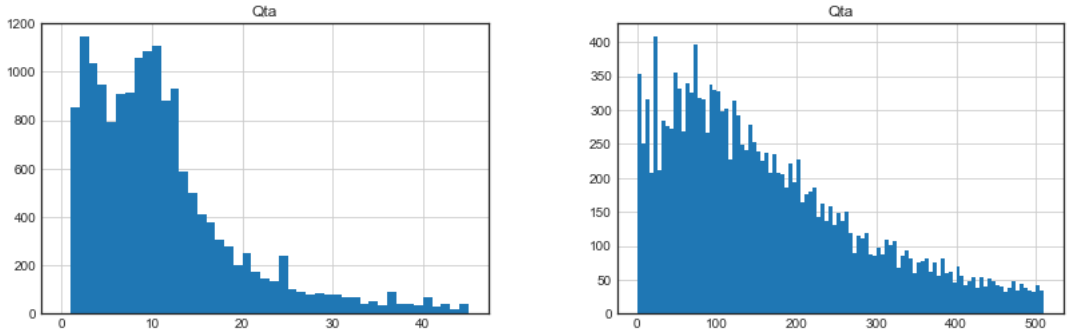


Figura 3: Distribuzione degli acquisti per paese o zona geografica.

Numero medio di prodotti uguali acquistati per carrello Utilizzando gli attributi BasketID e Qta è possibile determinare il numero medio di prodotti uguali acquistati per carrello. Per ottenere un grafico più leggibile, si è deciso di visualizzare solo il 90% dei carrelli. In Figura 4a è riportato il grafico ottenuto. Dal grafico si nota che il 90% dei carrelli ha un numero medio di prodotti uguali ≤ 44 e che ci sono due picchi in corrispondenza circa dei valori 3 e 11.

Numero totale di prodotti acquistati per carrello Utilizzando gli attributi BasketID e Qta è possibile determinare il numero totale di prodotti acquistati per carrello. Anche in questo caso, per ottenere un grafico più leggibile, è stato visualizzato solo il 90% dei carrelli. In Figura 4b è riportato il grafico ottenuto. Si nota che il 90% dei carrelli ha un numero totale di prodotti acquistati ≤ 505.



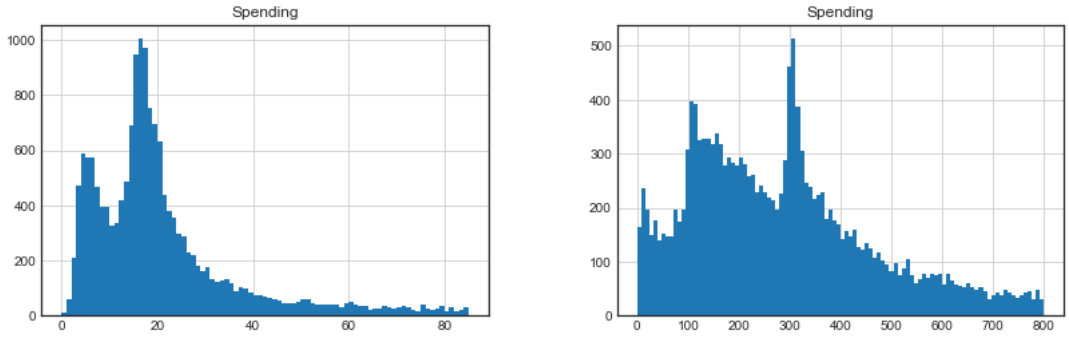
(a) Numero medio di prodotti uguali acquistati per carrello (b) Numero totale di prodotti acquistati per carrello

Figura 4: Grafici relativi al numero di prodotti per carrello.

Spesa media per prodotto acquistato Utilizzando gli attributi BasketID e Spending è possibile determinare la spesa media per prodotto acquistato. Per ottenere un grafico più leggibile, è stato deciso di visualizzare il 90% dei carrelli. In Figura 5a è riportato il grafico ottenuto. Dal

grafico si nota che il 90% dei carrelli ha una spesa media per prodotto ≤ 82.84. Inoltre, ci sono due picchi in corrispondenza circa dei valori 5 e 17.

Spesa totale per carrello Utilizzando gli attributi BasketID e Spending è possibile determinare la spesa totale fatta per carrello. Per ottenere un grafico più leggibile, è stato deciso di visualizzare solo il 90% dei carrelli. In Figura 5b è riportato il grafico ottenuto. Dal grafico si nota che il 90% dei carrelli consiste in una spesa totale ≤ 789.96. Inoltre, ci sono due picchi in corrispondenza circa dei valori 100 e 300.



(a) Spesa media per prodotto acquistato

(b) Spesa totale per carrello

Figura 5

Correlazione tra coppie di attributi In seguito è riportata la matrice di correlazione tra gli attributi numerici significativi del dataset, cioè gli attributi Sale, Qta e Spending. Si nota che Qta e Spending sono leggermente correlati, infatti una quantità maggiore di un certo prodotto implica necessariamente un aumento della spesa per comprare tale prodotto.

	Sale	Qta	Spending
Sale	1.000000	-0.088281	0.094269
Qta	-0.088281	1.000000	0.653559
Spending	0.094269	0.653559	1.000000

Figura 6: Correlazione tra coppie di attributi

1.4 Calcolo degli attributi aggiuntivi

È stato creato un nuovo dataset, chiamato "clienti", dove ogni record rappresenta un cliente diverso. Ogni record ha i seguenti attributi:

- **I**: rappresenta il numero totale di oggetti acquistati dal cliente durante l'intero periodo di osservazione.
- **Iu**: rappresenta il numero di oggetti distinti acquisti dal cliente durante l'intero periodo di osservazione.
- **Imax**: rappresenta il massimo numero di oggetti acquistati da un cliente in un carrello.

- **NumBasket**: rappresenta il numero totale di carrelli che il cliente ha fatto durante il periodo di osservazione.
- **Recency**: rappresenta il numero di giorni che il cliente è stato inattivo, con riferimento alla data più recente contenuta nel dataset.
- **Frequency**: rappresenta il numero totale di ordini fatti dal cliente, durante l'intero periodo di osservazione.
- **MonetaryValue**: rappresenta quanto il cliente ha speso in totale, durante l'intero periodo di osservazione.
- **EntropyDays**: rappresenta l'entropia di Shannon rispetto al giorno della settimana in cui l'utente fa acquisti. Indica quanto il cliente tende a fare acquisti sempre nello stesso giorno della settimana.
- **EntropySpending**: rappresenta l'entropia di Shannon rispetto alla spesa totale per ordine. Indica quanto il cliente tende a spendere sempre circa la stessa cifra.
- **EntropyProducts**: rappresenta l'entropia di Shannon rispetto ai prodotti acquistati dall'utente. Indica quanto il cliente tende ad acquistare sempre gli stessi prodotti.

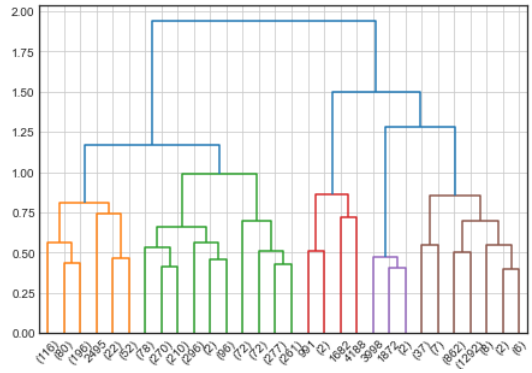
1.5 Eliminazione di attributi ridondanti

Per identificare e eliminare eventuali attributi ridondanti è stata utilizzata la matrice di correlazione. Sono stati eliminati gli attributi I, Iu, NumBasket e EntropySpending in quanto hanno un'alta correlazione con i restanti attributi. In particolare, l'attributo I ha correlazione 0.92 con l'attributo MonetaryValue; l'attributo Iu ha correlazione 0.88 con l'attributo Frequency; l'attributo NumBasket ha correlazione 0.75 con l'attributo Frequency e l'attributo EntropySpending ha correlazione 0.92 con l'attributo EntropyProducts. Inoltre, si è deciso di eliminare anche l'attributo EntropyDays. Infatti, nel dataset ci sono molti clienti che hanno fatto un solo acquisto e quindi l'attributo EntropyDays risulta essere poco significativo. Quindi, il dataset su cui sarà effettuato il clustirng contiene gli attributi Imax, Recency, Frequency, MonetaryValue e EntropyProducts.

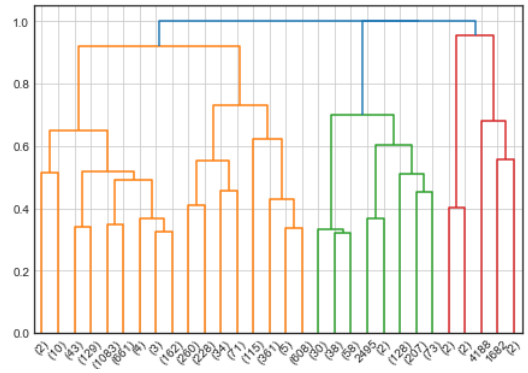
2 Clustering

2.1 Clustering Gerarchico

Per prima cosa è stato eseguito il clustering gerarchico, così da ottenere una stima del numero di cluster e utilizzare tale informazione in congiunzione al knee method per ottenere un buon valore di K per l'algoritmo KMeans. Il clustering gerarchico è stato eseguito usando due metriche diverse: la distanza Euclidea e la distanza di Chebyshev. Per ognuna di queste metriche sono stati usati tre tipologie di linkage: Complete, Single e Average. In seguito sono riportati i dendrogrammi (generati con l'opzione truncate_mode = "lastp") ritenuti significati.



(a) Dendrogramma ottenuto usando il la distanza Euclidea e linkage Complete



(b) Dendrogramma ottenuto usando la distanza di Chebyshev e il linkage Complete

Dall'analisi dei dendrogrammi si è concluso che un possibile valore di K da utilizzare nell'algoritmo KMeans potrebbe essere K = 3.

2.2 KMeans

Nel seguito è riportata l'analisi fatta relativamente all'algoritmo KMeans.

Knee Method per stimare il valore di K Si è utilizzato il Knee Method sul dataset "clienti" per ottenere un'ulteriore stima del valore di K da utilizzare nell'algoritmo KMeans. Le metriche che sono state utilizzate sono SSE, Silhouette e Separation. Nella Figura 8 sono riportati i grafici di queste metriche al variare di K, ossia del numero di cluster.

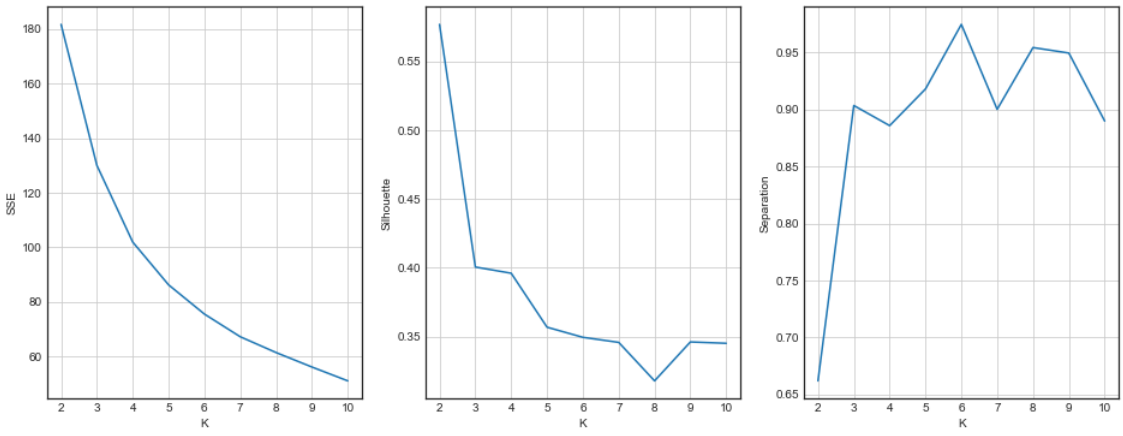


Figura 8: Grafici delle metriche al variare di K.

Dai grafici risulta che i valori migliori di K sono 2 e 3, in quanto la Silhouette ha valore massimo per K = 2, la Separation ha valore minimo per K = 2 e SSE decresce più lentamente per K < 3. Quindi, considerando anche la stima di K ottenuta dall'analisi dei dendrogrammi, è stato scelto di utilizzare il valore K = 3.

KMeans sul dataset clienti In Figura 9 è mostrata una visualizzazione in due dimensioni, rispetto agli attributi Recency e EntropyProducts, del clustering ottenuto. In Tabella 1 sono

riportati i centroidi ottenuti con K = 3. Si osserva che il centroide 3 è quello che ha valore maggiore degli attributi Imax, Frequency, MonetaryValue e EntropyProducts e ha valore minore per l'attributo Recency, potrebbe perciò rappresentare il gruppo di clienti che tendono a effettuare più acquisti e spendere generalmente di più. Dall'altro lato, il centroide 2 è quello che ha valore minore degli attributi Imax e MonetaryValue e ha invece valore maggiore per l'attributo Recency, potrebbe perciò rappresentare il gruppo di clienti che tendono a effettuare pochi acquisti e spendere generalmente meno. Il centroide 1 invece potrebbe essere rappresentativo del gruppo di clienti che hanno effettuato nel sito una spesa né particolarmente alta né particolarmente bassa.

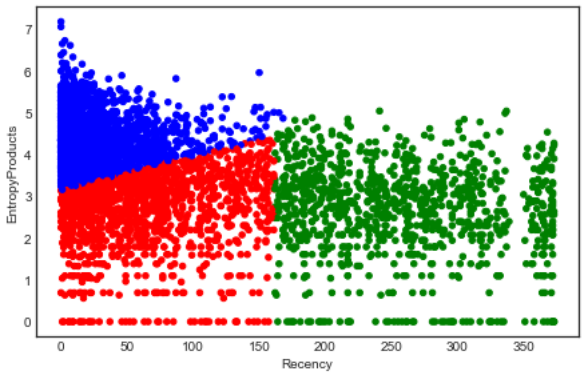


Figura 9: Visualizzazione in due dimensioni rispetto agli attributi Recency e EntropyProducts.

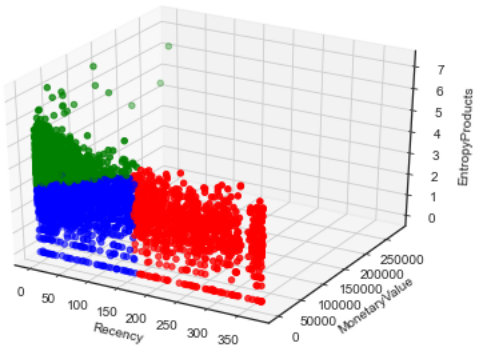
	Imax	Recency	Frequency	MonetaryValue	EntropyProducts
Centroide 1	268.46	65.29	23.29	789.77	2.67
Centroide 2	211.69	257.60	25.92	479.36	2.67
Centroide 3	505.00	28.07	171.35	3416.13	4.38

Tabella 1: Centroidi ottenuti applicando KMeans con K = 3.

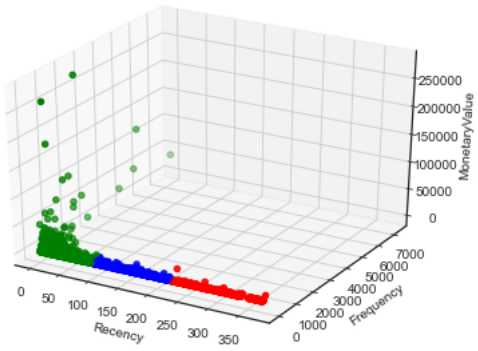
Kmeans sui dataset ridotti È stato applicato l'algoritmo KMeans con K = 3 a dataset ridotti, ovvero dataset in cui ciascun cliente è rappresentato da soli 3 attributi. In questo modo è possibile visualizzare con più chiarezza il clustering ottenuto. Tale approccio, tuttavia, comporta l'esclusione di alcuni attributi e quindi il rinunciare ad alcune informazioni relative ai clienti.

Sono stati creati 10 dataset ridotti, uno per ogni possibile sottoinsieme di tre attributi. Quindi, è stato applicato l'algoritmo KMeans su ognuno di essi. In Figura 10 sono riportati i grafici relativi ai dataset ridotti con cui sono stati ottenuti clustering più significativi:

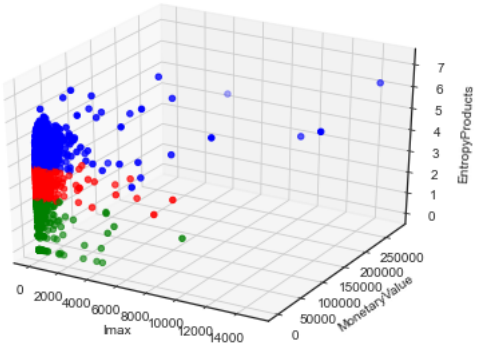
- In Figura 10a è riportato il risultato del clustering sul dataset ridotto formato dagli attributi Recency, MonetaryValue e EntropyProducts.
- In Figura 10b è riportato il risultato del clustering sul dataset ridotto formato dagli attributi Recency, Frequency e MonetaryValue.
- In Figura 10c è riportato il risultato del clustering sul dataset ridotto formato dagli attributi Imax, MonetaryValue e EntropyProducts.



(a) Visualizzazione del clustering rispetto agli attributi Recency, MonetaryValue e EntropyProducts.



(b) Visualizzazione del clustering rispetto agli attributi Recency, Frequency e MonetaryValue.



(c) Visualizzazione del clustering rispetto agli attributi Imax, MonetaryValue e EntropyProducts.

Figura 10: Visualizzazioni del clustering ottenuto su alcuni dei dataset ridotti.

2.3 DBSCAN

Knee method per per stimare il valore di Epsilon Si è scelto di provare i seguenti valori di $K = \{2, 4, 8, 16, 32, 64, 128, 256, 512\}$. Quindi, per ognuno di essi è stato individuato il miglior valore di epsilon usando il knee method. I valori di K e i corrispondenti valori di epsilon sono riportati in Tabella 2.

K	**epsilon**
4	0.1
8	0.1
16	0.1
32	0.2
64	0.25
128	0.25
256	0.3
512	0.4

Tabella 2: Parametri K e epsilon usati nell'algoritmo DBSCAN.

Clustering usando DBSCAN Applicando l'algoritmo DBSCAN al dataset è stato ottenuto come risultato sempre un unico cluster e alcuni punti classificati come rumore, come riportato nella Tabella 3. In conclusione, l'algoritmo DBSCAN non sembra essere particolarmente efficace per questo dataset.

K	epsilon	Dimensione dei Cluster ottenuti
4	0.1	$c_1 = 46$ (rumore), $c_2 = 4278$
8	0.1	$c_1 = 48$ (rumore), $c_2 = 4276$
16	0.1	$c_1 = 49$ (rumore), $c_2 = 4275$
32	0.2	$c_1 = 23$ (rumore), $c_2 = 4301$
64	0.25	$c_1 = 18$ (rumore), $c_2 = 4306$
128	0.25	$c_1 = 19$ (rumore), $c_2 = 4305$
256	0.3	$c_1 = 19$ (rumore), $c_2 = 4305$
512	0.4	$c_1 = 8$ (rumore), $c_2 = 4316$

Tabella 3: Cluster ottenuti applicando l'algoritmo DBSCAN.

2.4 XMeans

Infine, è stato applicato al dataset l'algoritmo XMeans. Impostando come numero massimo di cluster il valore 3, sono stati ottenuti 3 cluster con centroidi simili a quelli ottenuti precedentemente applicando l'algoritmo KMeans con K = 3. In Tabella 4 sono riportati i tre centroidi.

	Imax	Recency	Frequency	MonetaryValue	EntropyProducts
Centroide 1	272.88	136.44	30.42	634.91	2.74
Centroide 2	202.32	290.14	22.27	427.73	2.53
Centroide 3	427.72	29.72	125.32	2649.48	3.87

Tabella 4: Centroidi ottenuti applicando XMeans con numero massimo di cluster pari a 3.

Anche in questo caso si ha che il centroide 3 potrebbe essere rappresentativo della classe di clienti che tende a fare più acquisti e spendere di più, mentre il centroide 2 potrebbe rappresentare la classe di clienti che tende a fare pochi acquisti e spendere poco. Infine, il centroide 1 potrebbe rappresentare la classe di clienti che ha effettuato un numero di acquisti e una spesa totale né particolarmente alta né particolarmente bassa.

Si è poi provato ad applicare l'algoritmo XMeans impostando un numero massimo di cluster maggiore di 3. In questi casi l'algoritmo ha restituito un numero di cluster uguale al numero

massimo di cluster impostato, ad esempio impostando come numero massimo di cluster il valore 7, sono stati ottenuti 7 cluster.

Tuttavia, considerando anche l'analisi effettuata per gli algoritmi di clustering applicati in precedenza (Kmeans e Clustering Gerarchico), si può ipotizzare di suddividere i clienti rappresentati nel dataset in tre gruppi differenti.

3 Classificazione

Prima di allenare i modelli per effettuare la classificazione degli utenti, gli attributi del dataset "clienti" il cui valore è dipendente dalla durata del periodo di osservazione sono stati modificati, così che non dipendano più dal periodo di osservazione. In questo modo il modello di classificazione che verrà alla fine selezionato potrà essere usato per classificare nuovi utenti il cui periodo di osservazione può variare. In particolare i seguenti attributi sono stati modificati:

- Per ciascun utente, il valore dell'attributo *MonetaryValue* è stato diviso per 373, ovvero il numero di giorni del periodo di osservazione. Tale valore rappresenta la spesa media giornaliera dell'utente.
- Per ciascun utente, il valore dell'attributo *Frequency* è stato diviso per 373, ovvero il numero di giorni del periodo di osservazione. Tale valore rappresenta il numero medio di carrelli effettuati al giorno dall'utente.

Gli attributi *Imax*, *Recency* e *EntropyProducts* non sono invece stati modificati.

Inoltre, è stato creato un secondo dataset che contiene gli stessi attributi del dataset "clienti" e un attributo aggiuntivo chiamato *SMax*. Per ogni utente, il valore di questo attributo rappresenta la spesa totale massima effettuata dall'utente in un solo carrello. Ad esempio, se un cliente ha effettuato un carrello spendendo in totale 10 €, un altro carrello spendendo in totale 5 €, infine un altro carrello spendendo in totale 8 €, il valore dell'attributo *SMax* a lui associato è 10. Per ogni classificatore, sono state allenate e testate due versioni: la prima sul dataset senza l'attributo *SMax* e la seconda sul dataset con l'attributo *SMax*. Quindi, sono stati confrontati i risultati ottenuti.

3.1 Suddivisione dei clienti in classi

I clienti sono stati suddivisi in tre classi sulla base della spesa media giornaliera. I valori di soglia sono stati scelti osservando la distribuzione di tale attributo nel dataset. Si nota che il 25% dei clienti ha effettuato una spesa media giornaliera minore o uguale di 0.80 €, il 75% dei clienti ha effettuato una spesa media giornaliera minore o uguale di 4.29 €. Quindi, sono stati utilizzati questi valori per suddividere i clienti in classi nel seguente modo:

- **low-spending**: un cliente appartiene a questa classe se ha effettuato una spesa media giornaliera minore o uguale 0.80 €.
- **medium-spending**: un cliente appartiene a questa classe se ha effettuato una spesa media giornaliera compresa tra 0.80 € e 4.29 €.
- **high-spending**: un cliente appartiene a questa classe se ha effettuato una spesa media giornaliera superiore a 4.29 €.

Applicando la suddivisione in classi descritta ai due dataset, si ottiene che 1075 clienti appartengono alla classe low-spending, 2167 appartengono alla classe medium-spending e 1082 appartengono alla classe high-spending.

L'attributo che rappresenta la spesa media giornaliera è stato rimosso dall'insieme degli attributi usati per effettuare la classificazione, in quanto è stato utilizzato per la definizione delle classi.

3.2 Creazione del Training Set e Test Set

I due dataset sono stati suddivisi ciascuno in training set (70% dei record) e test set (30% dei record). Il training set e il test set hanno circa la stessa proporzione di elementi per classe del dataset originale. Gli algoritmi testati per la classificazione sono:

- Decision Tree, applicato sui dataset non standardizzati.
- K-nearest neighbors (KNN), applicato sui dataset standardizzati.
- Naive Bayesian Classifier, applicato sui dataset non standardizzati.
- Random Forest, applicato sui dataset non standardizzati.
- Support vector machine (SVM), applicato sui dataset standardizzati.

Nel caso del Decision Tree, Naive Bayesian Classifier e Random Forest si è scelto di applicare i modelli sui dataset non standardizzati in quanto tali modelli non necessitano di una standardizzazione preliminare dei dati.

Non è stata considerata la classificazione per mezzo di reti neurali dal momento che questi sono più adatti per la classificazione di serie temporali, immagini o testi, e necessitano di dataset di grandi dimensioni. Non è stata considerata neanche la classificazione basata su regole poiché in questo caso gli utenti sono stati suddivisi in tre diverse classi ma l'algoritmo RIPPER è utilizzabile solamente per la costruzione di classificatori binari.

3.3 Risultati ottenuti

Per scegliere gli iperparametri da utilizzare nei diversi algoritmi è stata applicata sul training set la Grid Search in congiunzione con la K-fold Cross Validation, con K=5, partendo da un insieme predefinito di iperparametri. In particolare, sono stati scelti gli iperparametri che massimizzano l'accuratezza sul training set.

Si è notato che considerando per ciascun utente anche l'attributo *SMax* si ottengono risultati generalmente migliori, ossia valori di accuratezza, precisione e recall più alti. Quindi, in seguito sono riportati solo i risultati dei modelli allenati sul dataset con l'attributo *SMax*.

In Tabella 5 sono riportati, per ciascun modello, i valori degli iperparametri utilizzati.

Modello	Iperparametro	Valore
Decision Tree		
	criterion	entropy
	max_depth	12
	min_samples_leaf	6
	min_samples_split	6
	splitter	best
KNN		
	algorithm	ball_tree
	metric	Minkowski
	n_neighbors	10
	p	1 (dist. Manhattan)
	weights	distance
Naive Bayesian Classifier		
	Nessun iperparametro	Nessun iperparametro
Random Forest		
	criterion	entropy
	max_depth	16
	min_samples_leaf	2
	min_samples_split	4
	n_estimators	100
SVM		
	C	100
	gamma	auto
	kernel	RBF

Tabella 5: Iperparametri utilizzati per ciascun modello.

Una volta selezionati gli iperparametri, ogni modello è stato allenato sul training set (standardizzato o meno a seconda del modello). Il classificatore ottenuto è stato quindi applicato sia al training set che al test set e sono state calcolate le seguenti metriche: accuratezza, precisione, recall e F1-score. In Tabella 6 sono riportati i risultati ottenuti.

	Training Set				**Test Set**			
Modello	**Accuratezza**	**Precisione**	**Recall**	**F1-Score**	**Accuratezza**	**Precisione**	**Recall**	**F1-Score**
Decision Tree	0.93	0.93	0.93	0.93	0.85	0.85	0.84	0.84
KNN	1.00	1.00	1.00	1.00	0.80	0.81	0.80	0.80
Naive Bayesian Classifier	0.78	0.79	0.78	0.78	0.77	0.79	0.76	0.76
Random Forest	0.99	0.99	0.99	0.99	0.87	0.87	0.87	0.87
SVM	0.90	0.90	0.90	0.90	0.88	0.88	0.89	0.88

Tabella 6: Risultati ottenuti applicando i diversi modelli al training e al test set.

3.3.1 Modello migliore

I modelli migliori risultano essere Random Forest e SVM, poiché con questi modelli sono stati ottenuti valori maggiori di accuratezza, precisione, recall e F1-Score. Tuttavia, tra i due modelli è preferibile SVM in quanto si ha una minor differenza tra i valori delle metriche ottenuti sul training set e sul test set.

In Tabella 7 sono riportati i valori delle metriche, relativamente alle classi, ottenuti applicando il modello SVM al test set.

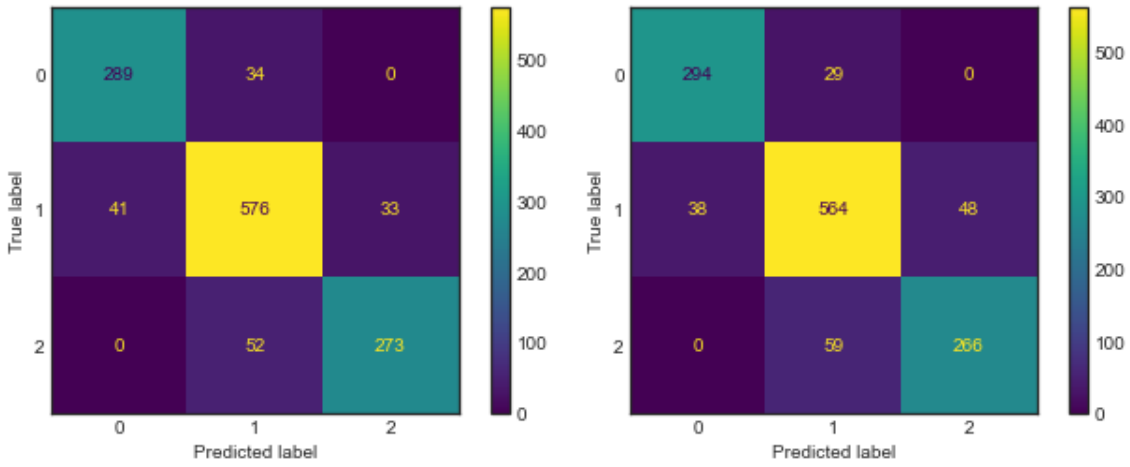
Classe	Precisione	Recall	F1-score	Support
low-spending	0.88	0.89	0.89	323
medium-spending	0.87	0.89	0.88	650
high-spending	0.89	0.84	0.87	325

Tabella 7: Valori delle metriche ottenuti applicando il modello SVM al test set.

In Tabella 8 sono riportati i valori delle metriche, relativamente alle classi, ottenuti applicando il modello Random Forest al test set.

Classe	Precisione	Recall	F1-score	Support
low-spending	0.89	0.91	0.90	323
medium-spending	0.87	0.87	0.87	650
high-spending	0.85	0.82	0.83	325

Tabella 8: Valori delle metriche ottenuti applicando il modello Random Forest al test set.



(a) Confusion Matrix relativa a SVM.

(b) Confusion Matrix relativa a Random Forest.

Figura 11: Confusion Matrix relative ai modelli migliori.

In Figura 11 sono rappresentate le confusion matrix ottenute applicando i modelli al test set. Si nota che la maggior parte degli errori sono dovuti a clienti della classe medium-spending (classe 1) classificati come clienti della classe low-spending (classe 0) o high-spending (classe 2), oppure viceversa. Tuttavia, succede raramente che un cliente di classe low-spending venga classificato come un cliente di classe high-spending, o viceversa.

È stato quindi scelto SVM come modello migliore. Infine, per questo modello è stata effettuata sull'intero dataset standardizzato una K-fold cross-validation finale, con K = 5, per ottenere una stima migliore dell'errore di generalizzazione. Si è ottenuto un'accuratezza media sul training set pari all' 89% e un'accuratezza media sul test set dell' 88%.

SVM risulta essere il modello migliore anche utilizzando il dataset che non contiene l'attributo *SMax*. In questo caso tuttavia si ottengono valori di accuratezza, precisione, recall e F1-score minori, rispetto a quando invece l'attributo *SMax* viene considerato. Come esempio sono riportati

in Tabella 9 i valori delle metriche, relativamente alle classi, ottenuti applicando il modello SVM al test set.

Classe	Precisione	Recall	F1-score	Support
low-spending	0.79	0.74	0.76	323
medium-spending	0.79	0.84	0.81	650
high-spending	0.87	0.80	0.83	325

Tabella 9: Valori delle metriche ottenuti applicando il modello SVM al test set del dataset che non contiene l'attributo *SMax*.

4 Sequential Pattern Mining

Per individuare i "sequential patterns" frequenti presenti nel dataset, è stato utilizzato il modulo python "gsp.py", fornito durante il corso, che implementa l'algoritmo per effettuare il *generalized sequential pattern mining*. Per fare ciò, ciascun cliente è stato rappresentato con la rispettiva sequenza di carrelli, ogni carrello è un insieme di prodotti.

Dopo aver effettuato delle prove preliminari, è stato deciso di applicare l'algoritmo a dei sottoinsiemi di clienti, in quanto il tempo richiesto per eseguire l'algoritmo sull'intero dataset è risultato eccessivo (più di 10 ore sui computer a nostra disposizione). In seguito sono descritti i criteri con cui sono stati costruiti i sottoinsiemi di clienti:

- *Sottoinsieme 1*: sono stati considerati solo gli acquisti effettuati durante il mese di novembre, in quanto novembre è il mese in cui sono stati effettuati più acquisti. Quindi, il primo sottoinsieme di sequenze è stato estratto a partire da questo dataset ridotto. Il sottoinsieme contiene 1656 sequenze.
- *Sottoinsieme 2*: sono state considerate solo le sequenze relative a clienti che hanno effettuato, durante l'intero periodo di osservazione, un numero totale di carrelli maggiore di 4 e minore di 20. Il sottoinsieme contiene 1391 sequenze.

4.1 Risultati relativi al sottoinsieme 1

Applicando l'algoritmo di generalized sequential pattern mining, con support minimo = 30, sono state individuate solamente quattro sottosequenze composte da due elementi (insiemi di prodotti), tutte queste quattro sottosequenze contengono due oggetti. Le altre sottosequenze individuate contengono un solo elemento, che contiene un oggetto oppure due oggetti.

In Tabella 10 sono riportate le quattro sottosequenze trovate composte da due elementi.

Sottosequenza	Support
<{RABBIT NIGHT LIGHT}, {RABBIT NIGHT LIGHT}>	78/1656
<{PAPER CHAIN KIT 50'S CHRISTMAS}, {PAPER CHAIN KIT 50'S CHRISTMAS}>	46/1656
<{POPCORN HOLDER}, {POPCORN HOLDER}>	36/1656
<{PAPER CHAIN KIT VINTAGE CHRISTMAS}, {PAPER CHAIN KIT 50'S CHRISTMAS}>	30/1656

Tabella 10: Sottosequenze composte da due elementi.

Si nota che queste sottosequenze contengono prodotti festivi, come decorazioni natalizie e contenitori per popcorn. Si può ipotizzare che questo è dovuto al fatto che sono stati considerati solo

gli acquisti relativi al mese di novembre: molti clienti in questo periodo potrebbero aver acquistato prodotti in previsione delle festività natalizie. Si nota inoltre che diversi clienti hanno riacquistato, nell'arco del mese di novembre, lo stesso prodotto. Infatti, tre delle quattro sottosequenze contengono lo stesso prodotto ripetuto (in realtà, anche nella quarta sottosequenza, i prodotti acquistati sono varianti diverse della stessa tipologia di prodotto).

In Tabella 11 sono riportate le cinque sottosequenze con support più alto che contengono un solo elemento composto da due prodotti.

Sottosequenza	**Support**
<{PAPER CHAIN KIT 50'S CHRISTMAS, PAPER CHAIN KIT VINTAGE CHRISTMAS}>	133/1656
<{WOODEN HEART CHRISTMAS SCANDINAVIAN, WOODEN STAR CHRISTMAS SCANDINAVIAN}>	131/1656
<{WOODEN STAR CHRISTMAS SCANDINAVIAN, WOODEN TREE CHRISTMAS SCANDINAVIAN}>	91/1656
<{ALARM CLOCK BAKELIKE GREEN, ALARM CLOCK BAKELIKE RED}>	90/1656
<{CHOCOLATE HOT WATER BOTTLE, HOT WATER BOTTLE TEA AND SYMPATHY}>	88/1656

Tabella 11: Sottosequenze composte da un solo elemento e due oggetti, con support maggiore.

Anche in questo caso si nota che le sottosequenze contengono diversi prodotti relativi alle festività natalizie. Inoltre, si nota che diversi clienti hanno acquistato, nello stesso carrello, varianti diverse di uno stesso prodotto. Infatti, nel periodo natalizio, è comune acquistare tipologie di decorazioni diverse, ad esempio diversi tipi di palline per l'albero di Natale (la seconda sottosequenza in Tabella 11 è appunto un esempio di questo fatto).

Figura 12: WOODEN HEART CHRISTMAS SCANDINAVIAN (Presa da: Google Images).

4.2 Risultati relativi al sottoinsieme 2

Applicando l'algoritmo di generalized sequential pattern mining al sottoinsieme 2, con support minimo = 60, sono state individuate in totale 5004 sottosequenze. Nel seguito sono riportate solo alcune sottosequenze ritenute più significative, ovvero sottosequenze composte da un numero maggiore di elementi e che hanno support abbastanza alto.

In Tabella 12 sono riportate le cinque sottosequenze composte da quattro elementi che hanno support più alto (non sono state individuate sottosequenze con più di quattro elementi).

Sottosequenza	**Support**
<{WHITE HANGING HEART T-LIGHT HOLDER}, {WHITE HANGING HEART T-LIGHT HOLDER}, {WHITE HANGING HEART T-LIGHT HOLDER}, {WHITE HANGING HEART T-LIGHT HOLDER} >	107/1391
<{JUMBO BAG RED RETROSPOT}, {JUMBO BAG RED RETROSPOT}, {JUMBO BAG RED RETROSPOT}, {JUMBO BAG RED RETROSPOT} >	83/1391
<{JUMBO BAG RED RETROSPOT}, {JUMBO BAG RED RETROSPOT}, {JUMBO BAG DOILEY PATTERNS}, {JUMBO BAG RED RETROSPOT} >	69/1391
<{JUMBO BAG RED RETROSPOT}, {JUMBO BAG DOILEY PATTERNS}, {JUMBO BAG RED RETROSPOT}, {JUMBO BAG DOILEY PATTERNS} >	68/1391
<{ASSORTED COLOUR BIRD ORNAMENT}, {ASSORTED COLOUR BIRD ORNAMENT}, {ASSORTED COLOUR BIRD ORNAMENT}, {ASSORTED COLOUR BIRD ORNAMENT} >	67/1391

Tabella 12: Sottosequenze con quattro elementi e support più alto.

Si osserva che "JUMBO BAG RED RETROSPOT" e "ASSORTED COLOUR BIRD ORNAMENT" sono rispettivamente il terzo e il quarto prodotto più acquistato durante il periodo di

osservazione, come riportato in Figura 1.

Si può ipotizzare che, essendo i prodotti "JUMBO BAG RED RETROSPOT" e "JUMBO BAG DOILEY PATTERNS" delle borse per contenere vestiti, bucato, giocattoli ecc... un cliente potrebbe acquistare questa tipologia di prodotto più volte nel caso in cui, ad esempio, una borsa si rovini oppure nel caso in cui, soddisfatto del primo acquisto, necessiti in seguito di altri contenitori.

Figura 13: JUMBO BAG (Presa da: Google Images).

In Tabella 13 sono riportate alcune delle sottosequenze con tre elementi individuate. Sono state riportate solo queste sottosequenze perché si è notato che comunque la maggior parte delle sottosequenze contiene sempre le stesse tipologie di prodotti ("LUNCH BAG", "JUMBO BAG"...). Questi sono anche prodotti che spesso compaiono nella lista dei 10 prodotti più acquistati. Si è inoltre notato che ciascuna sottosequenza contiene solitamente oggetti della stessa tipologia, come si osserva anche dalle sottosequenze riportate nelle tabelle. Questo indica che diversi clienti hanno riacquistato più volte, nell'arco del periodo di osservazione di circa un anno, lo stesso prodotto o una sua variante. Si nota infine che questi sono prodotti economici e di uso comune, è quindi normale che possano essere acquistati più volte: è più inusuale acquistare più volte, nell'arco di un anno, prodotti costosi (ad esempio televisori, elettrodomestici, ecc.).

Sottosequenza	**Support**
< {LUNCH BAG RED RETROSPOT, LUNCH BAG BLACK SKULL}, {LUNCH BAG SUKI DESIGN}, {LUNCH BAG BLACK SKULL} >	63/1391
< {LUNCH BAG RED RETROSPOT, LUNCH BAG CARS BLUE}, {LUNCH BAG CARS BLUE}, {LUNCH BAG RED RETROSPOT} >	63/1391
< {LUNCH BAG RED RETROSPOT, LUNCH BAG PINK POLKADOT}, {LUNCH BAG PINK POLKADOT}, {LUNCH BAG RED RETROSPOT} >	63/1391
< {LUNCH BAG RED RETROSPOT}, {LUNCH BAG RED RETROSPOT}, {LUNCH BAG RED RETROSPOT} >	122/1391
< {REGENCY CAKESTAND 3 TIER}, {REGENCY CAKESTAND 3 TIER}, {REGENCY CAKESTAND 3 TIER} >	117/1391
< {PARTY BUNTING}, {PARTY BUNTING}, {PARTY BUNTING} >	99/1391

Tabella 13: Esempi di sottosequenze con tre elementi.