Metodi di clustering applicati in ambito e-commerce

Riccardo Confalonieri
CdLM Data Science
matr. 830404

Rebecca Picarelli
CdLM Data Science
matr. 834286

Silvia Ranieri
CdLM Data Science

Progetto di Machine Learning
Appello del 18 Gennaio 2020 - TEAM 04

Abstract

La domanda di ricerca alla base di questo progetto è: E' possibile suddividere i clienti di un sito di e-commerce sulla base dei loro acquisti al fine di migliorare le comunicazioni di marketing? Tutti i siti di e-commerce, più e meno famosi, inviano continuamente email e promozioni ai clienti al fine di invogliarli ad acquistare e per cercare di non perderli a favore di altri siti. Infatti, è stato dimostrato che mandare mail indifferenziate non è di aiuto: soltanto in pochi clienti le aprirebbero; inoltre, tra questi, un numero ancor minore si deciderebbe ad effettuare davvero degli acquisti. Per questo motivo con questo report ci si prefigge come scopo di riportare i risultati della clusterizzazione dei clienti di un e-commerce, attraverso il modello Recency-Frequency-Monetary (RFM), suddivisi in diversi gruppi in modo da poter ipoteticamente inviare comunicazioni di marketing più mirate e significative, rendendo incisiva ed efficace la comunicazione tra e-commerce e clienti.

Indice

Note tecniche

Per svolgere il lavoro si è utilizzata la piattaforma Knime [6] con l'ausilio dei linguaggi di programmazione Python [9] ed R [8] resi disponibili dalla piattaforma attraverso l'utilizzo di nodi snippet specifici.

1 Introduzione

Il fenomeno dell'e-commerce, in realtà presente fin dagli anni '60, è solo a partire dalla metà degli anni novanta del secolo scorso che ha iniziato a diffondersi sempre di più in tutto il mondo, diventando una parte pressoché indispensabile per la vita di moltissime persone. Tutto ciò è stato facilitato dal crescente livello di digitalizzazione della realtà odierna, senza dimenticare, ovviamente, il recente impatto dovuto all'emergenza Covid19, che ha portato grossi benefici e sviluppi in questo settore. Lo scopo di questo lavoro è quello di segmentare i clienti dell'e-commerce utilizzando la tecnica del RFM, ovvero uno dei modelli più noti per l'analisi

del marketing, che cerca di suddividere i clienti in base ad i tre seguenti parametri:

- Recency **(R)**: misura il tempo trascorso dall'ultimo acquisto di un certo cliente.
- Frequency **(F)**: riflette il numero di acquisti ripetuti da un cliente.
- Monetary **(M)**: indica l'ammontare speso dal cliente.

Questa segmentazione offrirebbe la possibilità di inviare offerte personalizzate in base al particolare tipo di cliente. Diversi studi negli anni hanno infatti dimostrato che l'invio di email e/o offerte senza alcuna differenziazione non funziona: in particolare i clienti abituali, insieme a quelli che spendono molto di più, sono più ricettivi nei confronti di possibili offerte.

2 Presentazione del dataset

Il dataset, reperibile sulle piattaforme Kaggle [4] e The UCI Machine Learning Repository [7], si riferisce ai movimenti commerciali realizzati nel periodo 01/12/2010 al 09/12/2011 relativi ad un'azienda realmente esistente con sede nel Regno Unito. Quest'ultima risulta essere adibita alla vendita online di svariati prodotti e articoli da regalo su scala internazionale, sia all'ingrosso che al dettaglio.
Il dataset nella forma originale risulta essere costituito da 541.909 righe, ogni riga corrisponde ad un singolo prodotto acquistato da un cliente. Il dataset infatti non risulta aggregato per fattura ma bensì ogni singolo ordine è scomposto su tante righe quanti sono i prodotti acquistati.
Ogni riga è costituita dalle seguenti 8 colonne:

- InvoiceNo (categorica-nominale): Numero univoco di fattura associato a ciascuna transazione
- StockCode (categorica-nominale): codice associato a ogni prodotto
- Description (categorica-nominale): Nome di ogni prodotto
- Quantity (numerica-val.interi): Quantità prodotto per ogni transazione
- InvoiceDate (categorica-ordinale): Data e tempo fattura, giorno di ogni transazione
- UnitPrice (numerica-val.decimali): Prezzo unitario in sterline di ogni prodotto
- CustomerID (categorica-nominale): Codice univoco associato a ogni cliente
- Country (categorica-nominale): Stato in cui risiede il cliente

3 Esplorazione dei dati

Inizialmente, è stato opportuno dedicare alcuni passaggi alla pulizia del dataset. Come prima cosa, infatti, è stata riscontrata la presenza di 5.268 righe duplicate e di 135.080 valori mancanti riferiti alla variabile CustomerID, per le quali si è scelto di rimuovere completamente la riga non essendo associabili a nessun cliente e quindi non utili ai fini dell'analisi. In aggiunta, si può precisare che in numerosi casi in cui il codice del cliente risultava mancante, il prezzo unitario compariva con valore nullo, trattandosi di articoli danneggiati/non vendibili o in rari casi di possibili omaggi ai clienti.
Successivamente, è stata notata la presenza di 8872 casi in cui la variabile InvoiceNo compare con il prefisso "C" ad indicare che la transazione fa riferimento ad ordini cancellati (in queste istanze la variabile Quantity compare con valori negativi). Tuttavia prima di cancellare queste righe dal nostro dataset abbiamo deciso di rintracciare il corrispondente ordine positivo effettuato dal cliente cercando di rintracciare la riga con lo stesso customerID, stockCode, data di acquisto precedente a quella di cancellazione e con valore di Quantity opposto, considerando anche il caso in cui non venissero cancellati totalmente tutti i prodotti acquistati ma soltanto parzialmente.
Non sempre, però, tale abbinamento è stato possibile: talvolta avvengono restituzioni in denaro inserendo un valore manualmente (alcune volte parziale), oppure probabilmente rilasciando uno sconto, ma è anche plausibile che l'acquisto del prodotto non rientri nell'arco temporale coperto dal dataset. In questi casi dunque non è stato possibile rimuovere anche la controparte di ordine effettuata e si è proceduto ad eliminare soltanto la riga con quantity negativa.
Inoltre, ci si è soffermati sui valori assunti dalla variabile StockCode che in alcuni casi non si riferiva ad un codice prodotto ma bensì ad altre operazioni, in questi casi lo Stockcode risultava testuale invece che numerico e si è proceduto ad eliminare tali righe.
Si elencano di seguito le diciture riportate nel dataset affiancate dal contenuto della variabile Description:

- BANK CHARGES: "Bank Charges"
- C2: "Carriage"
- CRUK: "CRUK commission"

- D: “Discount”
- DOT: “Dotcom postage”
- M: “Manual”
- PADS: “Pads to match all cushions”
- POST: “Postage”

Come si può notare queste righe, associate a quantità negative non si riferiscono a spese realmente effettuate dal cliente e quindi abbiamo deciso di eliminarle. Nel dataset iniziale erano presenti anche altre description non relative a prodotti che però sono state eliminante durante la rimozione delle righe contenenti missing values.
Una volta eliminate tutte queste casistiche e mantenendo, quindi, solo le transazioni ritenute valide, il numero di righe nel dataset si è ridotto a 388.588. Inoltre, basandosi sui dati ripuliti, è stato anche possibile individuare ulteriori tendenze e aspetti relativi alle transazioni effettuando dei raggruppamenti per variabile: ad esempio, è possibile affermare che, escludendo il Regno Unito risultato in netto vantaggio con quasi 7 milioni, le cifre totali spese durante l'anno si concentrano in ordine decrescente nei Paesi Bassi (283.479,54), Irlanda, Germania, Francia e Australia (136.954,75), mentre il numero minore (inferiore a 1000) è associato ad Arabia Saudita, Bahrein e Repubblica Ceca. Tali posizioni, inoltre, rimangono pressoché invariate considerando anche il numero di spese effettuate, sempre nello stesso arco temporale. Sulla base di quest'ultimo, comunque, si può aggiungere che la quantità maggiore di spese è avvenuta nel mese di novembre (2634), seguito da ottobre (1889) e settembre. Gennaio e febbraio risultano essere quelli con i valori minori, forse perché ormai il periodo delle festività natalizie si è concluso; e, come nel caso precedente, anche le cifre complessive spese seguono tale andamento. Invece, valutando il periodo su base settimanale, si evince come prima cosa che non risultano esserci transazioni al sabato, e che da lunedì fino a giovedì il numero di queste e l'ammontare di denaro speso complessivamente sono crescenti, per poi ridursi chiaramente sia alla domenica che al venerdì. Infine, è stato possibile osservare che, ad esempio, tra i 3645 prodotti disponibili (sulla base dei diversi StockCode) circa 240 vengono acquistati in tutto 5 volte o con una frequenza addirittura minore. Il prodotto più venduto in assoluto è un modellino stilizzato da assemblare di un aeroplano della seconda guerra mondiale, comprato 53119 volte principalmente nel Regno Unito, Svezia, Irlanda e Giappone, soprattutto nei mesi di aprile e ottobre.

4 Feature selection e feature creation

Per riuscire a svolgere l'analisi di classificazione dei clienti che ci siamo proposti in fase iniziale abbiamo calcolato tre nuove variabili a partire da quelle a nostra disposizione che identificano i valori di *recency, frequency e monetary* (RFM). Per ottenere questi nuovi attributi è stato necessario effettuare alcune operazioni sul dataset a disposizione.
In prima istanza, dato che ogni riga del dataset riportava il prodotto acquistato e il suo prezzo unitario, è stato calcolato il prezzo totale dell'acquisto moltiplicando questi due valori. In questo modo sarà poi possibile ottenere un prezzo totale di acquisto per ogni ordine evaso. Inoltre si è convertita la data di acquisto dal formato stringa al formato Date in modo da poterla manipolare correttamente.
Una volta calcolato il prezzo totale di vendita di ogni item in base alla quantità acquistata è stato affrontato il problema che nel dataset gli ordini erano ancora suddiviso su più righe e dunque si è proceduto ad aggregare i record a nostra disposizione secondo la variabile CustomerID. Durante questo processo di aggregazione vengono calcolati i seguenti attributi per ogni cliente:

- *M:* ottenuto sommando la spesa effettuata da ogni singolo cliente per tutti gli ordini effettuati.
- *F:* calcolato contando quante ricevute con identificativo univoco sono state rilasciate ad ogni cliente.
- Data di ultimo acquisto

Al termine della fase di aggregazione si hanno dunque a disposizione due delle tre variabili a necessarie per svolgere la fase di classificazione dei clienti. Per ottenere l'ultima variabile, R, è stata calcolata la differenza in numero di giorni intercorsi tra il 10-12-2011 e la data di ultimo acquisto di ogni cliente. La scelta di una specifica data ultima da cui iniziare il conteggio è dovuta al fatto che nel dataset erano presenti gli acquisti effettuati tra il 1-12-2010 e il 9-12-2011, si ha dunque optato per assegnare il valore minimo 1 a tutti gli utenti che avessero effettuato un acquisto nell'ultima data disponibile.
Analizzando le tre variabili ottenute, è stato possibile notare che il range di valori assunto era molto ampio e variegato, inoltre la loro distribuzione risultava asimmetrica. Solitamente in letteratura tale situazione è affrontata discretizzando le variabili nel range $[1-5]$ per poi concatenarle e ottenere

così un unico campo, RFM, che racchiuda in se le informazioni del cliente. Per farlo è però necessario stabilire a priori una scala per tramutare i valori continui delle variabili nel nuovo range, questo passaggio è determinante e richiede una buona conoscenza del dominio di applicazione ed è molto soggettivo. Dato che la codifica scelta può influenzare notevolmente il risultato degli algoritmi di clustering abbiamo deciso di non procedere in questa direzione ma optare per trasformazioni matematiche delle variabili mantenendo i tre valori separati.
In prima istanza abbiamo dunque effettuato una trasformazione logaritmica delle variabili applicando la funzione *ln* e, successivamente, abbiamo standardizzato i valori ottenuti dalla trasformazione matematica applicando la tecnica z-score in modo che le tre variabili avessero la stessa importanza durante la fase di clusterizzazione. Inoltre questo passaggio ha permesso di ridurre l'influenza degli outliers presenti nel nostro dataset che, come già detto nei capitoli precedenti, abbiamo deciso di non rimuovere in quanto ritenuti valori "validi" perché realmente osservati, ma che comunque, se trascurati, potrebbero influire in negativo su alcuni algoritmi di clustering, come il K-means.
Avendo utilizzato queste trasformazioni si è però ottenuto nel caso specifico della recency una scala opposta, ovvero coloro che hanno effettuato più recentemente un acquisto avranno un valore di R minore, questo dettaglio risulta importante e dovrà essere adeguatamente osservato quando si andranno ad analizzare i cluster ottenuti.

5 Clustering

Per classificare i clienti secondo i tre valori appena calcolati si procede con la Cluster Analysis applicando 3 differenti algoritmi: *K-means, fuzzy e gerarchico*. Questi algoritmi sono stati tutti applicati con l'intento di suddividere il dataset in tre grosse categorie di clienti:

- Top clienti: ovvero i clienti più fedeli che generano la maggior parte degli introiti
- Clienti normali: quei clienti che tendono a tornare abbastanza frequentemente e che non vanno persi
- Clienti persi: ovvero quei clienti che non effettuano più transazioni

5.1 K-means

Questo algoritmo segue una procedura iterativa che cerca di partizionare il dataset in k sottogruppi distinti e non sovrapposti. L'obiettivo dell'algoritmo è minimizzare la distanza all'interno del cluster e al contempo creare cluster il più diversi possibile. Nel nostro particolare caso, dato che i dati erano continui si ha deciso di selezionare la misura di distanza euclidea.
Dato che questo algoritmo è sensibile all'ordinamento dei record abbiamo effettuato un sorting del dataset prima di effettuare la clusterizzazione, in modo da ridurne l'effetto, così facendo abbiamo ottenuto i seguenti risultati:

R	F	M	#record
0.69	-0,84	-0.80	1869
-1.27	1.53	1.37	764
-0.19	0.25	0.27	1691

Tabella 1: cluster k-means

Osservando i valori riportati nella tabella 1 si può notare come al primo cluster siano stati assegnati i clienti che spendono meno di tutti: hanno acquistato poche volte ed è molto che non effettuano nuovi acquisti[1]; dunque i *clienti persi*.
Al secondo cluster vengono invece assegnati quei clienti con valori molto alti in tutti e tre gli attributi, *clienti top*. Infine all'ultimo cluster appartengono tutti i clienti cosiddetti *normali*.
Questa suddivisione risulta più evidente dalla seguente rappresentazione grafica, ottenuta attraverso uno scatterplot 3D, in cui vengono riportati nei vari assi tutti e tre i valori.

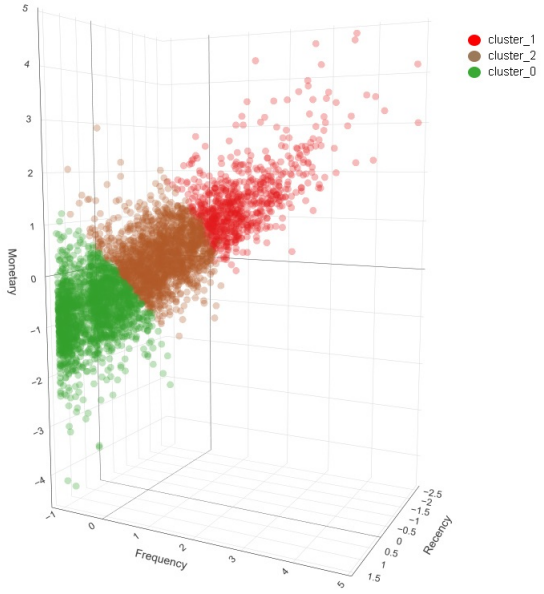

Figura 1: K-means 3D scatterplot

[1] I valori di recency alti sono assegnati ai clienti che non acquistano da più tempo

La classificazione così ottenuta non è però perfetta, vi sono alcuni clienti infatti che vengono erroneamente assegnati ad un cluster piuttosto che ad un altro in quanto non vi è una netta separazione tra i valori che garantirebbe di ottenere cluster meglio separati.

5.2 Fuzzy C-Means

Dato che come appena descritto la suddivisione non risulta così netta come auspicabile, è stato applicato anche il clustering di tipo fuzzy. Questa tipologia, molto simile al k-means , differisce da quest'ultimo sostanzialmente per il fatto che un determinato punto (i.e cliente) potrebbe appartenere a più di un cluster. Il risultato dell'algoritmo infatti oltre a restituire il cluster vincente, ovvero quello a cui con maggior probabilità appartiene un determinato cliente, riporta anche un valore di membership. Esso indica il grado con cui ogni punto appartiene ad un cluster. Infine, si precisa che come nel caso dell'algoritmo k-means è stata utilizzata la misura di distanza euclidea.
Soffermandosi un attimo sul risultato del cluster vincitore, è emerso che è lo stesso del k-means. Tuttavia, analizzandone anche il valore di membership è possibile ampliare l' analisi in modo da non escludere del tutto - da eventuali mailing list o promozioni - quei clienti il cui grado di membership non è completamente sbilanciato a favore di un cluster piuttosto che un altro.

5.3 Clustering gerarchico

Per questo clustering è stata adottata la strategia agglomerativa, resa disponibile da Knime, che è di tipo "bottom up". In partenza, dunque, i record sono visti tutti come cluster differenti e man mano che l'algoritmo avanza, questi vengono raggruppati due a due secondo una particolare metrica. Anche in questo caso ci si è basati sulla misura euclidea, già utilizzata per gli altri tipi di clustering utilizzando il criterio di collegamento denominato "average linkage".
La scelta del criterio non è stata casuale: per deciderlo è stata calcolata la *correlazione cofenetica* ottenendo i seguenti valori:

	Cophenetic value
Average	0.692
Complete	0.598
Ward	0.566
Single	0.501

Tabella 2: Valori cofenetici

Con riferimento alla Tabella 2 abbiamo deciso di utilizzare average come criterio in quanto restituiva lo score migliore tra le varie opzioni.
Applicando l'algoritmo e analizzando i risultati si nota subito che in questo caso la scelta di k=3 non risulta ottimale, si ottiene infatti una clusterizzazione ottima per i *clienti top* che vengono ottimamente individuati dall'algoritmo mentre le altre due tipologie di clienti non vengono correttamente suddivise ma anzi vengono tutti raggruppati in un unico secondo cluster ad eccezione di pochi clienti. Queste osservazioni possono essere fatte sia osservando come prima il risultato dello scatterplot 3D ma anche analizzando il dendogramma che si ottiene dall'esecuzione dell'algoritmo.

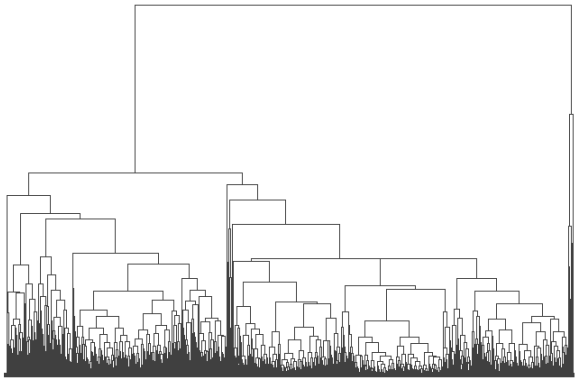

Figura 2: Dendogramma

Osservandolo la figura 2 si nota che i clienti risultano tutti molto vicini eccezion fatta per quelli all'estrema destra, questo tipo di clustering non risulta dunque molto adatto a partizionare i clienti in sole 3 categorie.

6 Validazione

La clusterizzazione, essendo una tecnica di machine learning non-supervisionata, a differenza della classificazione non permette una validazione semplice e ben definita dei risultati ottenuti."La convalida delle strutture di clustering è, infatti, la parte più difficile e frustrante dell'analisi dei cluster." [5]
Tuttavia, è chiaro che questo step riveste un'importanza notevole. Oltre alla valutazione soggettiva che può essere effettuata dall'esperto di dominio, vi sono altre tecniche applicabili per validare i risultati ottenuti. L'importanza di questo step è dovuta al fatto che gli algoritmi di clustering creano in qualsiasi caso una clusterizzazione a prescindere che i dati siano strutturati in modo da consentirlo o meno.

Di seguito si riportano alcune delle molteplici motivazioni che sono state considerate al fine di valutare il clustering:

- Valutare come il risultato del clustering modella i dati
- Determinare che si sia determinato il "corretto" numero di cluster
- Verificare la presenza di strutture non-randomiche nei dati
- Comparare le caratteristiche di due algoritmi di clustering per valutare quale è il migliore

Nella fase di validazione non è stato considerata la casistica della valutazione dei cluster rispetto ad informazioni esterne in quanto non disponibili nel dataset. Si è deciso di non creare nuove variabili, per evitare di produrre risultati falsati, che potessero identificare la classe di appartenenza del cliente in quanto necessita una considerevole conoscenza del dominio non a nostra disposizione.
Per svolgere questa parte di lavoro è stato necessario utilizzare diversi nodi RSnippet in quanto la sola piattaforma Knime non è sufficiente e non offre i nodi necessari per la validazione dei cluster.[2]

6.1 Indici interni

Gli indici interni misurano quanto una soluzione di clustering si adatti bene ai dati, quando i dati sono la sola informazione disponibile. Questi indici si basano sui concetti di coesione (Cluster Cohesion) e di separazione (Cluster Separation). Questi indicatori, infatti, misurano quantitativamente quanto la partizione ottenuta risponde all'obiettivo del clustering, individuando gruppi coesi e ben separati tra loro. Per questa validazione sono stati considerati due indici:

1. coefficiente di **Silhouette**: combina l'idea della coesione e della separazione e può essere calcolato per ognuna delle osservazioni del dataset. Per ogni punto si preferiscono valori alti, e dunque prossimi a 1. L'indice per un punto è così definito:

$$S_i = \frac{b_i - a_i}{max(b_i, a_i)} \in [-1, 1]$$

dove a_i è la distanza media tra e tutte le altre osservazioni nello stesso cluster e b_i è la distanza media minima dai punti in un altro cluster.
E' inoltre è possibile calcolare l'indice medio di silhouette per un cluster semplicemente calcolato la media del coefficiente di silhouette per i punti appartenenti al cluster considerato.
I risultati medi ottenuti da questo indice sono i seguenti:

	Silhouette
Hierarchical	0.386
K-means	0.342
Fuzzy	0.285

Tabella 3: Indice di Silhouette

Confrontando i risultati ottenuti il clustering gerarchico risulta essere il migliore anche se di poco rispetto a k-means.
Analizzando i grafici dei valori dei valori di silhouette ottenuti dai singoli punti del dataset notiamo però che vi sono delle differenze rilevanti. Nel caso del clustering gerarchico, riportato in figura 3, notiamo che un numero rilevante di punti ha un valore di silhouette negativo nonostante una buona silhouette media, inoltre si può notare, come già riportato nel paragrafo relativo a questo algoritmo, che il cluster identificato col colore blu non è praticamente individuabile in quanto rappresenta soltanto due record tra gli oltre 4000 a disposizione.

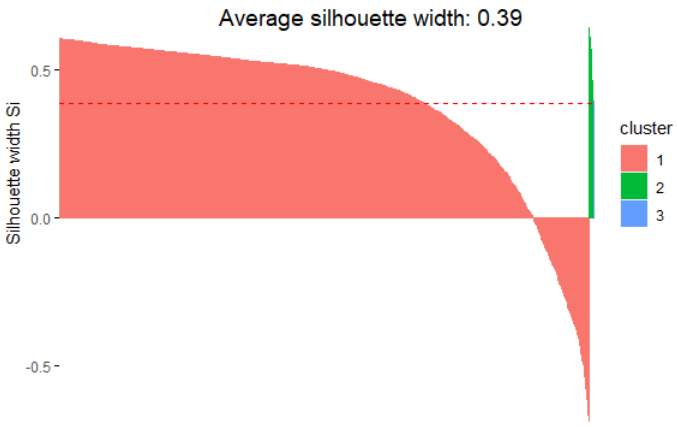

Figura 3: Indice di silhouette per il clustering gerarchico

Con riferimento alla figura 4 notiamo invece che i valori di silhouette ottenuti per i punti clusterizzati dall'algoritmo K-means sono quasi totalmente positivi e i pochi punti con valore negativo sono comunque molto più prossimi

[2]Per gli script si sono utilizzati i pacchetti cluster, ggplot2, clValid, kohonen, clue, factoextra, NBClust

allo zero rispetto agli score negativi analizzati in precedenza per il clustering gerarchico.

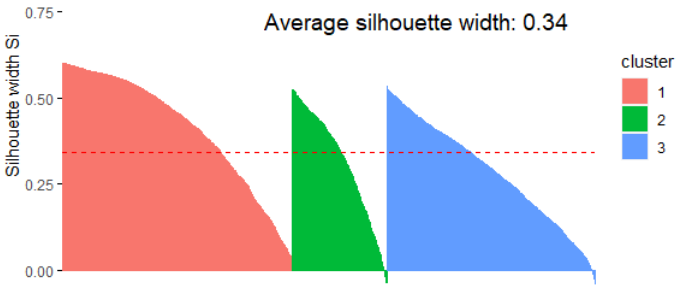

Figura 4: Indice di silhouette per l'algoritmo K-means

2. L'indice di **connectivity**: assume valore nel range $[0, +\infty]$ ed indica grado di connessione basandosi su k-nearest neighbors.
   Nel caso di questo indice, più basso è il valore più l'algoritmo di clustering è valido. I risultati ottenuti ponendo la dimensione di neighborhood pari a 2 sono in particolare i seguenti:

	Connectivity
Hierarchical	10
K-means	98.5
Fuzzy	117

Tabella 4: Indice di connectivity

Anche in questo caso dunque l'algoritmo gerarchico risulta essere il migliore tra i tre.

6.2 Numero di cluster

La scelta del numero di cluster pari a 3 come già menzionato precedentemente non è stata casuale ma è dovuta allo specifico obiettivo che ci eravamo posti in partenza. Ci siamo comunque chieste se una scelta diversa per il numero potesse incidere in modo positivo sull'analisi e sull'efficienza degli algoritmo.
Per questo motivo abbiamo calcolato tre indici, Dunn, Silhouette e Connectivity con un range variabile di cluster da 2 a 6. Questi indici consigliano di clusterizzare i dati utilizzando soltanto 2 cluster invece che 3, tuttavia i valori degli indici interni che si sarebbero ottenuti non sarebbero cosi diversi da quelli ottenuti con k=3 da noi utilizzato.
Per ottenere un quadro ancora più ampio abbiamo calcolato il numero di cluster ottimale da utilizzare secondo 30 indici diversi, da cui risulta che per k-means il numero ottimale di cluster sarebbero 2 (risultato ottenuto da 13 indici), anche se la seconda scelta sarebbe proprio k=3 proposta da 6 dei 30 indici.
Mentre lo stesso algoritmo per il clustering gerarchico propone come scelta ottimale 2 o 4 cluster, questo rispecchia anche i dati da noi ottenuti nel clustering gerarchico con k=3 ove un cluster era composto soltanto da due osservazioni e quindi non rilevante ai fini dell'analisi.

6.3 Paradigma di validità

Per verificare che esista un'effettiva struttura nei nostri dati è stato eseguito un test statistico con l'ipotesi nulla:

$$H_0 = \text{non esiste alcuna strutta per il dataset}$$

Per eseguire il test è stata assunta l'ipotesi che in letteratura troviamo sotto il nome di *Random Position Hypothesis* che afferma che tutte le posizioni degli m punti del dataset in una specifica regione di un piano n-dimensionale siamo ugualmente probabili.
In particolare, applicando il metodo dell'analisi di Monte Carlo viene generata una distribuzione empirica sulla base di n osservazioni, in questo caso $n = 1000$. Per calcolare il quantile di tale distribuzione è stato scelto il livello di significatività $\alpha = 0.01$. Inoltre per testare l'ipotesi $H0$ è stato utilizzato il coefficiente di Silhouette relativamente all'algoritmo k-means: si ha optato per questo algoritmo in quanto anche se il valore di silhouette è leggermente inferiore rispetto all'algoritmo gerarchico i cluster ottenuti risultano essere secondo la nostra valutazione soggettiva più validi e quindi abbiamo preferito testare $H0$ per questo algoritmo. Dall'esecuzione del test si ottiene la seguente distribuzione:

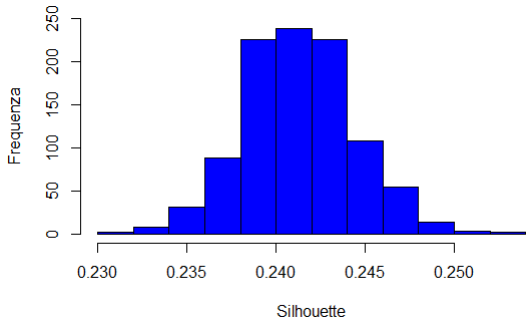

Figura 5: Distribuzione empirica del metodo di Monte Carlo

Dato che il quantile ottenuto è pari a 0.249 e differisce per una quantità maggiore $\alpha = 0.01$, fissato

precedentemente, rispetto al valore vero del coefficiente di silhouette pari a 0.342 allora concludiamo che l'ipotesi nulla H0 fissata precedentemente può essere rifiutata.

7 Criticità

Durante lo svolgimento del lavoro sono emerse alcune criticità dovute alla struttura del dataset. In primo luogo non avendo informazioni su come sono stati raccolti ed inseriti i dati si è dovuto procedere, per la fase di pulizia del dataset, prendendo decisioni soggettive che potrebbero inficiare sull'analisi di clustering poi condotta. Ad esempio la gestione degli ordini con quantità e/o prezzo unitario negativo, su cui si è scelto di rimuovere completamente il record potrebbe non essere corretta per tutte le casistiche presenti.
Inoltre i dati a disposizione si riferiscono sia a vendite al dettaglio che all'ingrosso senza però riportare, ne a livello di fattura ne di cliente, una suddivisione tra le due tipologie che avrebbe permesso di suddividere le due categorie ed effettuare così un'analisi più approfondita ed accurata. La presenza di ulteriori dati dati relativi al cliente, come ad esempio l'età, avrebbe inoltre permesso condurre analisi più dettagliate o anche specifiche per un certo target di clientela.

8 Conclusioni

Alla luce di quanto emerso nel corso dell'analisi descritta nei capitoli precedenti, si può affermare che i risultati ottenuti al fine di classificare i clienti per attuare una possibile campagna di marketing siano piuttosto ragionevoli a livello statistico. Tuttavia, è bene non dimenticare che alcuni aspetti relativi alla definizione e costruzione dei cluster dipendono fortemente dal contesto e dalle informazioni di cui si dispone, quindi non sempre si può essere certi che il quadro della situazione che si ottiene rispecchi fedelmente la realtà. Inoltre, alcune delle scelte prese nel corso del lavoro sono state dettate da considerazioni di natura soggettiva, come la scelta del numero di cluster.
Nonostante questo la suddivisione prodotta dagli algoritmi descritti in precedenza ha prodotto dei risultati che permettono di adottare misure di marketing più mirate. Il clustering gerarchico in particolare ha permesso di evidenziare in maniera abbastanza precisa i clienti top permettendo così di attuare politiche di marketing e promozioni adeguate per questa tipologia di clientela, sebbene non produce come desiderato una suddivisione altrettanto soddisfacente per le altre due tipologie di clienti.
Dall'altra parte invece la suddivisione prodotta da fuzzy e k-means risulta più conforme alle idee prefissateci in partenza suddividendo i clienti in tre grosse categorie, andando così non solo ad individuare i clienti top ma permettendo di selezionare i clienti su cui concentrarsi per il marketing senza inviare promozioni o sconti ai clienti che non acquistano da moltissimo tempo, i clienti "persi", evitando così di perdere ulteriori clienti.

9 Bibliografia

[1] Guy Brock et al. *clValid, an R package for cluster validation*. 2020. URL: http://cran.us.r-project.org/web/packages/clValid/vignettes/clValid.pdf.

[2] Malika Charrad et al. *Determining the Best Number of Clusters in a Data Set*. 2015. URL: https://cran.r-project.org/web/packages/NbClust/NbClust.pdf.

[3] A. Joy Christy et al. "RFM ranking – An effective approach to customer segmentation". In: *Journal of King Saud University - Computer and Information Sciences* (2018). DOI: https://doi.org/10.1016/j.jksuci.2018.09.004. URL: http://www.sciencedirect.com/science/article/pii/S1319157818304178.

[4] *E-commerce data, Actual transactions from UK retailer*. URL: https://www.kaggle.com/carrie1/ecommerce-data.

[5] Anil K. Jain e Richard C. Dubes. *Algorithms for Clustering Data*. USA: Prentice-Hall, Inc., 1988. ISBN: 013022278X.

[6] *KNIME*. URL: https://www.knime.com/.

[7] *Online Retail Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/online+retail.

[8] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: https://www.R-project.org/.

[9] Python Core Team. *Python: A dynamic, open source programming language*. Python Software Foundation. 2015. URL: https://www.python.org/..

[10] *Trasformazioni dei dati con test per normalità e outlier*. URL: http://www.dsa.unipr.it/soliani/capu13.pdf.

[11] Wikipedia. *RFM (market research)*. 2020. URL: https://en.wikipedia.org/wiki/RFM_(market_research).