Progetto C

Vincenzo Barreca
Giovanni Casano
Vincenzo Galia
Manuele Mormorio
Francesco Settecase
Gabriele Zimmerhofer

Indice

1 Introduzione

Il presente progetto si propone di condurre uno studio di profilazione dei clienti basato sull'analisi dell'indice RFM (Recency, Frequency, Monetary) utilizzando un Dataset contenente informazioni sulle transazioni di una società di vendita online con sede nel Regno Unito nel periodo compreso tra il 01/12/2010 e il 09/12/2011. L'obiettivo principale è identificare le eventuali abitudini dei clienti nei confronti della società al fine di realizzare una pubblicità mirata e personalizzata.

Il fulcro di questa analisi sta nell'individuare le features dell'indice RFM. Questo indice, composto da tre componenti chiave, fornisce una misura della relazione tra i clienti e l'azienda. Le tre features che compongono l'indice RFM sono:

- Recency (R): Rappresenta il numero di giorni trascorsi dall'ultima transazione effettuata dal cliente. Un indice basso indica una transazione più recente, mentre un valore alto una transazione lontana o inesistente.
- Frequency (F): Indica il numero totale di transazioni effettuate da un cliente durante un certo periodo di tempo. Un valore alto indica una maggiore frequenza di acquisti, mentre un valore basso indica una minore frequenza.
- Monetary (M): Rappresenta l'importo totale delle transazioni effettuate da un cliente durante un certo periodo di tempo. Un valore alto indica un maggiore valore monetario generato dal cliente, mentre un valore basso indica un contributo monetario più basso.

2 Analisi del Dataset

Per condurre il nostro studio, abbiamo a disposizione un Dataset che ci fornisce tutte le informazioni necessarie.

Prima di procedere con l'analisi, è bene andare a visualizzare quelle che sono le informazioni necessarie contenute nel Dataset, in modo da avere una panoramica generale adeguata. Per fare ciò, è possibile fare uso di alcuni metodi fondamentali della libreria Pandas, nello specifico il metodo *shape*, il quale restituisce la dimensione in termini di numero di righe e numero di colonne del Dataset, e il metodo *describe*, il quale restituisce diverse informazioni statistiche per ogni colonna.

L'uso di questi metodi fornisce i seguenti risultati:

Dataset dimension: (541909, 8)

	Quantity	*UnitPrice*	*CustomerID*
count	541,909	541,909	406,829
mean	9.552250	4.611114	15,287.690570
std	218.081158	96.759853	1,713.600303
min	-80,995.000000	-11,062.060000	12,346.000000
25%	1.000000	1.250000	13,953.000000
50%	3.000000	2.080000	15,152.000000
75%	10.000000	4.130000	16,791.000000
max	80,995.000000	38,970.000000	18,287.000000

Tabella 1: Risultati del metodo *describe* per le colonne del dataset

I risultati mostrati nella tabella, ottenuti tramite il metodo *describe*, forniscono informazioni interessanti sull'andamento delle transizioni. Si nota, per esempio, che il valore della funzione *count* per la colonna *CustomerID* è inferiore rispetto alle altre due colonne considerate, suggerendo la presenza di un numero significativo di elementi con valori nulli.

2.1 Features

Prima di eseguire un qualunque tipo di operazione all'interno del Dataset, può essere utile focalizzare l'attenzione sulle features generali che lo compongono.

Tramite il metodo *columns*, appartenente sempre alla libreria Pandas, è possibile visualizzare quelle che sono le features che compongono il Dataset. Il risultato che otteniamo è il seguente:

Columns of dataset: Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate', 'UnitPrice', 'CustomerID', 'Country']

Il nome di ciascuna colonna è scelto in modo da rappresentare il tipo di informazione contenuta in ognuno di essa.

2.2 Data Cleaning

Tramite l'osservazione dei risultati ottenuti durante l'uso del metodo *describe*, emerge un aspetto importante. All'interno del Dataset sono presenti numerosi osservazioni con valori mancanti in corrispondenza della colonna *CustomerID*. Questo risultato, ci suggerisce di applicare delle adeguate strategie di Data Cleaning per la gestione dei valori mancanti, in modo tale da poter avere delle analisi più precise.

Per affrontare questa problematica, abbiamo adottato la strategia di eliminare tutte le osservazioni contenenti valori mancanti. Abbiamo utilizzato il metodo *dropna*, che ci ha restituito un dataset privo di valori mancanti per tutte le colonne presenti nella Tabella 1. Infatti, rieseguendo i metodi *shape* e *describe*, otteniamo i seguenti risultati:

Dataset dimension: (406829, 8)

	Quantity	*UnitPrice*	*CustomerID*
count	406,829	406,829	406,829
mean	12.061303	3.460471	15,287.690570
std	248.693370	69.315162	1,713.600303
min	-80,995.000000	0.000000	12,346.000000
25%	2.000000	1.250000	13,953.000000
50%	5.000000	1.950000	15,152.000000
75%	12.000000	3.750000	16,791.000000
max	80,995.000000	38,970.000000	18,287.000000

Tabella 2: Risultati del metodo *describe* dopo l'eliminazione dei valori mancanti

Dopo questa fase di pulizia, il Dataset risulta essere in una forma più adeguata. Possiamo dunque procedere con l'estrazione delle features di nostro interesse e iniziare l'analisi dei dati.

3 Analisi dei dati

3.1 Estrazione delle features

Di vitale importanza per lo svolgimento della profilazione è l'estrazione delle features Recency, Frequency e Monetary. Queste, dopo averle normalizzate mediante metodo *StandardScalar*, saranno pronte per essere analizzate tramite la tecnica del Clustering.

Di seguito è riportato il codice che effettua l'estrazione delle features RFM dal Dataset.

```
rfm_data = data.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (data['InvoiceDate'].max()
        - x.max()).days,
```

```
    'InvoiceNo': 'count',
    'Quantity': 'sum'
})

rfm_data.columns = ['Recency', 'Frequency', 'Monetary']
```

Nel codice appena riportato, viene utilizzato il metodo *groupby* per raggruppare i dati in base al valore della colonna *CustomerID*. Questo ci permette di creare gruppi separati per ogni cliente, consentendoci di poter studiare il loro comportamento individualmente.

Successivamente, utilizziamo il metodo *agg* per eseguire delle operazioni di aggregazione sui dati ottenuti nei singoli gruppi. Per fare ciò, abbiamo definito un dizionario che conterrà le operazioni aggregate cercate.

Per calcolare la feature Recency, applichiamo il metodo *lambda* per ottenere il numero di giorni trascorsi dall'ultima transazione per ogni cliente. Per la feature Frequency, utilizziamo il metodo *count* per contare il numero totale di transazioni eseguite per ciascun cliente. Per Monetary, invece, utilizziamo il metodo *sum* per calcolare il costo complessivo dei prodotti acquistati.

Infine, andiamo a rinominare le colonne del Dataset ottenuto per rendere i nomi delle colonne più significative.

Dopo aver fatto ciò, applichiamo il metodo *StandardScalar* per normalizzare le features ottenute. Questo metodo ci consente di avere un set di variabili standardizzate, con media zero e deviazione standard pari a uno, che potranno essere più facilmente confrontabili.

```
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_data)
```

3.2 Scelta del numero di cluster

Prima di poter applicare un vero e proprio algoritmo di Clustering, nello specifico il *K-means*, è necessario individuare qual è il numero ottimale di cluster da utilizzare.

Questa scelta è fondamentale, in quanto andrà a determinare la precisione e l'accuratezza del risultato finale. Le tecniche che abbiamo utilizzato sono il metodo *Elbow* e l'indice di *Silhoutte*, dai quali abbiamo ottenuto i seguenti risultati:

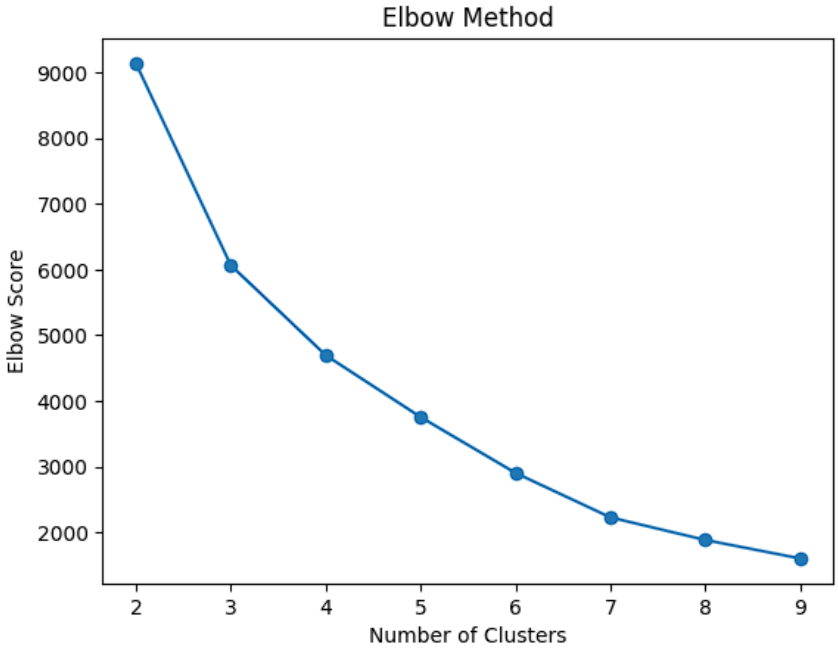

Figura 1: Grafico *elbow* per la scelta del numero di cluster

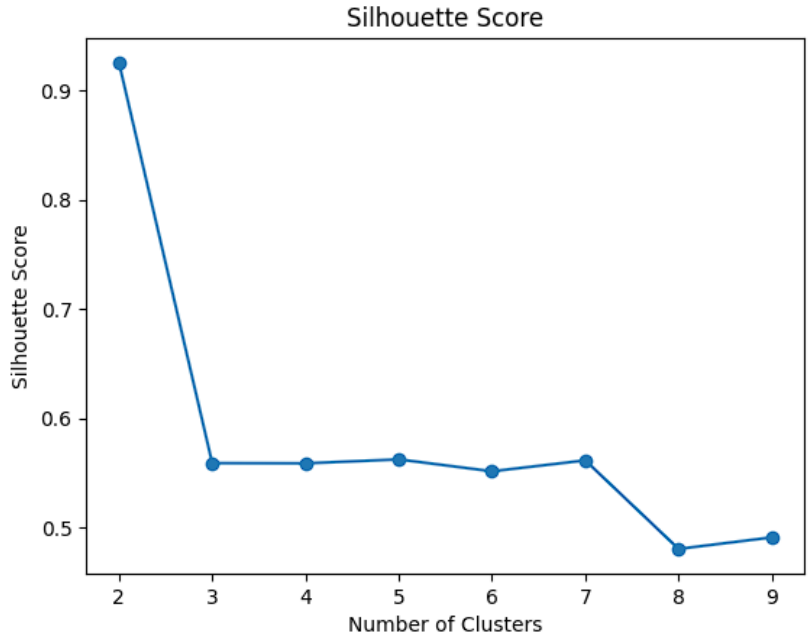

Figura 2: Grafico *Silhoutte* per la scelta del numero di cluster

Osservando il grafico del metodo *Elbow* in Figura 1, notiamo la presenza di punti di discontinuità, o di flessione, che possono essere utilizzati come possibili candidati per il numero di cluster da utilizzare. Tali punti si trovano in corrispondenza dei valori 2 e 3.

All'interno del grafico del metodo *Silhoutte* in Figura 2, dobbiamo individuare i punti che massimizzano l'indice. Questo valore rappresenta la coesione ottimale tra i cluster. Dunque, il punto più alto del grafico corrisponde anche al numero ottimale di cluster da utilizzare.

Analizzando bene i grafici, emerge dunque che il numero ottimale di cluster da utilizzare sia 2. Tuttavia, l'utilizzo di questo numero di cluster non fornisce informazioni sufficienti per poter proseguire l'analisi. Pertanto, il numero di cluster scelto è 3.

3.3 Applicazione del Clustering

Dopo aver determinato il numero di cluster da usare, possiamo procedere con l'esecuzione dell'algoritmo di Clustering. In generale, il Clustering è una tecnica di apprendimento non supervisionato che ci consente di identificare pattern e strutture all'interno dei dati che si stanno analizzando. Tale approccio è particolarmente utile quando ci ritroviamo a dover analizzare dei dati che non hanno delle etichette di classe associate.

L'algoritmo usato in questo caso è il *K-means*, nonché uno dei più diffusi. Lo scopo del *K-means* è quello di partizionare i dati in k gruppi sulla base dei loro attributi. Grazie all'uso di questo algoritmo, siamo in grado di identificare dei gruppi omogenei di clienti.

Per eseguire l'algoritmo di Clustering, identificando i cluster e i relativi centri di massa, viene eseguito il seguente codice:

```
kmeans = KMeans(n_clusters=3, random_state=42, n_init='auto')
kmeans.fit(rfm_scaled)

centroids = scaler.inverse_transform(kmeans.cluster_centers_)
```

Nel codice appena riportato, eseguiamo l'algoritmo *K-means* andando a partizionare i dati in tre cluster, raggruppando gli elementi in modo appropriato. Inoltre, identifichiamo anche i centroidi, ossia i punti centrali di ogni cluster.

A questo punto, possiamo andare a vedere graficamente i risultati ottenuti:

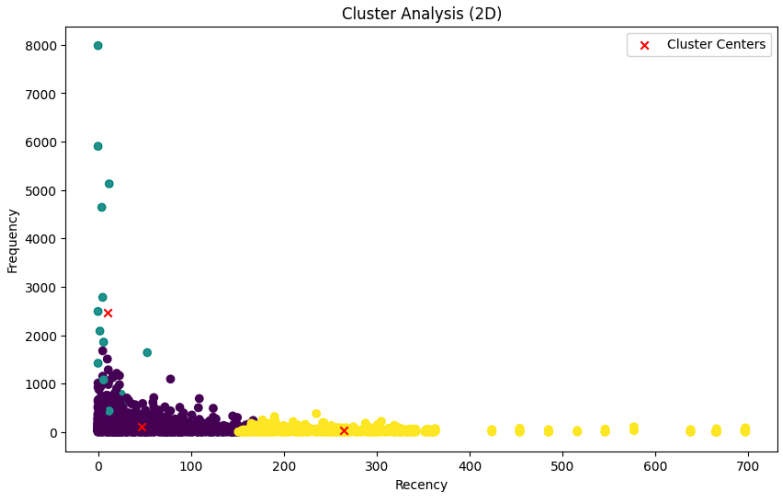

Figura 3: Grafico 2D dei risultati del Clustering

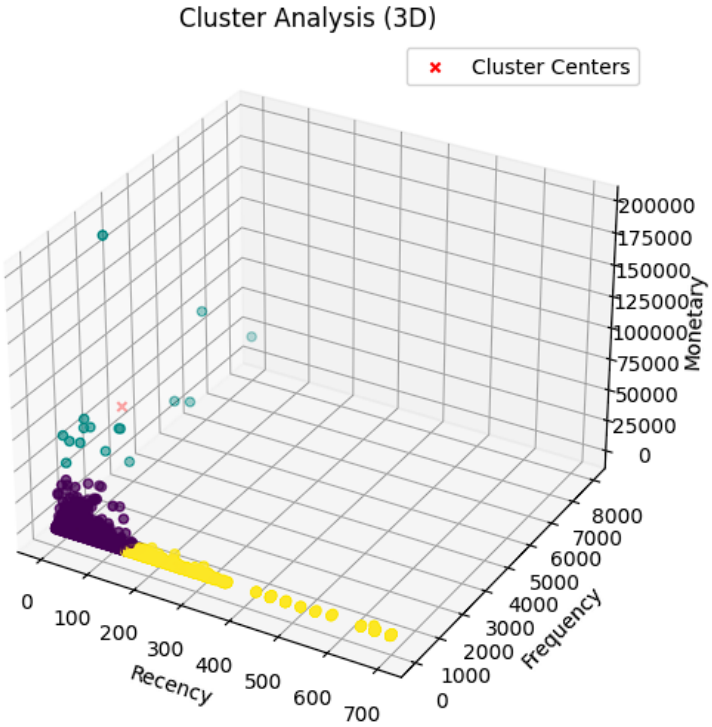

Figura 4: Grafico 3D dei risultati del Clustering

Osservando i grafici ottenuti, si nota la formazione di tre principali gruppi di clienti. Tuttavia, i cluster rilevanti sono solamente due.

Il primo gruppo, rappresentato dal colore viola, va a racchiudere al suo interno quella porzione di clienti che hanno una frequenza d'acquisto maggiore, che hanno anche effettuato transazioni con una buona somma di denaro spesa in un periodo relativamente recente. Questo cluster ci permette di identificare quella parte di clienti che è più fidelizzata nei confronti della società.

Il secondo gruppo, rappresentato dal colore giallo, comprende quella fetta di clienti che effettua acquisti con bassa frequenza e che non hanno interagito con la società da molto tempo. Questo cluster ci permette di identificare tutti quei clienti che sono considerati inattivi e che eseguono acquisti sporadici.

Queste osservazioni ottenute tramite il Clustering, ci permettono di poter andare ad applicare delle strategie di marketing mirate in modo da eseguire delle campagne pubblicitarie.

All'interno dei grafici, si nota una presenza piuttosto rilevante di dati *outliers*. Questi potrebbero andare a distorcere i centri di massa dei vari cluster e di conseguenza andare ad inficiare sull'analisi complessiva. Per questo motivo, è necessario andare a valutare se rimuovere questi valori o semplicemente ignorarli, qualora il loro contributo risulta essere minimo.

L'eliminazione degli *outliers* porta ad avere una sovrapposizione di cluster che rende poco chiara la distinzione dei gruppi di clienti. Se si volesse scendere ancora più nel dettaglio, cercando di ottenere dei risultati ancora più ottimali, sarebbe necessario l'applicazione di ulteriori tecniche di analisi.

4 Conclusioni

I risultati ottenuti tramite Clustering, dopo aver pulito i dati ed estratto le features di nostro interesse, hanno evidenziato la presenza di due principali gruppi di clienti con caratteristiche differenti.

Studiando le abitudini di acquisto e il rapporto esistente tra questi gruppi di clienti e la società, siamo in grado di realizzare delle strategie di marketing per soddisfare due obiettivi. Il primo obiettivo è quello di rafforzare la fidelizzazione dei clienti appartenenti al primo gruppo nei confronti della società. Il secondo obiettivo è stimolare i clienti del secondo gruppo, cercando di farli interagire in modo più interessato e attivo con la società.

Per il primo gruppo di clienti, una strategia efficace potrebbe consistere nella creazione di *programmi fedeltà*. Questi potrebbero offrire dei vantaggi esclusivi o delle promozioni uniche provando a mantenere la frequenza di acquisto e l'in-

terazione con la società costante nel tempo, premiando la fedeltà dimostrata. Inoltre, si potrebbe far uso di newsletter personalizzate in base alle esigenze dei singoli clienti, basandosi per esempio sugli articoli verso cui hanno mostrato interesse nel passato. Ulteriore incentivo potrebbe essere dato da sconti speciali, cashback o programmi di affiliazione consigliando l'uso della piattaforma ad altre persone.

Per il secondo gruppo, invece, se desideriamo riportare i clienti ad utilizzare nuovamente la piattaforma, è necessario applicare delle strategie di *re-engagement*. Un metodo efficace potrebbe essere quello di inviare delle comunicazioni personalizzate, mirate a ricordare agli utenti le loro interazioni precedente, evidenziato sconti e promozioni presenti per prodotti verso cui hanno mostrato interesse o per articoli correlati. Inoltre, potrebbe essere opportuno realizzare delle offerte ad hoc, diverse rispetto a quelle già presenti, al fine di stimolare un ulteriore acquisto e sfruttare gli eventuali dati raccolti per future strategie ancora più personalizzate.

In conclusione, tramite le tecniche utilizzate, siamo riusciti ad individuare due principali gruppi di clienti e sviluppare delle strategie mirate per entrambi. Queste strategie hanno lo scopo di massimizzare la fidelizzazione dei clienti, aumentare il loro numero e generare un incremento significativo dei guadagni per la società.