Ingegnerizzazione delle caratteristiche

1. Introduzione

Questo documento ha lo scopo di spiegare i passaggi e i ragionamenti posti all'attività dell'ingegnerizzazione delle caratteristiche.

Questo task si pone l'obbiettivo di creare delle nuove colonne del dataframe ottenendole da dati preesitenti. Questo compito è essenziale in quanto spesso dataset non hanno tutte le informazioni necessarie per una corretta analisi o perchè modelli come quello scelto in questo caso, RFM spiegato successivamente, hanno bisogno di dati particolari.

Nei capitoli successivi verranno spiegati i passaggi per la creazione di queste nuove features e prima ancora ci sarà una spiegazione del modello RFM.

2. Modello RFM

Il modello RFM (Recency, Frequency e Monetary) è una tecnica utilizzata nell'analisi dati per segmentare la clientela e valutare il comportamento d'acquisto. Questo modello si basa sulle tre caratteristiche:

- Recency: Indica quanto recentemente un cliente ha effettuato un pagamento, nel mio caso è stato assegnato un valore giornaliero, ma può essere anche settimanale o mensile a seconda dei casi e dalla dimensione e temporalità del dataset;
- Frequency: Misura il numero di acquisti effettuatu da un cliente nell'arco di un tempo circoscritto, in questo caso nella totale interezza in quanto di un solo anno;
- Monetary: Misura il totale speso da un cliente sempre nell'arco di un tempo circoscritto.

Queste tre features sono essenziali per comprendere i comportamenti dei clienti, in maniera da capire la disponibilità a rispondere positivamente a nuove proposte o ad esempio anche alla loro fedelizzazione. Questo permette di personalizzare strategie di marketing più mirate per ciascun segmento.

Il modello RFM non viene utilizzato soltato per la divisione in cluster della clientela ma anche per attività comela Retention e la previsione delle vendite.

3. Recency

La creazione del nuovo attributo Recency è stato quello leggermente più complicato ma comunque abbastanza basilare.

Si è dovuto prima convertire la colonna in formato datetime della libreria pandas, successivamente si è creata una variabile max_date che estraeva la data massima presente nel dataset tramite la funzione `max_date = max(df["Date"])` .

Una volta conclusa questa operazione veniva creata una nuova colonna dove veniva sottrata a max_date la data dell'acquisto del prodotto e tenuta la porzione di giorni `df["Recency"] = (max_date - df["Date"]).dt.days`.

Veniva poi creato un nuovo dataframe tramite la funzione groupby su "customerID" e ottenendo il valore minimo per cliente sulla colonna "Recency": `df_r = df.groupby("CustomerID")["Recency"].min().reset_index()`.

4. Frequency

La creazione dell'attributo Frequency risulta essere più semplice, viene subito creato un nuovo dataframe sempre tramite funzione groupby su "customerID" andando a contare le "InvoiceNo" e poi semplicemente rinominando "InvoiceNo" in "Frequency".

Riporto le due righe di codice per la creazione della colonna:

```
df_f = df.groupby("CustomerID")["InvoiceNo"].count().reset_index()
df_f.rename(columns={"InvoiceNo": "Frequency"}, inplace=True)
```

5. Monetary

La creazione di Monetary prevede prima la creazione di una colonna "Total" formata attraverso il prodotto del costo unitario per la quantità acquistata.

successivamente si procede similarmente come le altre features attraverso un groupby, sempre con il pivot su "customerID" andando però a fare la somma dei valori di "Total": `df_m = df.groupby("CustomerID")["Total"].sum().reset_index()`.

Una volta creati i tre distinti dataframe è stato fatto un merge su "customerID" ottenendo un unico dataframe strutturato con "customerID", "Recency", "Frequency", "Total (Monetary)".

	CustomerID	Total	Frequency	Recency
0	12346.0	77183.60	1	325
1	12347.0	4310.00	182	2
2	12348.0	1437.24	27	75
3	12349.0	1457.55	72	18
4	12350.0	294.40	16	310

Figure 1: Dataframe RFM