CAPITOLO 3

Clustering

3.1 ETL per Clustering

Il dataset, precedentemente descritto, per la fase relativa alla clusterizzazione, è stato sottoposto a una fase preliminare di ETL e preprocessing al fine di rendere più efficace e robusta la successiva analisi clustering.

Inizialmente, i dataset di training e test sono stati unificati in un unico dataset per garantire una base di informazioni completa, composta da da 129.880 osservazioni e 25 variabili, per le tecniche non supervisionate che sono state adottate

Successivamente sono state eliminate alcune colonne, in particolare gli attributi **id** e **Unnamed: 0** in quanto rappresentano esclusivamente identificatori univoci e non forniscono informazioni utili ai fini del clustering.

Il loro utilizzo, avrebbe solamente provocato l'introduzione di rumore all'interno dei risultati, aumentando inutilmente la **dimensionalità** dello spazio delle feature.

Successivamente è stata individuata la variabile **satisfaction**, che rappresenta l'informazione di interesse, la variabile target. Poiché le tecniche di clustering adottate sono non supervisionate, tale variabile è stata esplicitamente esclusa dall'insieme delle feature, evitando così fenomeni di **data leakage**.

La variabile è stata tuttavia mantenuta nel dataset originale per consentire l'interpretazione dei cluster ottenuti.

Il dataset restante, è stato suddiviso in

- Variabili numeriche : 18;
- Variabili categoriche: 4.

I valori mancanti sono stati gestiti tramite imputazione:

- per le variabili numeriche è stata utilizzata la mediana;

- per le variabili categoriche è stata utilizzata la moda, ovvero il valore più frequente.

Al termine di questa fase, il dataset risulta privo di valori mancanti. È stato inoltre verificato che non fossero presenti osservazioni duplicate.

In seguito, poiché molte delle tecniche successive richiedono dati numerici, le variabili categoriche sono state trasformate tramite **one-hot encoding**, utilizzando una codifica con eliminazione del primo livello per evitare ridondanze.

A seguito di questa trasformazione, il numero totale di feature è risultato pari a 23.

Inolte, tutte le feature sono state sottoposte a un processo di standardizzazione tramite **StandardScaler()**, trasformando ciascuna variabile affinché presentasse media pari a zero e deviazione standard unitaria.

Questa fase è fondamentale per garantire che tutte le variabili contribuiscano in maniera equilibrata alle analisi di riduzione della dimensionalità.

In assenza di standardizzazione, le feature caratterizzate da valori più elevati avrebbero avuto un peso sproporzionato nel calcolo dei risultati.

La standardizzazione applicata segue la trasformazione *z-score*, definita dalla seguente formula:

$$z = \frac{x - \mu}{\sigma}$$

Questa fase di preprocessing consente di ridurre l'impatto della curse of dimensionality e costituisce un prerequisito essenziale per l'applicazione delle tecniche di riduzione dimensionale (PCA, t-SNE, UMAP) e degli algoritmi di clustering (K-Means, DBSCAN e GMM) analizzati nelle sezioni successive.

Di seguito, viene riportato in breve, un semplcie output relativo alla fase di preprocessing:

```
CHECK DATI FINALI CLUSTERING
Duplicati presenti: 0
Valori Mancanti totali dopo imputazione: 0
Colonne numeriche originali: 18
Colonne categoriche originali: 4

Riepilogo:
Righe: 129880
Feature dopo encoding: 23
Colonne numeriche: Age, Flight Distance, Inflight wifi service,
Departure/Arrival time convenient, Ease of Online booking,
Gate location, Food and drink, ...
Colonne categoriche: Gender, Customer Type, Type of Travel, Class
```

3.2 Analisi della varianza delle feature

Al fine di comprendere l'eterogeneità delle variabili e **motivare la scelta della standardizzazione**, è stata condotta un'analisi della varianza delle feature nello spazio originale dei dati, prima dell'applicazione dello z-score.

Per ciascuna variabile numerica e per le variabili categoriche codificate tramite one-hot encoding sono state calcolate la varianza e la deviazione standard.

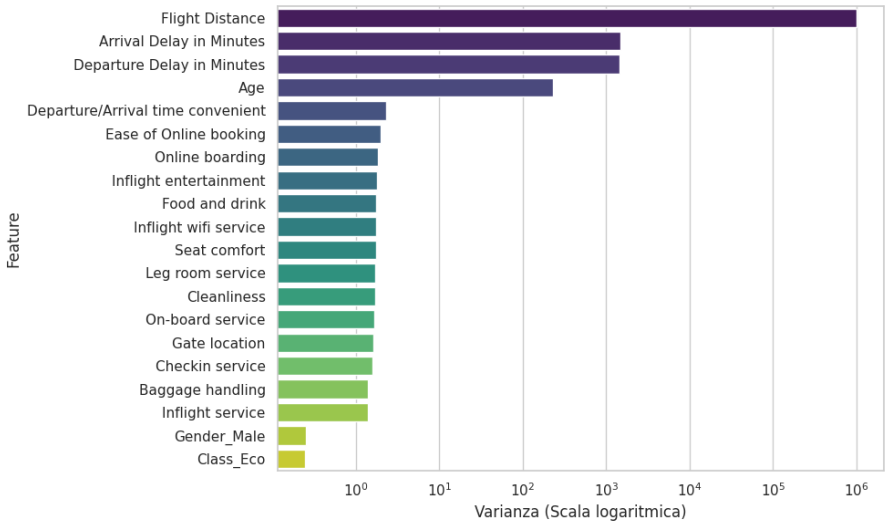

Figura 3.1: Top 20 feature per varianza nello spazio originale dei dati.

Dalla Figura 3.1 emerge un importante eterogeneità nelle scale delle variabili, in particolare, *Flight Distance* mostra una varianza dell'ordine di 10^6,nettamente superiore, mentre la maggior parte delle variabili di servizio presenta varianze prossime all'unità.

Questa analisi è stata svolta con l'obiettivo di giustificare l'applicazione della standardizzazione: in assenza di tale trasformazione, le variabili caratterizzate da scale più ampie tenderebbero a dominare le misure di distanza utilizzate dagli algoritmi di clustering.

3.3 PCA

L'analisi delle componenti principali (PCA) è stata applicata ai dati in seguito alla fase di standardizzazione con l'obiettivo di ridurre la dimensionalità dello spazio delle feature rendendo più agevole l'applicazione delle tecniche di clustering.

In Figura 3.2 è mostrato lo Scree Plot, che rappresenta la varianza spiegata da ciascuna componente principale.

Dal grafico si osserva come le prime componenti catturino una quota significativa della variabilità totale del dataset, mentre il contributo delle componenti successive decresce progressivamente. Questo andamento suggerisce che una parte rilevante dell'informazione è concentrata nelle prime direzioni principali dello spazio dei dati.

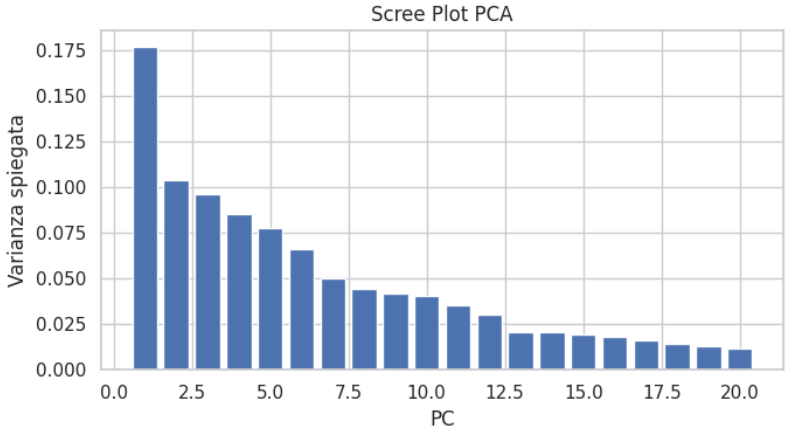

Figura 3.2: Scree Plot della PCA

Per valutare in modo più chiaro quanta informazione venga mantenuta al crescere del numero di componenti, in Figura 3.3 è riportata la varianza spiegata cumulata. Dal grafico emerge che le prime dieci componenti principali permettono di spiegare circa **l'80% della varianza complessiva**. Sulla base di questo risultato, si è scelto di utilizzare uno spazio **PCA a 10 dimensioni** nelle analisi successive, ottenendo un compromesso adeguato tra riduzione dimensionale e conservazione dell'informazione.

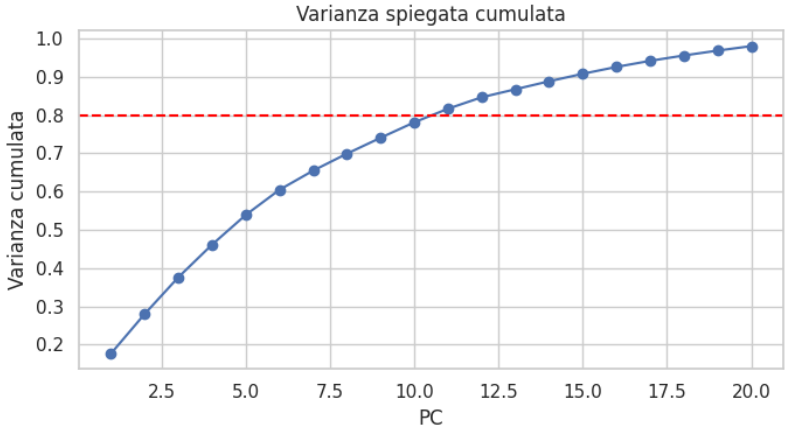

Figura 3.3: Varianza spiegata cumulata della PCA

Di seguito, in Figura 3.4, viene ripotata una proiezione dei dati sulle prime due componenti principali.

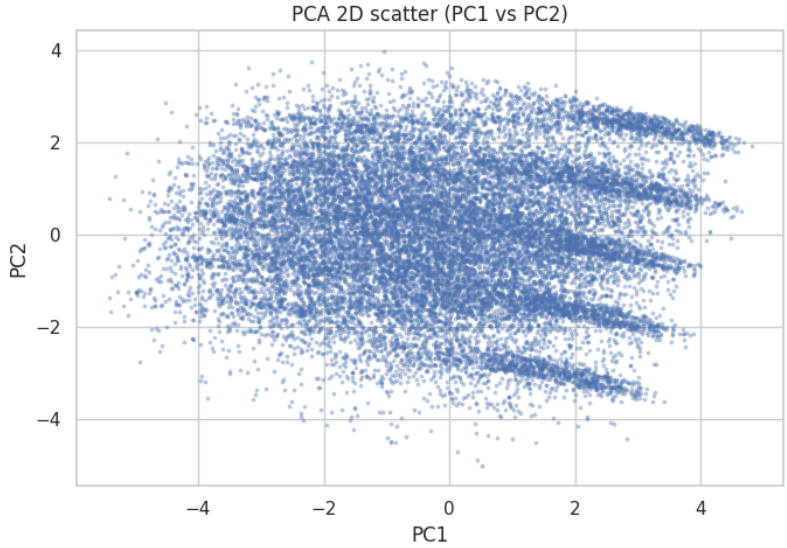

Figura 3.4: Proiezione dei dati sulle prime due componenti principali

Questa visualizzazione, sebbene la PCA non sia una tecnica di clustering, ci garantisce di osservare quella che è la **struttura globale** dei dati e la presenza di eventuali pattern/sovrapposizioni nello spazio ridotto.

La distribuzione dei punti evidenzia una **struttura complessa e non linearmente separabile**, giustificando l'utilizzo di tecniche di riduzione dimensionale non lineari e di clustering nelle fasi successive.

L'obiettivo successivo è stato quello di analizzare il contributo delle singole variabili alle componenti principali. Per tale motivo, è stata generata ed esaminata la matrice dei loadings.

Sempre in relazione alle prime dieci componenti principali, in Figura 3.5 è riportata la heatmap dei loadings.

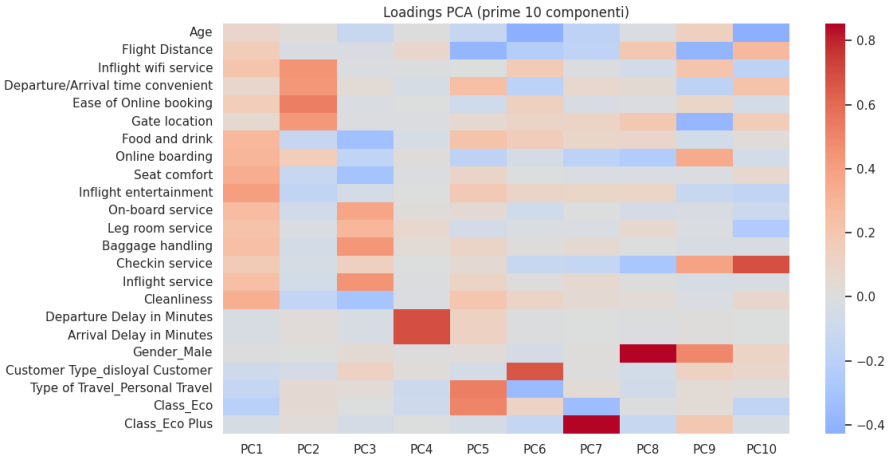

Figura 3.5: Heatmap dei loadings PCA (prime 10 componenti)

Dalla figura si osserva come diverse variabili legate alla qualità del servizio, ai ritardi e alle caratteristiche del viaggio contribuiscano in modo significativo alle principali direzioni di variazione dei dati.

Infine, al fine di individuare le feature maggiormente rilevanti nello spazio ridotto, è stata calcolata l'importanza complessiva di ciascuna variabile come somma dei valori assoluti dei loadings sulle prime dieci componenti principali. I risultati sono riportati in Figura 3.6.

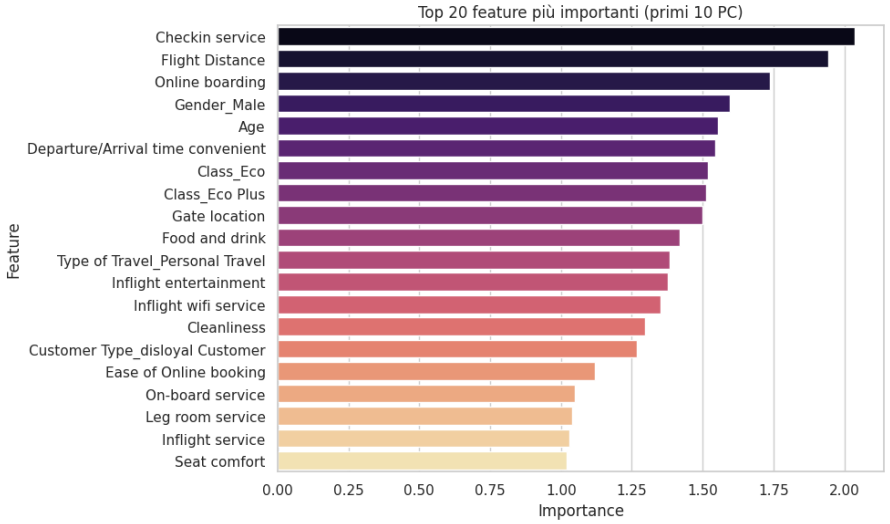

Figura 3.6: Top 20 feature più importanti nei primi 10 PC

3.4 Riduzione dimensionale non lineare tramite t-SNE

Successivamente alla riduzione dimensionale lineare mediante PCA, è stata applicata una **tecnica di riduzione dimensionale non lineare** al fine di ottenere una rappresentazione bidimensionale dei dati più adatta alla fase di clusterizzazione.

La tecnica **t-Distributed Stochastic Neighbor Embedding** (t-SNE) è un metodo di riduzione della dimensionalità progettato principalmente per la visualizzazione di dati complessi in spazi a bassa dimensione, tipicamente bidimensionali.

L'obiettivo principale di t-SNE è preservare le relazioni locali tra le osservazioni, proiettando punti simili nello spazio originale in posizioni vicine anche nello spazio ridotto.

t-SNE lavora trasformando le distanze tra le osservazioni in probabilità di vicinato. Due punti che sono molto simili nello spazio originale hanno una probabilità elevata di essere vicini, mentre punti poco simili hanno una probabilità molto bassa.

L'algoritmo costruisce quindi:

- una distribuzione di probabilità delle similarità nello spazio ad alta dimensionalità;
- una distribuzione analoga nello spazio bidimensionale.

Attraverso un processo iterativo di ottimizzazione, t-SNE cerca una proiezione tale da rendere le due distribuzioni il più simili possibile, preservando soprattutto le relazioni locali.

In questo modo, punti simili vengono collocati vicini nello spazio ridotto, mentre punti con bassa probabilità di vicinanza nello spazio ad alta dimensionalità vengono separati, producendo visualizzazioni molto efficaci di strutture e cluster complessi, anche in presenza di relazioni non lineari.

L'algoritmo t-SNE è stato applicato utilizzando i seguenti parametri di input:

- **Input dei dati**: le prime 10 componenti principali ottenute tramite PCA, al fine di ridurre il rumore e migliorare la stabilità della proiezione.
- **n_components = 2**: numero di dimensioni dello spazio di output, scelto per ottenere una rappresentazione bidimensionale dei dati.
- **perplexity = 30**: parametro che controlla il bilanciamento tra la preservazione delle strutture locali e globali, interpretabile come una stima del numero di vicini considerati per ciascun punto.
- **random_state = 42**: seme del generatore di numeri casuali, fissato per garantire la riproducibilità dei risultati.
- **numero di osservazioni = 20 000**: sottoinsieme del dataset originale, estratto in modo casuale ma riproducibile per ridurre il costo computazionale.

Per migliorare la stabilità numerica e ridurre il rumore, t-SNE non è stato applicato direttamente ai dati originali standardizzati, ma alle prime dieci componenti principali ottenute tramite PCA. Inoltre, a causa dell'elevata numerosità del dataset, l'algoritmo è stato eseguito su un sottoinsieme di 20,000 osservazioni, selezionato in modo casuale ma riproducibile.

La Figura 3.7 mostra la proiezione bidimensionale ottenuta. Dal grafico si osserva la presenza di strutture complesse e non linearmente separabili, caratterizzate da addensamenti locali e regioni ben distinte nello spazio.

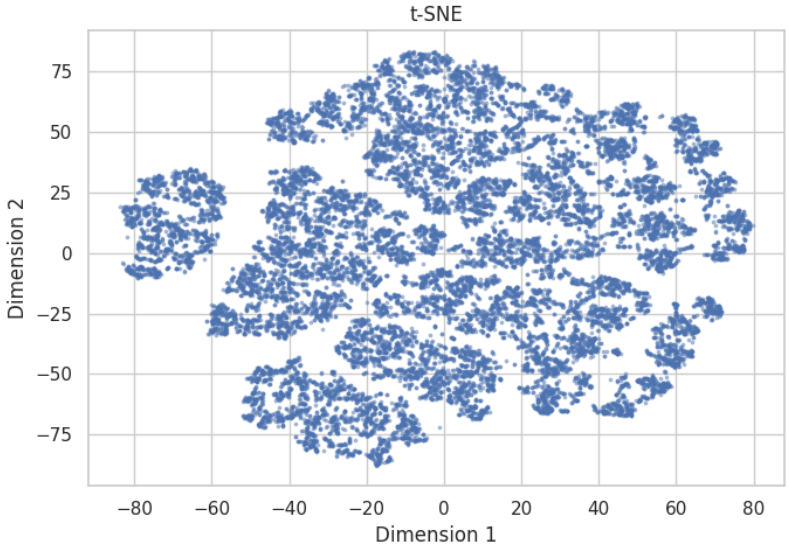

Figura 3.7: Rappresentazione bidimensionale dei dati tramite t-SNE.

3.5 K-Means

Il metodo **K-Means** è una tecnica di clustering per partizionamento ampiamente utilizzata nell'analisi esplorativa dei dati, grazie alla sua semplicità concettuale e alla capacità di individuare gruppi omogenei all'interno di un dataset.

L'algoritmo richiede la definizione di un unico parametro, **k**, che rappresenta il **numero di cluster da individuare**.

L'obiettivo principale di K-Means consiste nel **minimizzare** la **distanza intra-cluster**, ovvero la distanza tra ciascun punto e il centroide del cluster di appartenenza, e **massimizzare** la **distanza inter-cluster**, favorendo una separazione netta tra i gruppi individuati.

Il funzionamento dell'algoritmo si basa su un processo iterativo che alterna l'assegnazione delle osservazioni al centroide più vicino e il ricalcolo dei centroidi come media dei punti appartenenti a ciascun cluster. Il procedimento termina quando non

si verificano più variazioni significative nelle assegnazioni. Sebbene K-Means sia computazionalmente efficiente rispetto ad altre tecniche di clustering, la sua complessità cresce con l'aumentare della dimensionalità e della numerosità del dataset. Per questo motivo, nel presente lavoro l'algoritmo è stato applicato a uno spazio a dimensionalità ridotta ottenuto tramite t-SNE.

3.5.1 Applicazione K-Means

Per l'applicazione dell'algoritmo è stato necessario determinare un valore adeguato del parametro k.

Una scelta non appropriata di tale parametro può condurre a fenomeni di underfitting, nel caso di un numero di cluster troppo ridotto, oppure di overfitting, nel caso opposto. Al fine di individuare un valore ottimale di k, è stata condotta un'analisi preliminare considerando valori compresi tra 2 e 10.

In particolare, è stato utilizzato il **metodo del gomito**, che si basa sull'analisi della *Within-Cluster Sum of Squares* (WCSS) al variare del numero di cluster. Tale metodo consente di individuare un punto a partire dal quale l'aggiunta di nuovi cluster non comporta una riduzione significativa della varianza intra-cluster.

In Figura 3.8 è riportato l'andamento della WCSS al variare del numero di cluster. Dal grafico si osserva una marcata diminuzione della WCSS passando da $k = 2$ a $k = 3$, seguita da una riduzione progressivamente meno accentuata per valori superiori di k. Questo comportamento suggerisce la presenza di un "gomito" in corrispondenza di $k = 3$.

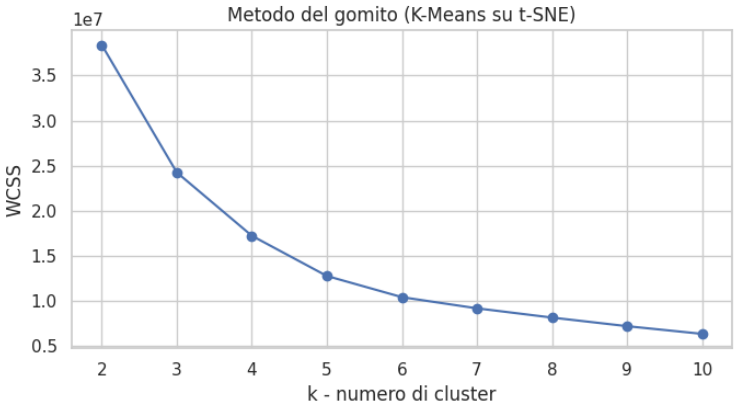

Figura 3.8: Metodo del gomito (Elbow).

Parallelamente, è stato calcolato l'indice di silhouette medio, al fine di valutare la qualità del clustering in termini di compattezza interna dei cluster e separazione tra gruppi distinti.

In Figura 3.9 è mostrato l'andamento dell'indice di silhouette al variare di k. Sebbene il valore massimo venga raggiunto per $k = 6$, si osserva che il valore ottenuto per $k = 3$ risulta comunque elevato e comparabile, a fronte di una minore complessità del modello.

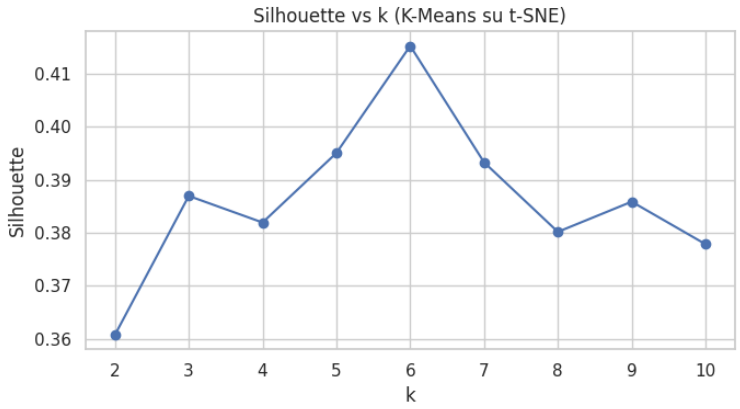

Figura 3.9: Indice di silhouette medio in funzione del numero di cluster.

Per ciascun valore di k, l'algoritmo K-Means è stato eseguito mantenendo fissi i principali parametri di configurazione, al fine di rendere confrontabili i risultati ottenuti. In particolare:

- il numero di inizializzazioni dei centroidi **n_init** è stato impostato pari a 20;
- il parametro **max_iter** è stato mantenuto al valore di default pari a 300.

Dall'analisi congiunta dei due criteri emerge che il valore $k = 3$ rappresenta un buon compromesso tra qualità del clustering e semplicità del modello.

Per questo motivo, nel prosieguo dell'analisi è stato adottato un valore di $k = 3$ per l'applicazione finale dell'algoritmo K-Means.

3.5.2 Analisi Risultati

L'applicazione finale dell'algoritmo K-Means nello spazio bidimensionale ottenuto tramite t-SNE ha prodotto **tre** cluster ben distinti, con un valore medio dell'indice di silhouette pari a 0.387, come mostrato in Figura 3.10.

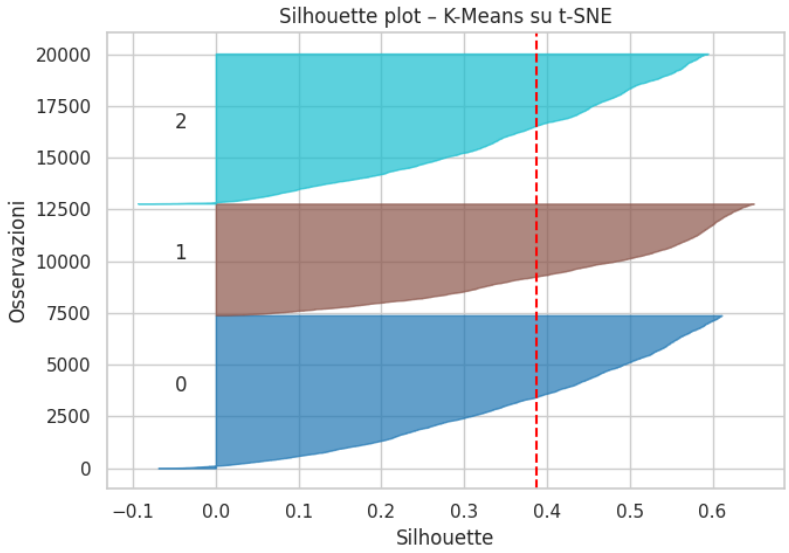

Figura 3.10: Grafico della Silhouette.

Tale valore indica una separazione complessivamente discreta tra i gruppi. I tre cluster ottenuti contengono, come riportato in Figura 3.11,

- Cluster 0: 7381 elementi;
- Cluster 1: 5380 elementi;
- Cluster 2: 7239 elementi;

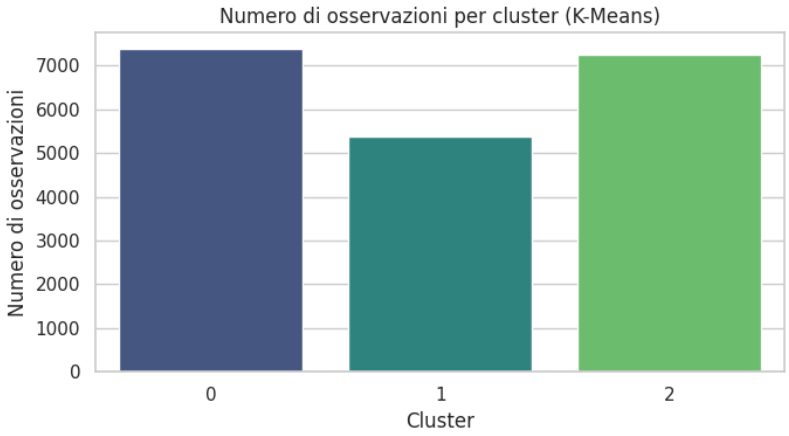

Figura 3.11: Composizione dei cluster K-Means.

L'applicazione finale dell'algoritmo K-Means con $k = 3$ è illustrata in Figura 3.12, che evidenzia la presenza di tre regioni spazialmente coerenti, ciascuna associata a un cluster distinto.

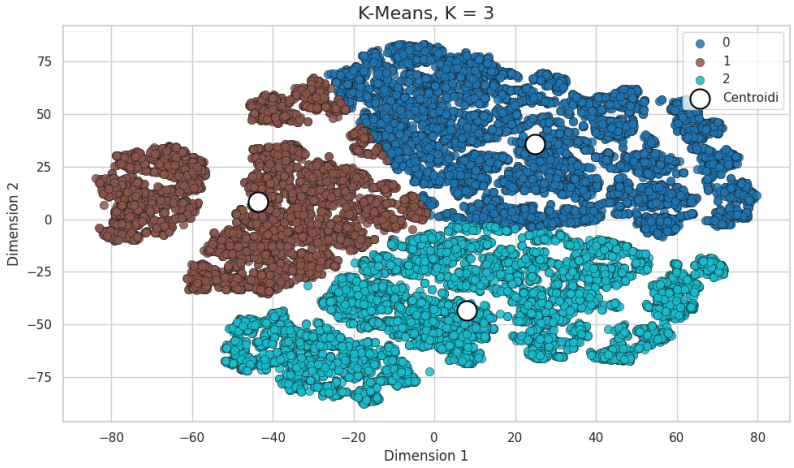

Figura 3.12: Clustering K-Means nello spazio t-SNE con $k = 3$.

I centroidi, riportati in figura, risultano chiaramente separati e collocati in posizioni rappresentative delle rispettive regioni, suggerendo una buona capacità dell'algoritmo di individuare strutture significative nello spazio ridotto.

Per analizzare in modo più approfondito i cluster individuati, è stata condotta una fase di **profilazione dei gruppi**, finalizzata a mettere in evidenza le principali differenze tra i cluster emersi.

L'analisi è stata inizialmente svolta dal punto di vista quantitativo, considerando le **medie standardizzate delle variabili numeriche originali**, riportate nella heatmap di Figura 3.13.

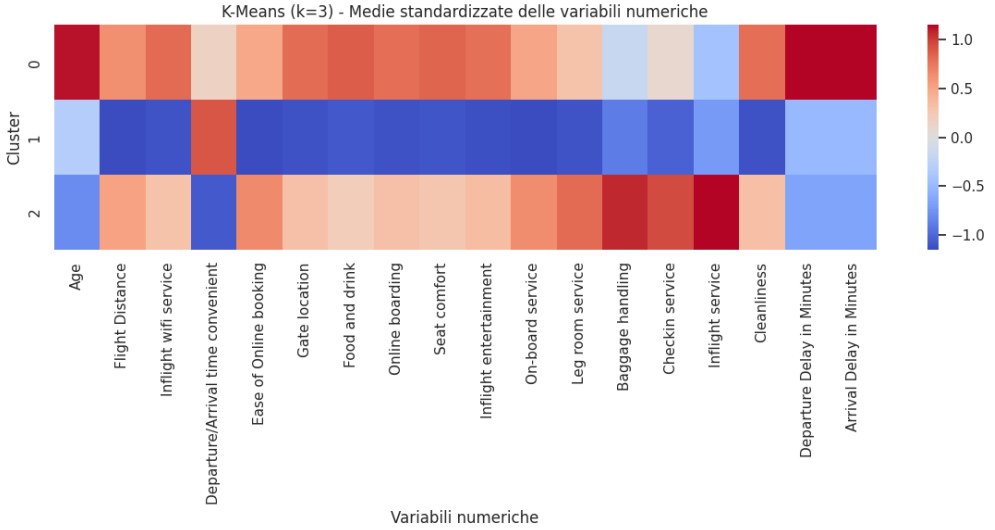

Figura 3.13: Medie standardizzate delle variabili numeriche per ciascun cluster K-Means.

Dalla heatmap emergono profili chiaramente distinti tra i cluster. In particolare:

- il **Cluster 0** presenta valori superiori alla media per età e per i ritardi in partenza e in arrivo;
- il **Cluster 1** mostra valori inferiori alla media per molte variabili di servizio e per la distanza di volo, indicando passeggeri associati a tratte più brevi e a una percezione complessivamente meno positiva del servizio;
- il **Cluster 2** è caratterizzato da valori elevati per variabili operative come *check-in service*, *baggage handling* e *inflight service*, suggerendo una migliore esperienza complessiva.

Per facilitare l'interpretazione dei cluster, è stata inoltre condotta un'analisi delle **variabili più discriminanti** per ciascun gruppo, individuando per ogni cluster le cinque variabili con valore standardizzato più alto e più basso rispetto alla media globale.

I risultati confermano quanto osservato dalla heatmap.

Di seguito viene riportato un esempio di output:

```
Top / bottom variabili per cluster
- Cluster 0
TOP 5 variabili (sopra media globale):
Arrival Delay in Minutes      1.15
Departure Delay in Minutes    1.15
Age                           1.12
Food and drink                0.88
```

```
Seat comfort                  0.84

BOTTOM 5 variabili (sotto media globale):
Inflight service                    -0.43
Baggage handling                    -0.18
Checkin service                      0.09
Departure/Arrival time convenient    0.16
Leg room service                     0.30

Output analoghi sono stati ottenuti per gli altri cluster.
```

Un ulteriore elemento di analisi riguarda la **distribuzione della soddisfazione** nei diversi cluster. In Figura 3.14 è riportata la percentuale di passeggeri soddisfatti e insoddisfatti per ciascun cluster.

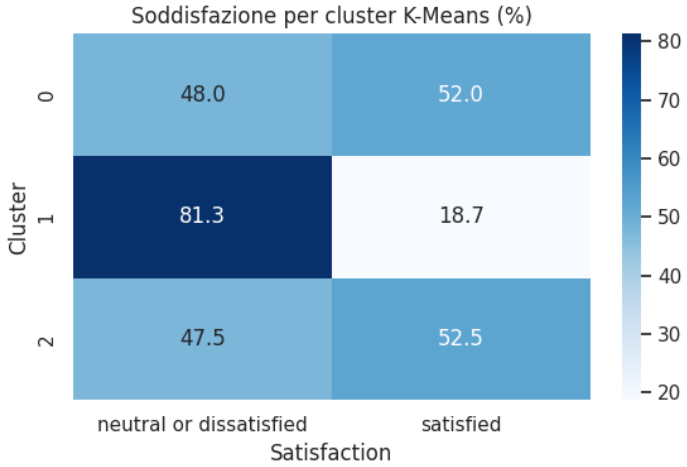

Figura 3.14: Distribuzione percentuale della soddisfazione per cluster.

Il Cluster 1 presenta la quota più elevata di passeggeri insoddisfatti (oltre l'80%), mentre i Cluster 0 e 2 mostrano una distribuzione più equilibrata, con una lieve prevalenza di passeggeri soddisfatti.

Questa tendenza è confermata anche dal livello medio di soddisfazione, mostrato in Figura 3.15, che evidenzia il valore più elevato per il Cluster 2, seguito dal Cluster 0, mentre il Cluster 1 risulta nettamente il meno soddisfatto.

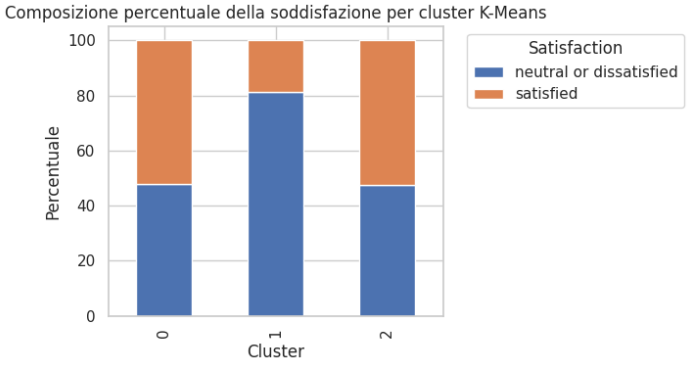

Figura 3.15: Livello medio di soddisfazione per cluster K-Means.

L'analisi è stata infine estesa alla classe di viaggio. Come illustrato in Figura 3.16, il Cluster 1 è composto prevalentemente da passeggeri in classe Economy, mentre i Cluster 0 e 2 presentano una maggiore prevalenza di passeggeri Business.

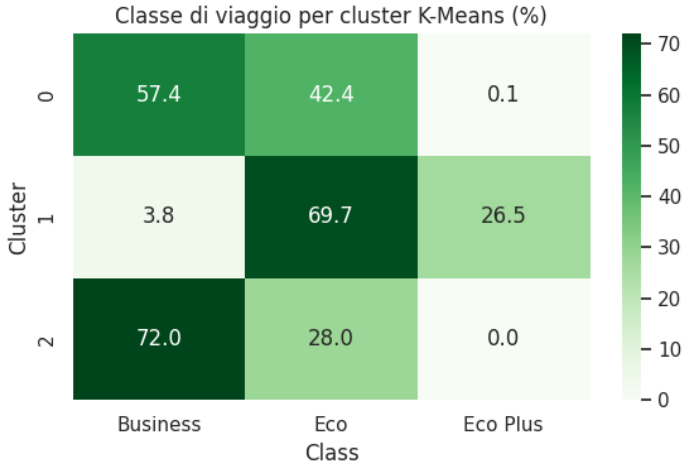

Figura 3.16: Distribuzione della classe di viaggio per cluster.

Infine, la distribuzione della distanza di volo (Figura 3.17) mostra differenze significative tra i cluster, con il Cluster 1 associato a tratte più brevi e i Cluster 0 e 2 caratterizzati da distanze medie superiori e maggiore variabilità.

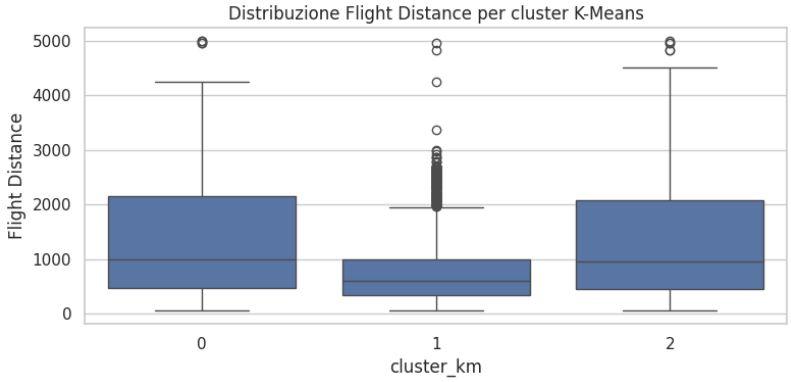

Figura 3.17: Distribuzione della distanza di volo per cluster.

Nel complesso, la fase di profilazione conferma che i cluster individuati mediante K-Means nello spazio t-SNE risultano ben separati dal punto di vista geometrico e presentano caratteristiche operative e di soddisfazione chiaramente differenziate.

3.6 DBSCAN

L'algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) è una tecnica di clustering basata sulla densità, che raggruppa le osservazioni in base alla presenza di regioni ad alta concentrazione di punti. A differenza di K-Means, DBSCAN non richiede di fissare **a priori** il numero di cluster, ma identifica automaticamente i gruppi e può inoltre etichettare come rumore (outlier) i punti che non appartengono a nessuna regione sufficientemente densa.

DBSCAN è governato principalmente da due parametri:

- ε (eps): raggio che definisce il vicinato di un punto;
- min_samples: numero minimo di punti nel vicinato ε affinché un punto venga considerato "core" e possa generare un cluster.

DBSCAN, come K-Means, è stato applicato nello spazio bidimensionale t-SNE, al fine di lavorare su una rappresentazione compatta e più adatta alla ricerca di strutture locali.

3.6.1 Applicazione DBSCAN

Per applicare DBSCAN è stato necessario selezionare opportunamente **eps** e **min_samples**, poiché la scelta di tali parametri influenza direttamente:

- il numero di cluster individuati;

- la % di rumore;
- la qualità complessiva della partizione.

Per tale motivo, inizialmente, è stato calcolato il k-distance plot per un intervallo, dove k=[5, 8, 10, 12, 15, 20],(associati ai candidati min_samples) in modo da individuare un possibile punto di "gomito", utile come indicazione empirica per la scelta di ε.

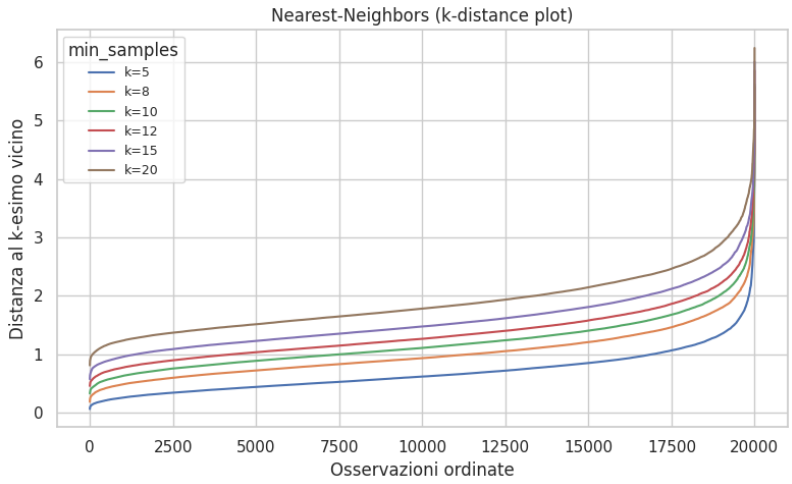

Figura 3.18: k-distance plot per diversi valori di min_samples.

Successivamente, è stata eseguita una ricerca a griglia su più combinazioni di ε e min_samples, monitorando contemporaneamente:

- numero di cluster;
- % di rumore;
- silhouette (calcolata sia su t-SNE, in modo indicativo, sia su PCA 10D come controllo aggiuntivo);
- bilanciamento tra dimensioni dei cluster, per evitare cluster dominante e micro-cluster.

A supporto della scelta sono state prodotte anche le matrici riassuntive (mean noise distance, numero cluster, silhouette score) per visualizzare come cambiano le metriche al variare dei parametri.

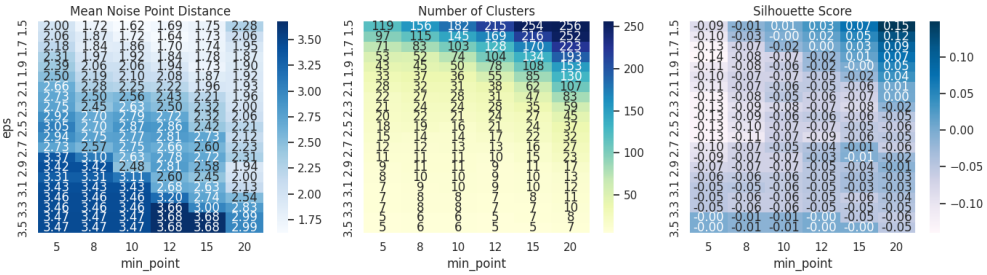

Figura 3.19: Matrice di supporto al tuning DBSCAN (rumore, numero cluster, silhouette).

Sulla base del compromesso tra rumore molto basso, numero di cluster gestibile e separazione accettabile nello spazio t-SNE, è stata selezionata la configurazione:

- $\varepsilon = 2.8$;
- min_samples = 15.

Questa configurazione produce il seguente output:

```
DBSCAN finale con parametri: eps=2.8, min_samples=15
Cluster (senza rumore): 15
Rumore: 0.34%
Silhouette t-SNE (qualitativa): 0.233
```

3.6.2 Analisi dei risultati

L'applicazione finale di DBSCAN nello spazio t-SNE con ε=2.8 e **min_samples**=15 è mostrata in Figura 3.20.

Per rendere la visualizzazione più vicina allo stile di K-Means, sono stati mostrati anche punti rappresentativi dei cluster (ricordiamo che, DBSCAN non ha centroidi veri e propri).

Figura 3.20: Clustering DBSCAN nello spazio t-SNE.

Dai cluster ottenuti, si evince la presenza di cluster più grandi e vari cluster più piccoli coerentemente con la natura dell'algoritmo.

Le varie osservazioni che compongono i cluster sono riportate in Figura Figura 3.21

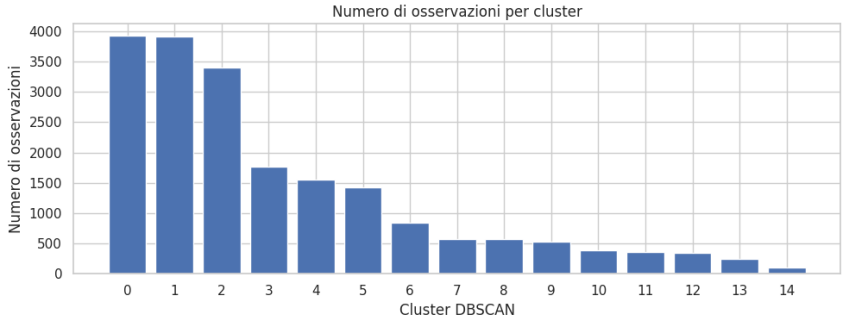

Figura 3.21: Numero di osservazioni per cluster.

Un aspetto chiave, per l'analisi dei risultati, è la distribuzione della soddisfazione nei cluster (Figura 3.22).

Qui emerge una separazione quasi netta tra gruppi "quasi totalmente soddisfatti" e gruppi con prevalenza di insoddisfatti.

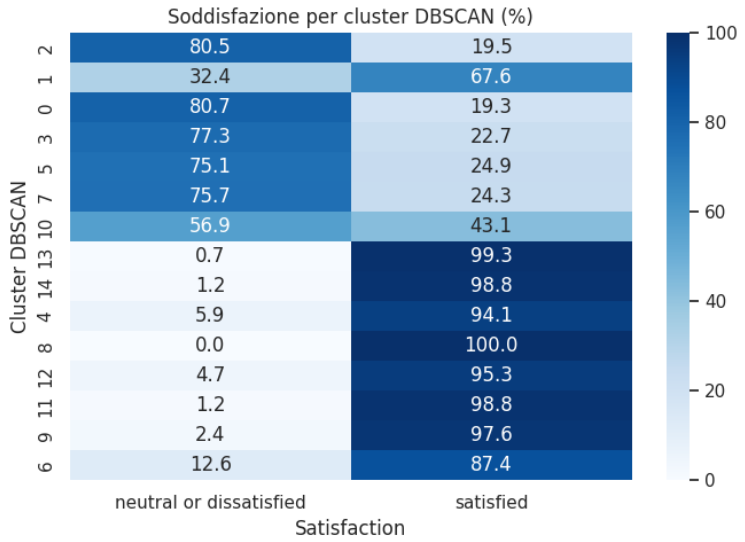

Figura 3.22: Distribuzione percentuale della soddisfazione per cluster DBSCAN.

Il grafico del livello medio di soddisfazione in Figura 3.23 conferma questo pattern.

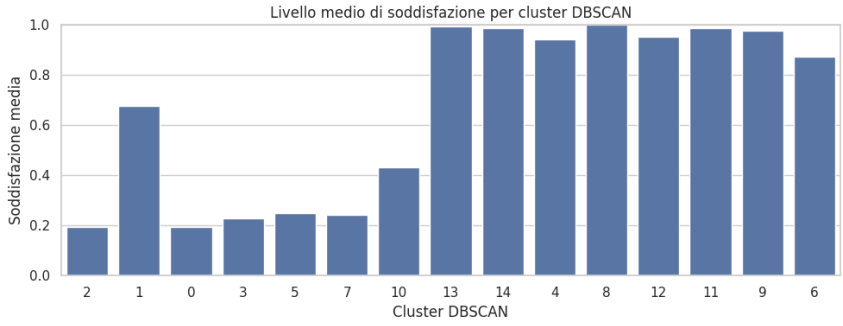

Figura 3.23: Livello medio di soddisfazione per cluster DBSCAN.

Un ulteriore aspetto fondamentale per l'analisi dei risultati è rappresentato dalla classe di viaggio, che risulta fortemente discriminante in diversi cluster (Figura 3.24).

In particolare:

- alcuni cluster sono quasi esclusivamente Business;

- Cluster 7 è totalmente Eco Plus;
- altri cluster sono prevalentemente Economy.

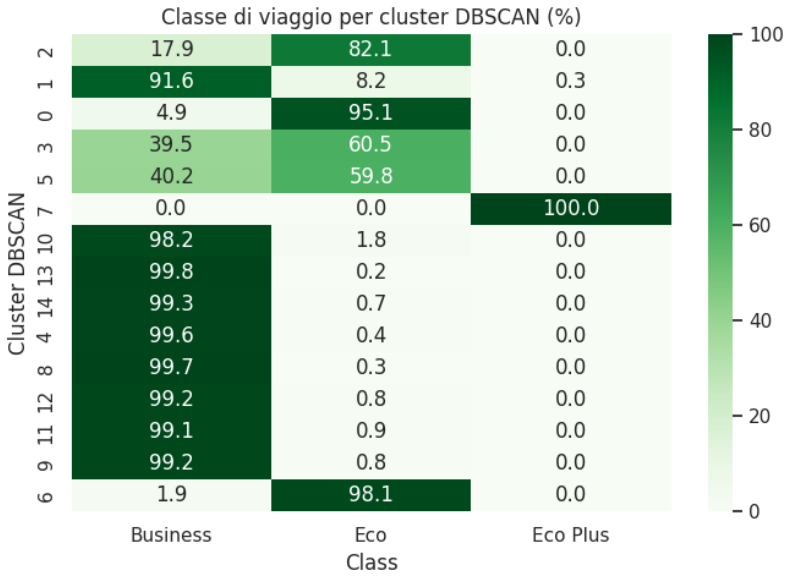

Figura 3.24: Distribuzione percentuale della classe di viaggio per cluster.

Infine, la distanza di volo mostra differenze significative tra i cluster (Figura 3.25). Alcuni gruppi risultano associati a tratte più brevi, altri a tratte mediamente più lunghe e con maggiore variabilità.

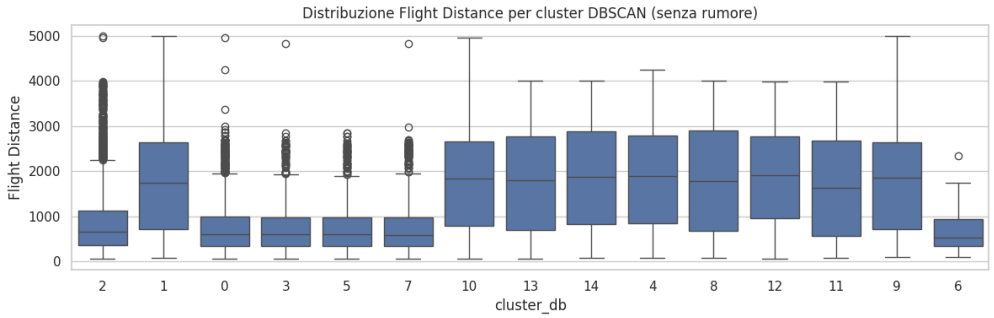

Figura 3.25: Distribuzione della Flight Distance per cluster DBSCAN (senza rumore).

- Il valore di silhouette su t-SNE pari a **0.233** suggerisce una separazione moderata tra cluster nello spazio ridotto;

- La silhouette su PCA 10D risulta invece pari a **-0.001** indicando che la separazione non è particolarmente "pulita" nello spazio PCA.

3.6.3 DBSCAN CON UMAP

Nonostante i risultati ottenuti con DBSCAN su t-SNE siano soddisfacenti, l'uso di t-SNE presenta alcune limitazioni:

- forte enfasi sulle strutture locali;
- sensibilità ai parametri.

Per questi motivi, l'analisi è stata estesa introducendo **UMAP (Uniform Manifold Approximation and Projection)**, un metodo di riduzione dimensionale che preserva meglio la struttura globale dei dati, mantenendo al contempo una buona separazione locale.

In termini intuitivi, UMAP può essere visto come un'evoluzione concettuale rispetto a t-SNE. Mentre t-SNE si concentra quasi esclusivamente sulla preservazione delle relazioni di vicinato più strette, UMAP cerca di mantenere contemporaneamente: (i) le relazioni locali tra osservazioni simili e (ii) una coerenza globale della struttura dei dati.

Per chiarire il motivo del suo utilizzo, si consideri un esempio semplice. Si supponga di avere tre gruppi di passeggeri:

- un gruppo di clienti Business con alta soddisfazione;
- un gruppo di clienti Economy con bassa soddisfazione;
- un gruppo intermedio con caratteristiche miste.

Nello spazio originale ad alta dimensione, questi gruppi risultano distinti ma non completamente separati.

Applicando t-SNE, i punti appartenenti a ciascun gruppo tendono a collassare in insiemi molto compatti e ben separati; tuttavia, le distanze relative tra i gruppi non sono necessariamente informative, poiché t-SNE non preserva la struttura globale dello spazio. Di conseguenza, due cluster lontani nel piano t-SNE non sono necessariamente "più diversi" di due cluster più vicini.

UMAP, invece, costruisce una rappresentazione che tiene conto anche della disposizione complessiva dei gruppi, preservando in modo più fedele le relazioni tra cluster diversi. Questo rende lo spazio UMAP particolarmente adatto come input per algoritmi di clustering basati sulla densità, come DBSCAN, in quanto consente di individuare regioni dense ben separate senza frammentare eccessivamente i gruppi.

UMAP è stato applicato allo spazio PCA a 10 dimensioni, garantendo coerenza con le fasi precedenti.

Il risultato ottenuto, mostrato in Figura 3.26 evidenzia una struttura a gruppi ben separati, indicando che lo spazio UMAP è adatto a un successivo clustering.

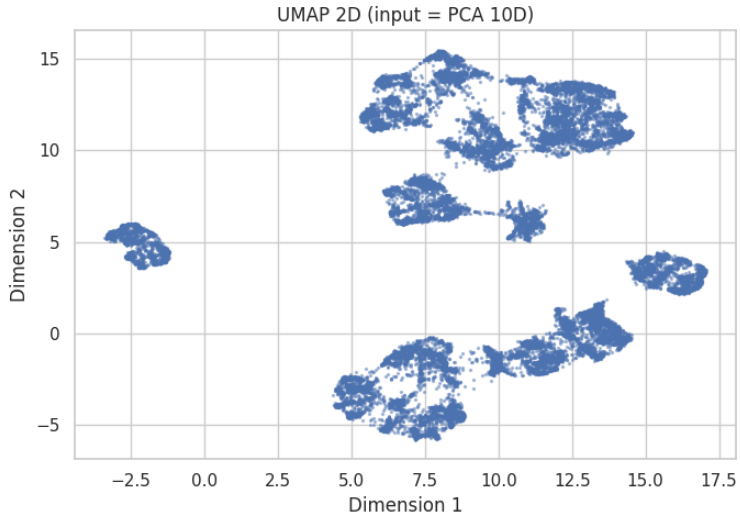

Figura 3.26: Rappresentazione bidimensionale dei dati tramite UMAP.

Anche in questo caso la selezione dei parametri è avvenuta tramite:

- k-distance plot nello spazio UMAP, mostrato in Figura 3.27;
- ricerca a griglia su più combinazioni di ε e min_samples.

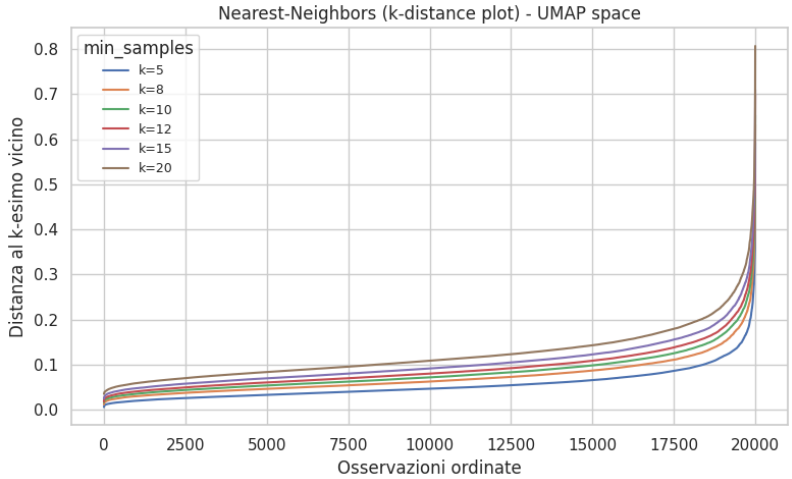

Figura 3.27: K-distance plot per diversi valori di *min_samples*.

Anche in questo caso, sono state prodotte le matrici riassuntive (mean noise distance, numero cluster, silhouette score) per visualizzare come cambiano le metriche al variare dei parametri. Le matrici ottenute sono riportato di seguito, in Figura 3.28

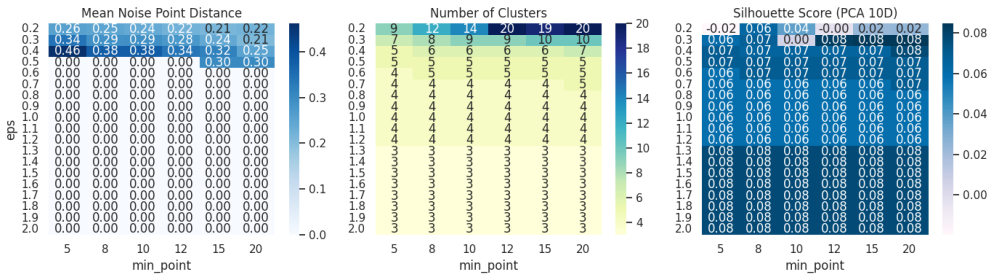

Figura 3.28: Matrice di supporto al tuning DBSCAN (rumore, numero cluster, silhouette).

Il compromesso tra rumore molto basso, numero di cluster e separazione accettabile, ci ha portato a selezionare la configurazione:

- $\varepsilon = 0{,}4$;
- min_samples = 20.

Questa configurazione produce il seguente output:

```
DBSCAN finale (UMAP) con parametri: eps=0.4, min_samples=20
Cluster (senza rumore): 7
Rumore: 0.07%
Silhouette UMAP (indicativa): 0.515
Silhouette PCA 10D: 0.078
```

L'applicazione finale di DBSCAN nello spazio UMAP con i parametri selezionati è mostrata in Figura 3.29.

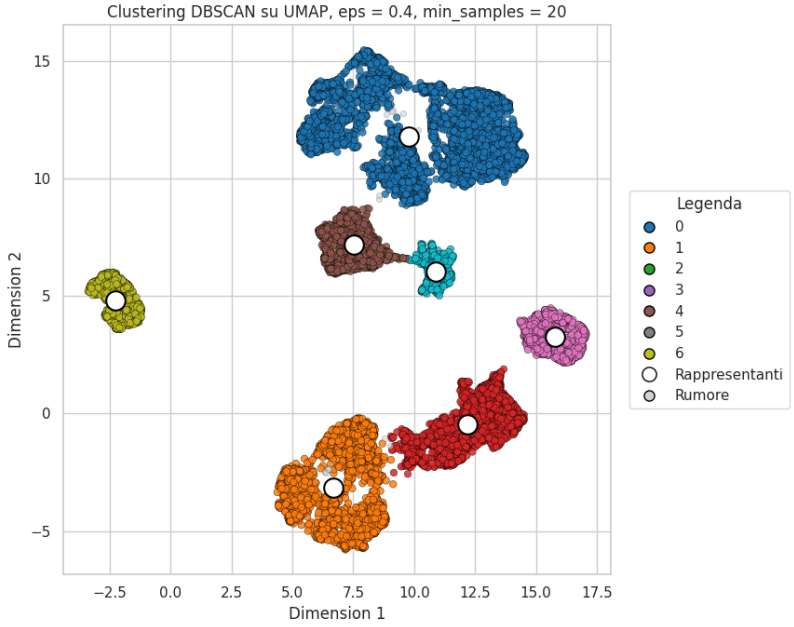

Figura 3.29: Clustering DBSCAN nello spazio UMAP.

Dai cluster ottenuti, si evince la presenza di un cluster principale di circa 7,300 osservazioni e cluster minori comunque sufficientemente popolati.

Il numero di osservazione per cluster è riportato di seguito in Figura 3.30.

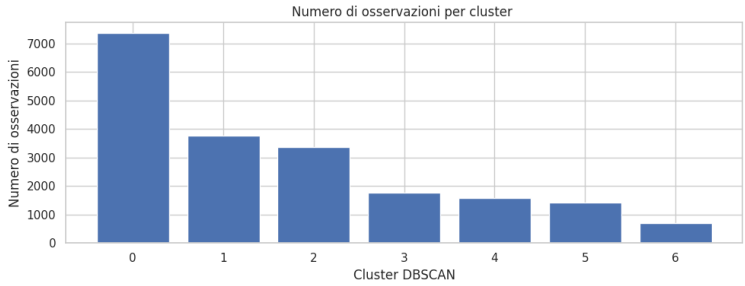

Figura 3.30: Numero di osservazioni per cluster.

Anche per questa clusterizzazione, nell'analisi dei risultati, risulta fondamentale la distribuzione della soddisfazione per cluster (Figura 3.31).

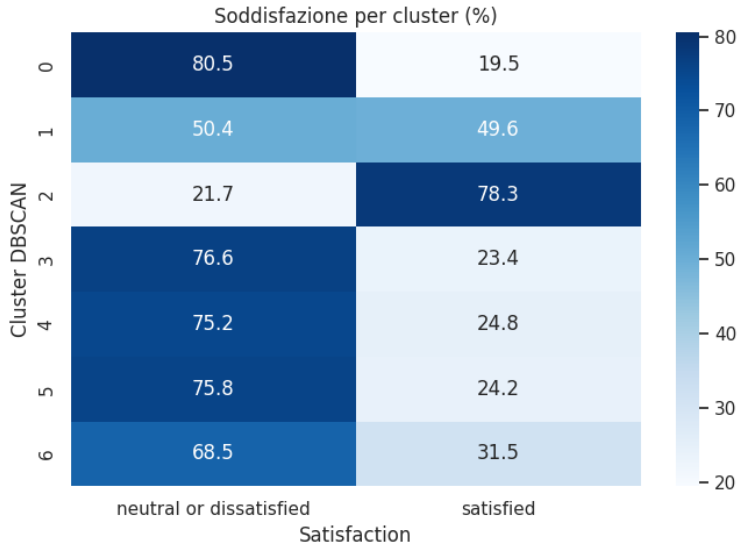

Figura 3.31: Distribuzione percentuale della soddisfazione per cluster (UMAP+DBSCAN).

Dalla distribuzione è possibile notare come:

- Esistono cluster con un livello di soddisfazione nettamente superiore alla media, che rappresentano clientela con un'esperienza di viaggio generalmente positiva (Cluster 2);
- Esistono cluster in cui la soddisfazione e l'insoddisfazione risultano più bilanciate, suggerendo profili intermedi (Cluster 1);
- Esistono cluster in cui prevale l'insoddisfazione (Cluster 0,3,4,5,6).
-

Di seguito, un ulteriore aspetto fondamentale per l'analisi dei risultati è rappresentato dalla classe di viaggio, che mostra pattern altrettanto distinti.
I risultati relativi sono mostrati in Figura 3.32.

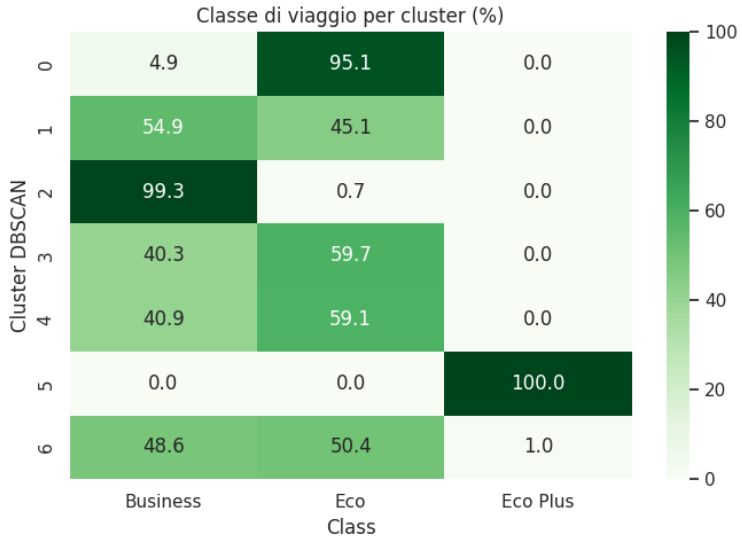

Figura 3.32: Distribuzione della classe di viaggio per cluster (UMAP+DBSCAN).

Dai risultati si può notare come:

- emergono cluster quasi monoclasse (Cluster 0,2,5).