



Università degli Studi di Milano - Bicocca  
Dipartimento di Informatica, Sistemistica e Comunicazione  
Corso di Laurea in Informatica

## Analisi della Mobilità Urbana a Milano Analisi Non-Supervisionata della Mobilità in Sharing

**Relatore:** Prof. Giuseppe Vizzari  
**Correlatore:** Dott.ssa Blerina Spahiu

**Relazione prova finale di:**  
Samuele Carbone  
Matricola 899661

# Contents

|  |           |
|--|-----------|
| <b>1 Introduzione</b>  | <b>7</b>  |
| 1.1 Abstract . . . . .   | 7         |
| 1.2 Contesto e motivazioni . . . . .   | 7         |
| 1.3 Obiettivi . . . . .  | 8         |
| 1.3.1 Obiettivi principali . . . . .   | 8         |
| 1.3.2 Obiettivi del lavoro . . . . .   | 8         |
| 1.4 Struttura della relazione . . . . .  | 9         |
| <b>2 Modelli e approcci abilitanti</b>   | <b>10</b> |
| 2.1 BallTree . . . . .   | 10        |
| 2.2 Apprendimento automatico e apprendimento non supervisionato . . . . .                  | 11        |
| 2.2.1 Tipi di apprendimento automatico . . . . .   | 11        |
| 2.2.2 Apprendimento non supervisionato . . . . .   | 12        |
| 2.2.3 Importanza e applicazioni pratiche . . . . .   | 13        |
| 2.3 Tecniche di clustering . . . . .   | 13        |
| 2.3.1 K-Means . . . . .  | 14        |
| 2.3.2 DBSCAN . . . . .   | 15        |
| 2.3.3 Confronto tra K-Means e DBSCAN . . . . .   | 16        |
| 2.4 Tecniche di riduzione dimensionale . . . . .   | 17        |
| 2.4.1 PCA (Principal Component Analysis) . . . . .   | 17        |
| 2.4.2 t-SNE (t-Distributed Stochastic Neighbor Embedding) . . . . .                        | 18        |
| 2.4.3 UMAP (Uniform Manifold Approximation and Projection) . . . . .                       | 19        |
| 2.5 Reti neurali e Autoencoder . . . . .   | 20        |
| 2.5.1 Introduzione alle reti neurali . . . . .   | 20        |
| 2.5.2 Il deep learning . . . . .   | 20        |
| 2.5.3 Autoencoder . . . . .  | 20        |
| 2.5.4 Varianti degli autoencoder . . . . .   | 20        |
| 2.5.5 Addestramento e regolarizzazione . . . . .   | 21        |
| 2.6 Rilevamento di anomalie . . . . .  | 21        |
| 2.6.1 Introduzione . . . . .   | 21        |
| 2.6.2 Approcci principali . . . . .  | 21        |
| 2.6.3 Autoencoder per il rilevamento di anomalie . . . . .                                 | 22        |
| 2.6.4 Varianti di autoencoder per l'anomaly detection . . . . .                            | 23        |
| 2.6.5 Misure di performance . . . . .  | 23        |
| 2.7 Metriche di valutazione: Silhouette Score, Davies-Bouldin, Calinski-Harabasz . . . . . | 23        |
| 2.7.1 Silhouette Score . . . . .   | 23        |
| 2.7.2 Indice di Davies-Bouldin (DBI) . . . . .   | 23        |
| 2.7.3 Indice di Calinski-Harabasz (CHI) . . . . .  | 24        |
| <b>3 Dataset e preparazione dei dati</b>   | <b>25</b> |
| 3.1 Origine dei dati (Fluctuo, POI, dati temporali e geografici) . . . . .                 | 25        |
| 3.1.1 Dati sui Punti di Interesse (POI) . . . . .  | 25        |
| 3.1.2 Dati Fluctuo . . . . .   | 26        |
| 3.1.3 Dati di trasporto pubblico urbano . . . . .  | 26        |
| 3.2 Costruzione del tensore multidimensionale . . . . .                                    | 28        |
| 3.2.1 Rappresentazione dei Viaggi . . . . .  | 29        |

|          |  |           |
|----------|--|-----------|
| 3.3      | Pulizia e trasformazione delle variabili . . . . .                               | 30        |
| 3.3.1    | Gestione di valori mancanti o anomalie . . . . .                                 | 30        |
| 3.3.2    | Conversione e codifica delle variabili . . . . .                                 | 30        |
| 3.4      | Integrazione dei dataset e operazioni preliminari . . . . .                      | 31        |
| 3.4.1    | Operazioni preliminari . . . . .   | 31        |
| 3.4.2    | Dati Fluctuo . . . . .   | 31        |
| 3.4.3    | Dati POI . . . . .   | 32        |
| 3.4.4    | Integrazione nel dataset . . . . .   | 33        |
| 3.5      | Normalizzazione . . . . .  | 33        |
| <b>4</b> | <b>Architettura e metodologia</b>  | <b>35</b> |
| 4.1      | Pipeline sperimentale . . . . .  | 35        |
| 4.1.1    | Caricamento, preparazione del dataset e normalizzazione . . . . .                | 35        |
| 4.1.2    | Normalizzazione e conversione in tensori . . . . .                               | 35        |
| 4.1.3    | Progettazione dell'Autoencoder . . . . .   | 36        |
| 4.1.4    | Training e Test del modello . . . . .  | 36        |
| 4.1.5    | Rilevamento delle anomalie . . . . .   | 36        |
| 4.1.6    | Estrazione e salvataggio degli embedding . . . . .                               | 36        |
| 4.1.7    | Clustering sugli embedding . . . . .   | 36        |
| 4.1.8    | Riduzione dimensionale per la visualizzazione . . . . .                          | 37        |
| 4.2      | Progettazione dell'Autoencoder . . . . .   | 37        |
| 4.2.1    | Architettura e iperparametri . . . . .   | 37        |
| 4.2.2    | Creazione dei set di training e test . . . . .                                   | 38        |
| 4.2.3    | Training e Test del modello . . . . .  | 39        |
| 4.3      | Rilevamento delle anomalie . . . . .   | 40        |
| 4.4      | Salvataggio degli embedding e valutazione del modello . . . . .                  | 41        |
| 4.4.1    | Valutazione del modello . . . . .  | 41        |
| 4.4.2    | Salvataggio degli embedding . . . . .  | 41        |
| 4.5      | Clustering sugli embedding . . . . .   | 41        |
| 4.5.1    | Metodo di clustering . . . . .   | 41        |
| 4.6      | Tecniche di riduzione dimensionale applicate . . . . .                           | 41        |
| 4.6.1    | UMAP (Uniform Manifold Approximation and Projection) . . . . .                   | 42        |
| 4.6.2    | Esclusione di t-SNE . . . . .  | 42        |
| 4.6.3    | Esclusione di PCA . . . . .  | 42        |
| <b>5</b> | <b>Risultati sperimentali</b>  | <b>43</b> |
| 5.1      | Andamento del training (MSE e loss) . . . . .                                    | 43        |
| 5.1.1    | Addestramento con Regolarizzazione L1/L2 . . . . .                               | 43        |
| 5.1.2    | Addestramento senza regolarizzazione L1/L2 . . . . .                             | 44        |
| 5.1.3    | Spiegazione della loss e correlazione della regolarizzazione L1/L2 . . . . .     | 46        |
| 5.2      | Ricostruzione e visualizzazione dell'errore . . . . .                            | 47        |
| 5.3      | Identificazione delle osservazioni con errore di ricostruzione elevato . . . . . | 48        |
| 5.3.1    | Definizione della soglia di anomalia . . . . .                                   | 48        |
| 5.3.2    | Analisi delle features con errore elevato . . . . .                              | 48        |
| 5.3.3    | Visualizzazione nello spazio latente . . . . .                                   | 48        |
| 5.4      | Valutazione tramite metriche geometriche . . . . .                               | 49        |
| 5.5      | Clustering dei dati . . . . .  | 50        |
| 5.5.1    | Ottenimento degli embeddings . . . . .   | 51        |
| 5.6      | Clustering su dati ridotti con UMAP . . . . .                                    | 51        |
| 5.6.1    | Introduzione . . . . .   | 52        |
| 5.6.2    | Cluster 2 . . . . .  | 53        |
| 5.6.3    | Cluster 9 . . . . .  | 56        |
| 5.6.4    | Analisi comparativa dei cluster 2 e 9 . . . . .                                  | 60        |
| 5.7      | Analisi per numero di cluster (14, 50, 100, 500, 1000, 2000) . . . . .           | 62        |
| 5.8      | Analisi con k = 14 . . . . .   | 63        |
| 5.9      | Analisi con k = 50 . . . . .   | 63        |
| 5.9.1    | Introduzione . . . . .   | 64        |
| 5.9.2    | Cluster 38: viaggi anomali durante i giorni feriali . . . . .                    | 64        |
| 5.9.3    | Cluster 39: spostamenti concentrati nei giorni festivi e weekend . . . . .       | 65        |

|          |  |           |
|----------|--|-----------|
| 5.9.4    | Confronto e interpretazione . . . . .                                | 65        |
| 5.10     | Analisi con $k = 100$ . . . . .                                      | 65        |
| 5.10.1   | Cluster 20: distribuzione settimanale e festiva . . . . .            | 66        |
| 5.10.2   | Cluster 20: distribuzione mensile e giornaliera . . . . .            | 66        |
| 5.10.3   | Cluster 20: fascia oraria . . . . .                                  | 66        |
| 5.10.4   | Cluster 20: durata degli spostamenti . . . . .                       | 67        |
| 5.10.5   | Cluster 20: interpretazione complessiva . . . . .                    | 67        |
| 5.11     | Analisi con $k = 500$ . . . . .                                      | 67        |
| 5.11.1   | Cluster 50: spostamenti pendolari mattutini con bicicletta . . . . . | 68        |
| 5.11.2   | Cluster 237: mobilità extraurbana o periferica domenicale . . . . .  | 68        |
| 5.11.3   | Cluster 56: mobilità urbana serale con scooter . . . . .             | 69        |
| 5.11.4   | Confronto tra i cluster . . . . .                                    | 70        |
| 5.12     | Analisi con $k = 1000$ . . . . .                                     | 70        |
| 5.12.1   | Cluster 810 . . . . .  | 71        |
| 5.12.2   | Cluster 328 . . . . .  | 72        |
| 5.13     | Analisi con $k = 2000$ . . . . .                                     | 73        |
| 5.13.1   | Cluster 231 . . . . .  | 73        |
| 5.13.2   | Cluster 511 . . . . .  | 74        |
| 5.13.3   | Confronto tra i cluster 231 e 511 . . . . .                          | 75        |
| 5.14     | Conclusioni finali . . . . .   | 77        |
| 5.14.1   | Aspettative . . . . .  | 77        |
| 5.15     | Discussione e comparazione dei risultati . . . . .                   | 78        |
| <b>6</b> | <b>Conclusioni e sviluppi futuri</b>                                 | <b>80</b> |
| 6.1      | Valutazione dei metodi adottati . . . . .                            | 80        |
| 6.2      | Riflessioni conclusive . . . . .                                     | 80        |
| 6.3      | Limiti e criticità dell'approccio . . . . .                          | 81        |
| 6.4      | Implicazioni per l'analisi della mobilità urbana . . . . .           | 81        |
| 6.5      | Sviluppi futuri . . . . .  | 81        |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Algoritmo Balltree . . . . .   | 11 |
| 2.2  | Artificial intelligence, machine learning, and deep learning [7] . . . . .   | 11 |
| 2.3  | Esempi di compiti comuni nell'apprendimento automatico: classificazione, regressione, clustering e apprendimento semisupervisionato. . . . .   | 12 |
| 2.4  | Visualizzazione e rilevamento di pattern: t-SNE e identificazione di anomalie. . . . .   | 13 |
| 2.5  | Algoritmo K-Means illustrato: ad ogni iterazione vengono aggiornate le assegnazioni ai cluster e ricalcolati i centroidi. . . . .  | 14 |
| 2.6  | Clustering DBSCAN usando due diversi <code>neighborhood radiuses</code> . . . . .  | 16 |
| 2.7  | Architettura generale di un autoencoder: l'encoder comprime l'input in uno spazio latente, il decoder lo ricostruisce. . . . .   | 21 |
| 2.8  | Principali varianti di autoencoder: denoising, sparse, variazionale e convoluzionale. . . . .  | 22 |
| 3.1  | Cartografia delle infrastrutture per i trasporti generata a partire dai dati della <i>Overture Maps Foundation</i> . . . . .   | 26 |
| 3.2  | Azienda Fluctuo . . . . .  | 27 |
| 3.3  | Mappa delle linee di Superficie . . . . .  | 28 |
| 3.4  | Mappa delle linee metropolitane . . . . .  | 28 |
| 3.5  | Mappa delle linee ferroviarie . . . . .  | 29 |
| 4.1  | Autoencoder creato e utilizzato. . . . .   | 35 |
| 4.2  | Autoencoder creato e utilizzato. . . . .   | 39 |
| 5.1  | <code>Loss</code> di addestramento con regolarizzazione L1/L2 . . . . .  | 44 |
| 5.2  | <code>MSE</code> di addestramento con regolarizzazione L1/L2 . . . . .   | 44 |
| 5.3  | <code>loss</code> di addestramento con regolarizzazione L1/L2 . . . . .  | 45 |
| 5.4  | <code>MSE</code> di addestramento con regolarizzazione L1/L2 . . . . .   | 46 |
| 5.5  | Distribuzione completa degli errori di ricostruzione. . . . .  | 47 |
| 5.6  | features con il maggior errore medio assoluto tra valori originali e ricostruiti. . . . .  | 49 |
| 5.7  | Distribuzione delle osservazioni con errore elevato nel piano UMAP. . . . .  | 50 |
| 5.8  | Visualizzazione dei dati tramite UMAP in 2D e 3D . . . . .   | 52 |
| 5.9  | Distribuzione temporale per i cluster 2 e 9 ( $k = 12$ ). . . . .  | 61 |
| 5.10 | Distribuzione temporale per i cluster 38 e 39 ( $k = 50$ ). . . . .  | 66 |
| 5.11 | Distribuzione delle ore di partenza per il cluster 20 ( $k = 100$ ). La maggior parte degli spostamenti avviene tra mezzanotte e le prime ore del mattino, con un picco tra le 0 : 00 e le 2 : 00. . . . . | 67 |
| 5.12 | Distribuzione temporale per i cluster 50, 56 e 237 ( $k = 500$ ). . . . .  | 70 |
| 5.13 | Mappa geografica dei cluster 50, 56 e 237 ( $k = 500$ ). . . . .   | 71 |
| 5.14 | Distribuzione temporale per i cluster 328 e 810 ( $k = 1000$ ). . . . .  | 72 |
| 5.15 | Distribuzione temporale per i cluster 231 e 511 ( $k = 2000$ ). . . . .  | 76 |
| 5.16 | Mappa geografica dei cluster 231 e 511 ( $k = 2000$ ). . . . .   | 77 |

# List of Tables

|      |  |    |
|------|--|----|
| 2.1  | Confronto tra K-Means e DBSCAN [2, pp. 248, 258] . . . . .   | 17 |
| 4.1  | Sintesi della pipeline sperimentale adottata . . . . .   | 37 |
| 5.1  | Top 10 features con il maggiore errore assoluto medio nelle osservazioni con errore di ricostruzione elevato . . . . . | 48 |
| 5.2  | Valori di valutazione di metriche geometriche . . . . .  | 50 |
| 5.3  | Statistiche descrittive delle caratteristiche temporali del cluster 2 . . . . .  | 54 |
| 5.4  | Statistiche descrittive sull'utilizzo dei veicoli del cluster 2 . . . . .  | 55 |
| 5.5  | Caratteristiche geospaziali del cluster 2 . . . . .  | 55 |
| 5.6  | Caratteristiche delle zone di partenza e arrivo cluster 2 . . . . .  | 56 |
| 5.7  | Statistiche descrittive delle caratteristiche temporali del cluster 9 . . . . .  | 58 |
| 5.8  | Statistiche descrittive sull'utilizzo dei veicoli del cluster 9 . . . . .  | 59 |
| 5.9  | Caratteristiche geospaziali del cluster 9 . . . . .  | 59 |
| 5.10 | Caratteristiche delle zone di partenza e arrivo cluster 9 . . . . .  | 60 |
| 5.11 | Corrispondenza del cluster 2 con e senza l'utilizzo di UMAP . . . . .  | 63 |
| 5.12 | Corrispondenza del cluster 9 con e senza l'utilizzo di UMAP . . . . .  | 63 |
| 5.13 | Statistiche descrittive delle caratteristiche temporali del cluster 38 . . . . .                                       | 64 |
| 5.14 | Statistiche descrittive delle caratteristiche temporali del cluster 39 . . . . .                                       | 65 |
| 5.15 | Statistiche descrittive delle caratteristiche temporali del cluster 20 . . . . .                                       | 68 |
| 5.16 | Caratteristiche geospaziali del cluster 50 . . . . .   | 69 |
| 5.17 | Caratteristiche geospaziali del cluster 237 . . . . .  | 69 |
| 5.18 | Caratteristiche geospaziali del cluster 56 . . . . .   | 69 |
| 5.19 | Caratteristiche geospaziali del cluster 810 . . . . .  | 71 |
| 5.20 | Caratteristiche geospaziali del cluster 328 . . . . .  | 72 |
| 5.21 | Caratteristiche geospaziali del cluster 231 . . . . .  | 74 |
| 5.22 | Caratteristiche geospaziali del cluster 511 . . . . .  | 75 |
| 5.23 | Sintesi dei risultati ottenuti in base al numero di cluster $k$ . . . . .  | 78 |

## Ringraziamenti

Esprimo i miei più profondi ringraziamenti al **Prof. Giuseppe Vizzari** e alla **Dott.ssa Blerina Spahiu** per avermi offerto l'opportunità di svolgere questo stage, fornendomi tutto il materiale e la strumentazione necessari per lavorare nelle migliori condizioni possibili.

Ringrazio la mia famiglia per il supporto e l'incoraggiamento costante, che mi hanno permesso di proseguire il mio percorso di studi presso l'*Università degli Studi di Milano-Bicocca*.

Un sentito grazie va anche ai miei compagni di corso e amici, per il loro aiuto e la condivisione di questo cammino universitario.

Desidero inoltre ringraziare persone a me care, come **Maddalena Molinari**, **Alessia Zavettieri**, **Ludovico Merlo**, **Daniele Buser**, **Stefano Brighenti**, **Francesco Bianchi**, **Andrea Consonni**, **Daniele Besozzi** e tanti altri, per i consigli, gli spunti e il sostegno ricevuto durante il lavoro e nello studio presso l'*Università degli Studi di Milano-Bicocca*.

Infine, ringrazio ancora i miei compagni per i momenti divertenti e spensierati trascorsi insieme, che hanno contribuito a creare un clima sereno e stimolante durante la mia permanenza presso l'Istituto universitario.

# Chapter 1

## Introduzione

### 1.1 Abstract

In questa tesi viene analizzata la mobilità urbana nella città di Milano, prestando una particolare attenzione alla mobilità in sharing, come monopattini, biciclette, scooter e automobili. L'obiettivo principale è quello di identificare gruppi di spostamenti simili e comprenderne le caratteristiche e le motivazioni, utilizzando tecniche di apprendimento non supervisionato.

Dopo una fase iniziale di preparazione e integrazione dei dati, è stato sviluppato un Autoencoder. Il fine era ottenere una riduzione non lineare delle dimensioni, seguita da un'ulteriore riduzione effettuata dall'algoritmo di riduzione di dimensionalità UMAP per la visualizzazione bidimensionale e tridimensionale; Infine, è stato applicato l'algoritmo di clustering **K-Means**. I risultati ottenuti hanno permesso di individuare diversi gruppi di viaggi caratterizzati da specifici comportamenti in termini di fasce orarie, mezzi utilizzati e aree geografiche coinvolte.

Questi risultati offrono una prospettiva utile per comprendere meglio le modalità di utilizzo della mobilità condivisa e per supportare decisioni strategiche nella pianificazione urbana e nella gestione dei servizi di mobilità.

### 1.2 Contesto e motivazioni

Negli ultimi anni, le città moderne si sono confrontate con sfide sempre più complesse legate alla mobilità urbana, tra cui la congestione stradale, l'inquinamento atmosferico e la disuguaglianza nell'accesso ai trasporti. In questo contesto, i servizi di mobilità in sharing stanno emergendo come una valida alternativa ai mezzi di trasporto privati, offrendo soluzioni flessibili, sostenibili e adattabili ai bisogni dei cittadini.

La città di Milano rappresenta un caso di studio particolarmente rilevante, grazie alla diffusione capillare dei servizi di mobilità condivisa e alla disponibilità di dati open-access relativi agli spostamenti. Tuttavia, l'elevato volume e la complessità di tali dati rendono necessaria l'adozione di strumenti analitici avanzati per estrarre informazioni significative.

In questo ambito, le tecniche di apprendimento automatico, in particolare quelle non supervisionate, si rivelano particolarmente adatte, poiché permettono di individuare pattern nascosti nei dati senza la necessità di etichette o supervisione umana.

In parallelo, la crescente disponibilità di dati geolocalizzati e l'evoluzione delle tecniche computazionali hanno trasformato profondamente il modo in cui viene studiata la mobilità urbana. Non ci si limita più a considerare la distanza o la presenza di mezzi pubblici, ma si cerca di comprendere come i quartieri, le attività presenti e perfino la percezione dei luoghi influenzino gli spostamenti delle persone.

Uno degli studi più significativi in questo ambito è quello di Park et al. [12], che propongono il framework MobInsight. Questo sistema consente di interpretare la mobilità urbana in modo dettagliato, considerando non solo quanto le persone si spostano, ma anche perché lo fanno. Gli autori hanno integrato diverse fonti di dati, tra cui annotazioni degli utenti, dati open access comunali e, soprattutto, i log telefonici (CDR) della città di Barcellona (oltre 35 milioni di registrazioni in un mese), per tracciare gli spostamenti

reali. Con queste informazioni, hanno costruito profili semantici dei quartieri e utilizzato una rete neurale per analizzare come determinate caratteristiche (ad esempio, la presenza di negozi artigianali o musei) influenzino i flussi di mobilità. Il punto di forza del lavoro risiede nella capacità di spiegare la mobilità urbana non solo in termini quantitativi, ma anche qualitativi e semantici.

Un'altra prospettiva interessante è quella offerta da Martí et al. [10], che affrontano il tema della definizione dei confini dei quartieri urbani. Spesso, infatti, tali suddivisioni si basano su criteri amministrativi che non riflettono la realtà urbana vissuta quotidianamente. Lo studio propone invece una classificazione basata sulle attività effettivamente presenti sul territorio, utilizzando dati di Google Places. Applicato alla città di Alicante, questo approccio ha permesso di identificare “cluster funzionali”, ovvero aree caratterizzate da una similarità nei servizi, nelle attività commerciali e nelle funzioni svolte. Dopo una fase di pulizia e ricategorizzazione dei dati, gli autori hanno applicato un clustering spaziale per delineare questi nuovi quartieri funzionali, offrendo una lettura più aderente al tessuto urbano e alle dinamiche di mobilità tra le zone.

Un ulteriore contributo rilevante proviene dagli Stati Uniti, dove Lin e Long [9] hanno indagato il rapporto tra tipo di quartiere e comportamento di viaggio delle famiglie. Lo studio, basato sui dati del censimento (CTPP 2000) e del National Household Travel Survey (2001), ha classificato i quartieri in dieci tipologie, utilizzando 64 variabili socio-demografiche (tra cui densità abitativa, età media, reddito e numero di veicoli). I risultati sono stati poi confrontati con cinque indicatori di mobilità: numero di viaggi giornalieri, modalità di trasporto, distanza percorsa, tempo impiegato e chilometri percorsi in auto. Lo studio ha evidenziato come le caratteristiche del quartiere influenzino profondamente le abitudini di spostamento. Ad esempio, la presenza di un buon servizio di trasporto pubblico aumenta l'utilizzo di mezzi collettivi, anche tra chi possiede un'auto.

Questi tre studi, pur affrontando il tema da prospettive diverse, mettono in luce l'importanza di superare le analisi tradizionali della mobilità, fondate esclusivamente su distanze e flussi. Diventa sempre più evidente che i motivi, le condizioni socio-economiche e le caratteristiche del contesto urbano giocano un ruolo centrale. L'impiego di tecniche come il clustering, l'integrazione di dati geo-sociali e l'analisi semantica dei luoghi offre strumenti potenti per comprendere in modo più profondo e sfaccettato i fenomeni di mobilità. In questa tesi, si seguirà tale direzione, sviluppando un'analisi che non si limiti a osservare i flussi, ma che ne indagini anche le cause e il significato spaziale.

## 1.3 Obiettivi

### 1.3.1 Obiettivi principali

L'obiettivo principale della tesi è analizzare i dati relativi alla mobilità urbana in sharing nella città di Milano, con lo scopo di individuare pattern ricorrenti e comportamenti emergenti. In particolare, si mira a:

- Identificare gruppi omogenei di spostamenti tramite tecniche di clustering.
- Ridurre la dimensionalità dei dati in modo non lineare per facilitarne l'analisi e la visualizzazione.
- Comprendere le caratteristiche temporali e geografiche dei cluster ottenuti.

### 1.3.2 Obiettivi del lavoro

Per raggiungere l'obiettivo generale, il lavoro è stato suddiviso in diversi sotto-obiettivi operativi:

- Raccogliere, pulire e integrare i dati provenienti da fonti eterogenee (Fluctuo, POI, trasporto pubblico).
- Costruire un tensore multidimensionale rappresentativo degli spostamenti.
- Progettare e addestrare un Autoencoder al fine di ottenere una rappresentazione compatta e significativa dei dati.
- Applicare tecniche di clustering (K-Means) e di riduzione dimensionale (UMAP).
- Analizzare e interpretare i cluster ottenuti, evidenziando comportamenti caratteristici.

## 1.4 Struttura della relazione

La tesi è articolata nei seguenti capitoli:

- **Capitolo 2 – Modelli e approcci abilitanti:** introduce i concetti principali dell'apprendimento non supervisionato, delle tecniche di clustering e di riduzione dimensionale, con particolare attenzione a K-Means, UMAP e Autoencoder.
- **Capitolo 3 – Dataset e preparazione dei dati:** descrive le fonti dei dati, il processo di integrazione, trasformazione e normalizzazione, e la costruzione del tensore multidimensionale.
- **Capitolo 4 – Architettura e metodologia:** illustra la pipeline sperimentale sviluppata, la progettazione dell'Autoencoder, le tecniche di riduzione dimensionale e i metodi di clustering adottati.
- **Capitolo 5 – Risultati sperimentali:** riporta i risultati ottenuti, con visualizzazioni e analisi quantitative dei cluster individuati.
- **Capitolo 6 – Conclusioni e sviluppi futuri:** riassume i principali risultati, discute i limiti del lavoro e propone possibili estensioni future.

# Chapter 2

## Modelli e approcci abilitanti

### 2.1 BallTree

`Balltree` è una struttura dati sofisticata utilizzata in Python per organizzare e ricercare in modo efficiente dati multidimensionali [8].

Gli algoritmi `Balltree` sono implementati tramite le librerie Python come `Scikit-learn` al fine di permettere un potente strumento di ricerca su diverse dimensioni o caratteristiche del dataset [13]

Il `Balltree` è una struttura usata per organizzare dati in spazi con molte dimensioni, utile in ambiti come la geometria computazionale e l'apprendimento automatico. Funziona dividendo ricorsivamente i dati in gruppi racchiusi dentro ipersfere (cioè sfere in spazi multidimensionali). Ogni nodo dell'albero rappresenta una di queste sfere: quelli interni contengono altri gruppi, mentre le foglie contengono piccoli insiemi di punti.

La costruzione dell'albero avviene scegliendo un punto centrale e creando una sfera attorno ad esso. Si racchiude un sottoinsieme di dati e grazie a questa struttura gerarchica, è possibile effettuare ricerche. Le ricerche sono più veloci, ad esempio per trovare i punti più vicini, evitando di controllare tutti i dati uno per uno. Il Ball Tree è particolarmente efficace con dati ad alta dimensionalità ed è utilizzato in applicazioni come il clustering, classificazione e ricerche di vicini più prossimi.

Il BallTree crea una struttura ad albero in cui ogni nodo contiene una sfera (o ipersfera) che racchiude un gruppo di punti (vd. figura 2.1).

- Ogni nodo dell'albero ha due figli e rappresenta un sottoinsieme dei dati.
- Ogni sfera è definita da un centro (punto centrale) e un raggio (quanto si estende la sfera).
- I dati vengono divisi ricorsivamente (cioè più volte) in due gruppi, fino ad arrivare a gruppi piccoli nelle foglie.

Le componenti principali per gli algoritmi che usufruiscono di `Balltree` sono:

- **Ipersfera:** è come una sfera, ma in spazi con più di 3 dimensioni. Contiene tutti i punti entro una certa distanza dal centro.
- **Nodo:** rappresenta una sfera che contiene un gruppo di punti.
- **Nodo radice:** è il nodo iniziale, che copre tutti i dati.
- **Nodo foglia:** è l'ultimo livello dell'albero e contiene un piccolo gruppo di punti.
- **Suddivisione:** i dati vengono divisi per creare sfere più piccole e organizzate meglio.
- **Struttura gerarchica:** l'albero ha una forma “a rami”, dove ogni nodo ha due figli.
- **Distanze:** per calcolare i vicini si usano metriche come la distanza euclidea o Manhattan.
- **Ricerca del vicino:** l'albero permette di saltare sfere intere che non contengono punti più vicini del migliore trovato finora.

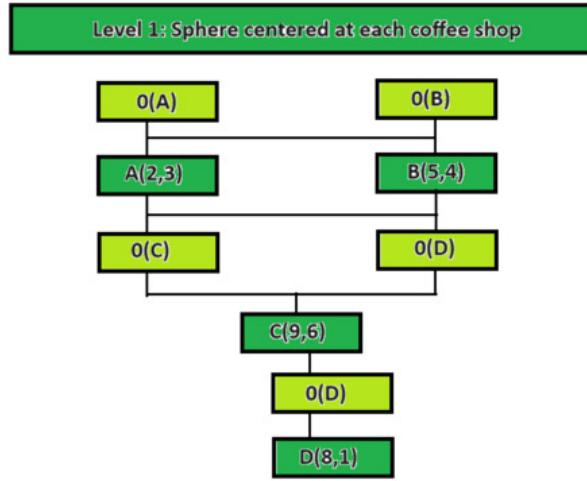


Figure 2.1: Algoritmo Balltree

- **Bilanciamento:** è importante che l'albero sia ben distribuito, così le ricerche saranno più veloci.
- **Costruzione:** si crea l'albero dividendo i dati e assegnando le sfere in modo efficiente.

## 2.2 Apprendimento automatico e apprendimento non supervisionato

Negli ultimi decenni, l'apprendimento automatico (Machine Learning) ha assunto un ruolo sempre più centrale nella ricerca scientifica, nello sviluppo tecnologico e nelle applicazioni industriali. Questo campo dell'intelligenza artificiale si occupa della costruzione di algoritmi e modelli matematici in grado di apprendere automaticamente dai dati, migliorando progressivamente le proprie prestazioni su un compito specifico senza che sia necessario programmare esplicitamente ogni comportamento del sistema.

Secondo Géron (2019), “*un sistema di apprendimento automatico è in grado di imparare da esempi e dati, individuando strutture o regole statistiche implicite, spesso troppo complesse per essere programmate manualmente*” [2]. Ciò differenzia profondamente l'apprendimento automatico dalla programmazione tradizionale, che si basa su regole predefinite e codificate da un essere umano. In quest'ottica, il Machine Learning rappresenta un nuovo paradigma computazionale, dove è il modello stesso a definire le regole operative sulla base dell'esperienza fornita dai dati [7] (vd. figura 2.2).

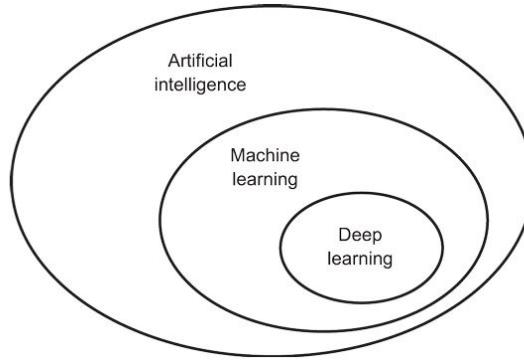
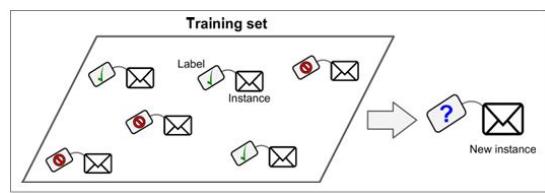


Figure 2.2: Artificial intelligence, machine learning, and deep learning [7]

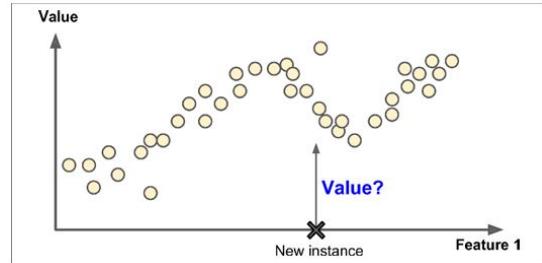
### 2.2.1 Tipi di apprendimento automatico

Il machine learning può essere suddiviso in varie categorie, sulla base della natura dei dati forniti in fase di addestramento. Le tre principali forme sono:

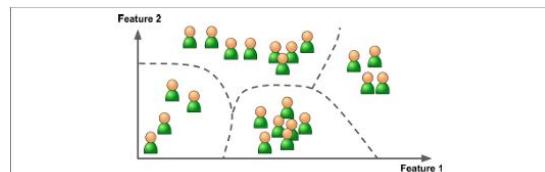
- **Apprendimento supervisionato (supervised learning)**: si utilizza quando i dati di addestramento sono etichettati, ovvero ogni esempio di input è associato a un output noto. L'obiettivo del modello è apprendere una funzione che mappi correttamente gli input negli output desiderati. Tipici esempi includono la classificazione (vd. figura 2.3a) (ad esempio distinguere email di spam da quelle lecite) e la regressione (vd. figura 2.3b) (prevedere il prezzo di una casa).
- **Apprendimento non supervisionato (unsupervised learning)**: viene impiegato quando i dati non sono etichettati. In questo contesto, l'obiettivo del modello è quello di esplorare la struttura latente dei dati, ad esempio raggruppando gli esempi simili tra loro (clustering), riducendo la dimensionalità o rilevando anomalie (vd. figura 2.3c).
- **Apprendimento semi-supervisionato e apprendimento per rinforzo** completano il quadro, integrando rispettivamente pochi dati etichettati in grandi set non etichettati, oppure un agente che interagisce con un ambiente e apprende attraverso ricompense (vd. figura 2.3d).



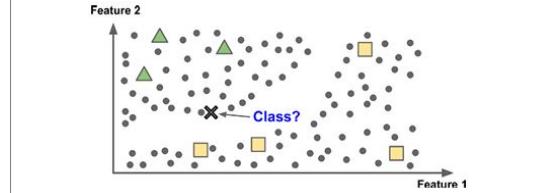
(a) Un insieme di addestramento etichettato per la classificazione dello spam (un esempio di apprendimento supervisionato) [2]



(b) Un problema di regressione: prevedere un valore dato un attributo in ingresso (di solito ci sono più attributi in ingresso, e talvolta anche più valori in uscita) [2]



(c) Clustering [2]



(d) Apprendimento semisupervisionato con due classi (triangoli e quadrati): gli esempi non etichettati (cerchi) aiutano a classificare una nuova istanza (la croce) nella classe dei triangoli piuttosto che in quella dei quadrati, anche se è più vicina ai quadrati etichettati [2].

Figure 2.3: Esempi di compiti comuni nell'apprendimento automatico: classificazione, regressione, clustering e apprendimento semisupervisionato.

## 2.2.2 Apprendimento non supervisionato

L'apprendimento non supervisionato rappresenta una sfida più aperta rispetto a quello supervisionato, in quanto il sistema non dispone di indicazioni esplicite sugli output corretti da apprendere. Di conseguenza, l'algoritmo deve cercare di individuare pattern ricorrenti, strutture latenti o regolarità statistiche nei dati in ingresso.

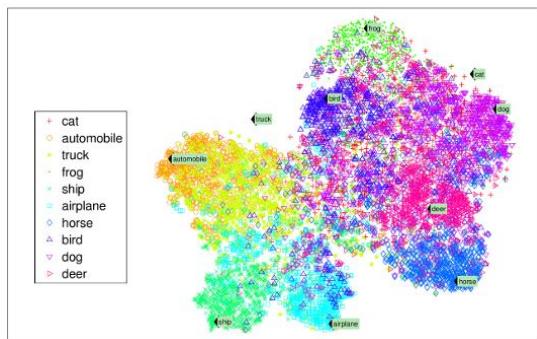
Secondo Chollet (2024), “l'apprendimento automatico si basa sull'idea di trasformare i dati in rappresentazioni utili; l'apprendimento non supervisionato si concentra proprio su questa trasformazione, senza però una direzione imposta da etichette o obiettivi esplicativi”[7].

I principali compiti dell'apprendimento non supervisionato includono:

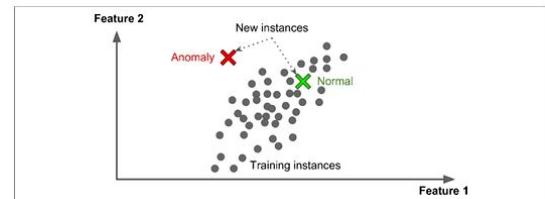
- **Clustering**: consiste nel raggruppare automaticamente i dati in insiemi (cluster) tali che gli elementi appartenenti allo stesso gruppo siano tra loro simili, mentre quelli appartenenti a gruppi

diversi siano dissimili. Gli algoritmi più noti e comuni sono: K-Means e DBSCAN. Il clustering è utilizzato ad esempio per segmentare utenti, immagini o documenti (vd. figura 2.3c).

- **Riduzione della dimensionalità:** consente di semplificare dataset ad alta dimensionalità trasformandoli in spazi di dimensioni inferiori, mantenendo quanto più possibile l'informazione originaria. Tecniche come la Principal Component Analysis (PCA), t-SNE e UMAP sono essenziali per la visualizzazione e l'analisi esplorativa (vd. figura 2.4a).
- **Rilevamento di anomalie (*anomaly detection*):** identifica dati che si discostano significativamente dal comportamento medio del dataset. Viene impiegato, ad esempio, nella rilevazione di frodi, nel monitoraggio di sistemi industriali o nella sicurezza informatica (vd. figura 2.4b).
- **Apprendimento generativo:** mira a modellare la distribuzione probabilistica dei dati per generare nuovi esempi plausibili. Tecniche come gli autoencoder, le Variational Autoencoder (VAE) e le Generative Adversarial Networks (GAN) rientrano in questo ambito.



(a) Esempio di visualizzazione con t-SNE che evidenzia cluster semantici [2]



(b) Rilevamento di anomalie [2]

Figure 2.4: Visualizzazione e rilevamento di pattern: t-SNE e identificazione di anomalie.

### 2.2.3 Importanza e applicazioni pratiche

L'apprendimento non supervisionato è particolarmente importante in contesti in cui:

- Le etichette sono assenti, difficili da ottenere o costose da produrre.
- È necessario esplorare grandi quantità di dati non strutturati (ad es. testo, immagini, log).
- Si vogliono costruire pipeline di pre-processing o pre-training per modelli supervisionati.

La riduzione della dimensionalità consente di visualizzare dati complessi in 2 o 3 dimensioni per esplorare pattern nascosti. I modelli generativi permettono di sintetizzare dati artificiali a partire da insiemi reali, con applicazioni che spaziano dalla creazione di immagini al completamento di testo.

Come osservato da Géron, “*l'apprendimento non supervisionato è spesso il punto di partenza di progetti complessi, e può essere estremamente utile anche quando non si dispone ancora di una chiara formulazione del problema*”[2].

## 2.3 Tecniche di clustering

Il clustering è una tecnica di apprendimento non supervisionato che mira a raggruppare dati simili tra loro in insiemi denominati *cluster*. Non richiede etichette per i dati e viene ampiamente utilizzato in scenari dove non si conosce a priori la struttura del dataset. Tra le principali applicazioni del clustering si trovano:

- segmentazione di utenti o clienti in base a comportamenti;
- rilevamento di anomalie o outlier;
- compressione dei dati e riduzione dimensionale;
- elaborazione di immagini, audio e testo;

- pre-elaborazione per apprendimento supervisionato.

Esistono molte tecniche di clustering, ma le più studiate e applicate sono K-Means e DBSCAN. È importante sottolineare che non esiste una definizione universalmente accettata di *cluster*. La sua interpretazione, infatti, dipende fortemente dal contesto applicativo e dalla metrica utilizzata [2, cap. 9].

### 2.3.1 K-Means

#### Descrizione generale

L'algoritmo K-Means è una delle tecniche di clustering più diffuse e consolidate. Il suo obiettivo è partizionare un insieme di dati in  $k$  cluster distinti minimizzando la distanza intra-cluster. Ogni cluster è rappresentato dal suo centroide, calcolato come la media dei punti che gli sono assegnati. L'algoritmo assume implicitamente che i cluster abbiano una forma sferica e una densità simile.

Il procedimento standard si sviluppa nei seguenti passaggi (vd. figura 2.5):

1. **Inizializzazione:** scelta di  $k$  centroidi iniziali, spesso con inizializzazione casuale, ma si preferisce il metodo K-Means++ che migliora la scelta iniziale [2].
2. **Assegnazione:** ogni punto viene assegnato al centroide più vicino secondo la distanza euclidea.
3. **Aggiornamento:** i centroidi vengono ricalcolati come media dei punti assegnati.
4. **Ripetizione:** si ripetono i passi 2 e 3 fino alla convergenza, cioè fino a quando le assegnazioni non cambiano più.

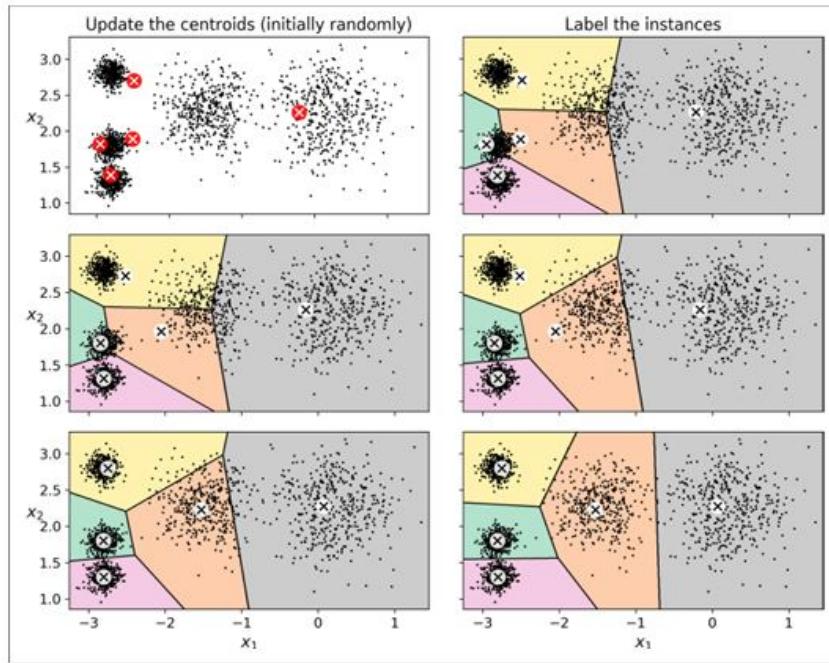


Figure 2.5: Algoritmo K-Means illustrato: ad ogni iterazione vengono aggiornate le assegnazioni ai cluster e ricalcolati i centroidi.

Il criterio di ottimizzazione è la *somma delle distanze quadratiche* tra i punti e i propri centroidi:

$$J = \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \quad (2.1)$$

dove  $\mu_i$  è il centroide del cluster  $C_i$ .

### Variazioni e ottimizzazioni

- **K-Means++**: inizializza i centroidi in modo più intelligente, selezionando punti iniziali distanziati tra loro. Questo approccio riduce significativamente la probabilità che l'algoritmo converga a un minimo locale sfavorevole [5].
- **Elkan's K-Means**: proposto da Charles Elkan nel 2003, questo algoritmo accelera l'esecuzione del K-Means classico evitando numerosi calcoli inutili della distanza. Sfrutta la disuguaglianza triangolare per mantenere limiti superiori e inferiori delle distanze tra punti e centroidi, riducendo i confronti diretti [4].
- **MiniBatchKMeans**: introdotto da David Sculley nel 2010, questa variante utilizza piccoli sottoinsiemi casuali (mini-batch) del dataset ad ogni iterazione per aggiornare gradualmente i centroidi. Questo consente di gestire grandi quantità di dati in memoria limitata e di ottenere una velocità di convergenza fino a 3-4 volte superiore rispetto all'algoritmo standard, con una riduzione marginale della qualità del clustering [6].

### Vantaggi

- Algoritmo semplice, intuitivo e computazionalmente efficiente: ha complessità temporale circa  $O(knm)$  dove  $n$  è il numero di punti,  $k$  il numero di cluster e  $m$  la dimensionalità dei dati.
- Si adatta bene a cluster compatti, ben separati e di forma sferica.
- Scalabile a dataset molto grandi (con l'uso di MiniBatchKMeans).

### Svantaggi

- Richiede di specificare il numero di cluster  $k$  a priori, il che spesso non è noto.
- Sensibile all'inizializzazione: scelte iniziali errate possono portare a soluzioni subottimali.
- Presume che tutti i cluster abbiano forma simile e densità comparabile.
- Non robusto agli outlier, che possono influenzare significativamente la posizione dei centroidi.

### Applicazioni tipiche

K-Means è stato impiegato con successo in numerosi ambiti:

- segmentazione clienti su base comportamentale;
- compressione di immagini, ad esempio riducendo la palette dei colori;
- clustering di documenti rappresentati tramite vettori TF-IDF;
- riduzione della dimensionalità come tecnica di preprocessing.

## 2.3.2 DBSCAN

### Descrizione generale

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [2, pp. 255–258] è un algoritmo che definisce i cluster come aree ad alta densità di punti separati da regioni a bassa densità. A differenza di K-Means, DBSCAN non presuppone alcuna forma per i cluster e non richiede di specificare il numero di cluster.

L'algoritmo considera i seguenti tipi di punti (vd. figura 2.6):

- **Core points**: hanno almeno  $minPts$  punti nel raggio  $\varepsilon$  (incluso se stesso).
- **Border points**: hanno meno di  $minPts$  vicini ma sono nel raggio di un core point.
- **Noise points**: non sono core né border point; vengono etichettati come rumore.

Il funzionamento dell'algoritmo è il seguente :

1. Si seleziona un punto arbitrario non ancora visitato.

2. Se è un core point, si forma un nuovo cluster che viene espanso iterativamente.
3. Si prosegue finchè tutti i punti sono stati visitati.

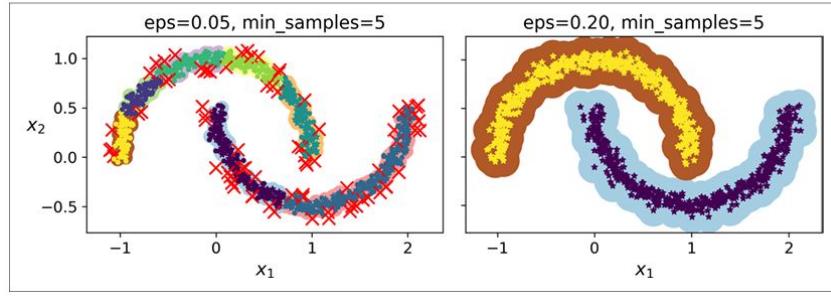


Figure 2.6: Clustering DBSCAN usando due diversi `neighborhood radiuses`.

### Parametri

- `epsilon ( $\varepsilon$ )`: raggio massimo per considerare due punti come “vicini”.
- `min_samples (minPts)`: numero minimo di punti per considerare un punto come `core`.

### Vantaggi

- Non richiede la specifica del numero di cluster.
- Può rilevare cluster di forma arbitraria (non necessariamente sferica).
- Rileva esplicitamente outlier come `noise`.
- Robusto in presenza di rumore e utile per analisi esplorative.

### Svantaggi

- La scelta di  $\varepsilon$  e `minPts` è critica e non è sempre semplice.
- Non adatto a cluster con densità molto variabili.
- Non dispone nativamente di un metodo `predict()` per l’assegnazione di nuovi dati; si consiglia l’uso combinato con classificatori (es. KNN) per risolvere il problema [2, p. 257].

### Applicazioni tipiche

DBSCAN viene applicato efficacemente in:

- clustering geografico e spaziale (es. hotspot criminali);
- rilevamento anomalie (es. frodi, guasti industriali);
- elaborazione di immagini mediche;
- scoperta di pattern locali in set di dati rumorosi.

### 2.3.3 Confronto tra K-Means e DBSCAN

Il confronto tra K-Means e DBSCAN mette in luce due approcci molto diversi al clustering. K-Means è un algoritmo semplice ed efficiente, particolarmente adatto a cluster di forma regolare e con densità simile, ma mostra limiti significativi in presenza di rumore o forme non sferiche. Al contrario, DBSCAN è un algoritmo basato sulla densità, più robusto al rumore e capace di identificare cluster di forma arbitraria, senza richiedere a priori il numero di gruppi. Tuttavia, richiede un’attenta scelta dei parametri  $\varepsilon$  e `minPts`, e non supporta nativamente la predizione su nuovi dati. La tabella (vd. tabella 2.1) seguente sintetizza le principali differenze tra i due metodi.

| Caratteristica           | K-Means                          | DBSCAN                                    |
|--------------------------|----------------------------------|---|
| Numero cluster richiesto | Sì                               | No  |
| Forma dei cluster        | Sferica                          | Arbitraria                                |
| Gestione outlier         | Pessima                          | Buona (outlier espliciti)                 |
| Robustezza al rumore     | Bassa                            | Alta                                      |
| Predizione su nuovi dati | Sì (con <code>predict()</code> ) | No (ma estendibile)                       |
| Complessità              | $O(kmn)$                         | $O(n \log n)$ (con struttura indicizzata) |
| Parametri                | $k$                              | $\varepsilon, minPts$                     |
| Sensibilità ai parametri | Alta                             | Alta (ma diversa)                         |

Table 2.1: Confronto tra K-Means e DBSCAN [2, pp. 248, 258]

## 2.4 Tecniche di riduzione dimensionale

La riduzione dimensionale è una fase fondamentale nel pre-processing dei dati, particolarmente utile in contesti ad alta dimensionalità come l'analisi di immagini, segnali o embedding generati da modelli complessi. Le tecniche di riduzione dimensionale possono essere suddivise in due categorie principali: **metodi di proiezione lineare** e **metodi di apprendimento molteplice (manifold learning)**. Tra i metodi più rappresentativi troviamo PCA, t-SNE e UMAP, ciascuno con finalità e proprietà differenti.

### 2.4.1 PCA (Principal Component Analysis)

La PCA, ovvero *Principal Component Analysis*, è una tecnica statistica di riduzione dimensionale appartenente alla classe dei metodi di proiezione lineare. Essa mira a trasformare un insieme di dati ad alta dimensionalità in un nuovo sistema di coordinate, in cui le nuove variabili, dette *componenti principali*, rappresentano le direzioni di massima varianza nei dati. In altre parole, la PCA identifica quegli assi lungo i quali i dati variano maggiormente, riducendo così la dimensionalità senza perdere informazione essenziale.

**Procedura generale.** La PCA può essere implementata seguendo questi passaggi fondamentali:

- **Centratura dei dati:** si sottrae la media di ciascuna variabile in modo che i dati abbiano media nulla.
- **Calcolo della matrice di covarianza:** questa matrice cattura la varianza e le correlazioni tra le variabili.
- **Decomposizione della matrice di covarianza:** si estraggono autovalori e autovettori.
- **Ordinamento degli autovettori:** si ordinano in base agli autovalori decrescenti, che indicano quanta varianza è spiegata da ciascun componente.
- **Proiezione:** si proiettano i dati sui primi  $k$  autovettori (componenti principali), ottenendo una rappresentazione in uno spazio di dimensione inferiore.

**Applicazioni.** La trasformazione ottenuta consente una rappresentazione compatta ed efficace dei dati. La PCA è comunemente utilizzata per:

- **Visualizzazione:** proiettando i dati in 2D o 3D per l'esplorazione visiva.
- **Compressione:** riducendo il numero di features mantenendo la massima informazione possibile.
- **Riduzione del rumore:** filtrando le componenti meno significative.
- **Pre-processing:** migliorando l'efficacia di altri algoritmi di machine learning.

**Varianti della PCA.** Esistono diverse estensioni e varianti della PCA, ciascuna pensata per specifiche esigenze computazionali o caratteristiche dei dati:

- **Incremental PCA (IPCA):** è una versione adattiva della PCA che consente di processare i dati in batch successivi, utile quando il dataset è troppo grande per essere caricato interamente in memoria.

**Scikit-Learn** implementa IPCA secondo l'approccio descritto da Ross et al.<sup>1</sup> [2, Capitolo 8, sezione **Incremental PCA**, pagg. 225].

- **Randomized PCA:** è una tecnica che utilizza proiezioni casuali per approssimare rapidamente le componenti principali. Questo approccio si basa sul lavoro di Halko et al.<sup>2</sup> ed è particolarmente efficiente per dataset molto grandi, grazie al fatto che richiede pochi passaggi sui dati e sfrutta al meglio l'hardware parallelo [2, Capitolo 8, sezione **Randomized PCA**, pagg. 225].
- **Kernel PCA:** estende la PCA classica a contesti non lineari, grazie all'utilizzo del **kernel trick**. Consente di mappare i dati in uno spazio di dimensione superiore dove le relazioni lineari sono più facilmente identificabili. Questo metodo è stato introdotto da Schölkopf et al.<sup>3</sup> [2, Capitolo 8, sezione **Kernel PCA**, pagg. 226–230].

### 2.4.2 t-SNE (t-Distributed Stochastic Neighbor Embedding)

Il **t-SNE**, introdotto da **van der Maaten e Hinton** nel 2008<sup>4</sup>, è un algoritmo di **manifold learning** sviluppato specificamente per la visualizzazione di dati ad alta dimensionalità. A differenza di tecniche come PCA, che conservano la struttura globale, **t-SNE** si concentra nel preservare la struttura locale, cioè le relazioni di vicinanza tra punti simili nello spazio originale [2, pp. 232–233].

**Principi di funzionamento.** L'algoritmo **t-SNE** costruisce due distribuzioni di probabilità [2]:

- La prima nello spazio originale, che riflette la probabilità che un punto  $x_j$  sia un vicino di  $x_i$ , calcolata con una gaussiana centrata in  $x_i$ .
- La seconda nello spazio ridotto (tipicamente 2D o 3D), che utilizza invece una distribuzione **t di Student** a una sola coda, più pesante rispetto alla gaussiana.

Successivamente, **t-SNE** cerca di minimizzare la *divergenza di Kullback-Leibler* (KL) tra le due distribuzioni. L'ottimizzazione viene effettuata tramite discesa del gradiente stocastico [2].

I passaggi principali sono:

- Calcolo delle distanze Euclidee tra i punti nel dataset originale.
- Costruzione delle probabilità condizionate e simmetriche nello spazio originale.
- Costruzione della distribuzione nello spazio ridotto usando la distribuzione **t di Student**.
- Ottimizzazione della divergenza KL tra le due distribuzioni per posizionare i punti nello spazio a bassa dimensionalità.

### Proprietà.

- Eccellente per visualizzare strutture locali e individuare **cluster nascosti** nei dati.
- Non conserva le distanze globali o le proporzioni metriche [2].
- Non fornisce una trasformazione *invertibile* o generalizzabile su nuovi dati.
- È sensibile a iperparametri critici, come:
  - **perplexity** (regola il bilanciamento tra struttura locale e globale),
  - **learning rate**,
  - numero di iterazioni.

<sup>1</sup>David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. *Incremental Learning for Robust Visual Tracking*. International Journal of Computer Vision, 77(1):125–141, May 2008.

<sup>2</sup>N. Halko, P. G. Martinsson, and J. A. Tropp. *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*. SIAM Review, 53(2):217–288, January 2011.

<sup>3</sup>Bernhard Schölkopf, Alexander Smola, Klaus-Robert Müller. *Kernel Principal Component Analysis*, Lecture Notes in Computer Science, vol. 1327. Springer, 1997.

<sup>4</sup>Laurens van der Maaten and Geoffrey Hinton. *Visualizing Data using t-SNE*. Journal of Machine Learning Research, 9(Nov):2579–2605, 2008.

**Applicazioni.** Nonostante i suoi limiti, t-SNE è ampiamente utilizzato per:

- Visualizzazione di **dataset di immagini** (es. MNIST, CIFAR-10) [2].
- Analisi esplorativa di **word embeddings** o **latent vectors** da reti neurali.
- Identificazione di anomalie e pattern nascosti in dataset complessi.

### 2.4.3 UMAP (Uniform Manifold Approximation and Projection)

UMAP è una tecnica di riduzione dimensionale non lineare sviluppata per rappresentare in uno spazio a bassa dimensionalità la struttura intrinseca di dati ad alta dimensionalità. È basata su solide fondamenta teoriche che combinano la topologia algebrica, la teoria delle varietà riemanniane e la teoria della misura. Introdotta da McInnes et al. [11], UMAP si distingue per la sua capacità di preservare sia la struttura locale sia, in parte, quella globale dei dati.

**Fondamenti matematici.** UMAP parte dal presupposto che i dati giacciono su una varietà riemanniana immersa in uno spazio euclideo ad alta dimensione. Per approssimare tale varietà, UMAP costruisce un grafo di prossimità ponderato tra i punti, basato su una misura di distanza locale controllata da un parametro di vicinanza (tipicamente `n_neighbors`) [11]. Questo grafo è interpretato come una fuzzy simplicial set, una struttura formale della topologia algebrica [14].

Successivamente, UMAP costruisce una versione semplificata dei dati riducendone le dimensioni, cercando di mantenere intatte le relazioni tra i punti. Per farlo, confronta la struttura originale con quella nello spazio ridotto usando una funzione di costo basata sulla **cross-entropy** che misura quanto le connessioni tra i punti nei due spazi siano simili [11]. In questo modo, UMAP cerca di preservare al meglio le vicinanze tra i dati.

**Vantaggi rispetto a t-SNE.** Rispetto ad altri metodi di riduzione dimensionale come t-SNE, UMAP presenta numerosi vantaggi:

- **Efficienza computazionale:** UMAP è generalmente più veloce e più scalabile su grandi dataset grazie alla sua implementazione ottimizzata e alla complessità algoritmica più favorevole [11].
- **Preservazione globale:** UMAP tende a preservare anche la struttura globale dei dati, mentre t-SNE si concentra quasi esclusivamente sulle relazioni locali [3].
- **Robustezza ai parametri:** UMAP richiede meno tuning parametrico rispetto a t-SNE e fornisce risultati più stabili al variare dei parametri [11, 14].
- **Versatilità:** Può essere utilizzato come passo di preprocessing prima di clustering, come metodo di visualizzazione e persino come metodo di apprendimento semi-supervisionato o supervisionato [3, 14].

**Applicazioni comuni.** Grazie alla sua versatilità e potenza, UMAP è utilizzato in una vasta gamma di applicazioni, tra cui:

- **Visualizzazione di embedding:** particolarmente utile per l'esplorazione di rappresentazioni generate da modelli di deep learning (es. autoencoder, word embeddings) [3].
- **Preprocessing per clustering:** riducendo la dimensionalità in modo significativo, UMAP può migliorare le prestazioni di algoritmi come DBSCAN, HDBSCAN e K-Means [15, 11].
- **Analisi esplorativa dei dati:** utile per l'esplorazione interattiva di dati ad alta dimensione, come in bioinformatica, scienze sociali o mobilità urbana [3].
- **Visualizzazione temporale o evolutiva:** in ambito biologico e di monitoraggio, permette di osservare traiettorie o transizioni nello spazio latente [3].

## 2.5 Reti neurali e Autoencoder

### 2.5.1 Introduzione alle reti neurali

Le reti neurali artificiali (*Artificial Neural Networks*, *ANN*) sono modelli computazionali ispirati alla struttura e al funzionamento del cervello umano. Esse rappresentano uno degli strumenti più potenti e flessibili nell'ambito dell'apprendimento automatico, in particolare nel contesto del **deep learning**. L'idea alla base delle reti neurali è quella di costruire un sistema composto da molteplici unità elementari, chiamate neuroni artificiali o **percetroni**, in grado di apprendere rappresentazioni complesse a partire dai dati grezzi.

Ogni neurone esegue una trasformazione non lineare di una combinazione pesata dei suoi input e propaga il risultato al livello successivo della rete. Una rete neurale profonda (*Deep Neural Network*, *DNN*) è caratterizzata dalla presenza di più strati nascosti (**hidden layers**) tra input e output, in cui ciascun layer apprende una trasformazione dei dati sempre più astratta [7].

Il processo di apprendimento avviene tipicamente mediante l'algoritmo della discesa del gradiente, che consente di minimizzare una funzione di **loss** (**loss function**), adattando iterativamente i pesi della rete in modo da migliorare la qualità della previsione sui dati di addestramento [2].

### 2.5.2 Il deep learning

Il termine **deep learning** si riferisce specificamente all'uso di reti neurali con numerosi strati nascosti per apprendere rappresentazioni gerarchiche dei dati. Come sottolineato da Chollet, “il deep learning è l'apprendimento automatico che utilizza reti neurali profonde come modello” [7]. Questo approccio si è rivelato particolarmente efficace in compiti complessi come la classificazione di immagini, il riconoscimento vocale, la traduzione automatica e la generazione di testo o immagini.

Secondo Géron, il successo del **deep learning** è dovuto alla combinazione di tre fattori principali [2]:

- la disponibilità di grandi quantità di dati
- la potenza computazionale fornita dalle GPU
- miglioramenti significativi negli algoritmi di addestramento e nelle architetture delle reti neurali

### 2.5.3 Autoencoder

Gli **autoencoder** sono una tipologia particolare di rete neurale utilizzata per apprendere una rappresentazione (codifica) compatta ed efficiente dei dati di input. L'architettura tipica di un autoencoder consiste in due componenti principali: un **encoder**, che comprime i dati in uno spazio latente di dimensione ridotta, e un **decoder**, che ricostruisce i dati originali a partire dalla rappresentazione latente (vd. figura 2.7).

L'obiettivo dell'**autoencoder** è minimizzare l'errore di ricostruzione tra input e output, generalmente misurato tramite l'errore quadratico medio (**MSE**, Mean Squared Error). Questo tipo di rete è non supervisionata: non richiede etichette, ma apprende direttamente dalle strutture statistiche dei dati [2].

Chollet sottolinea che gli **autoencoder** non si limitano a comprimere i dati, ma sono in grado di apprendere trasformazioni significative che permettono di scoprire strutture latenti complesse nei dati, rendendoli strumenti ideali per la riduzione della dimensionalità e il pretraining di reti più profonde [7].

### 2.5.4 Varianti degli autoencoder

Nel corso degli anni sono state sviluppate numerose varianti dell'architettura base degli autoencoder, ciascuna finalizzata a un obiettivo specifico:

- **Denoising Autoencoder**: introdotti per rendere l'**autoencoder** robusto al rumore. L'input viene corrotto con rumore casuale e l'**autoencoder** è addestrato a ricostruire l'input originale pulito [2] (vd. figura 2.8a).
- **Sparse Autoencoder**: si impone una regolarizzazione di tipo L1 sull'attivazione dei neuroni per favorire rappresentazioni sparse, ovvero attivazioni in cui solo pochi neuroni sono attivi contemporaneamente [2] (vd. figura 2.8b).

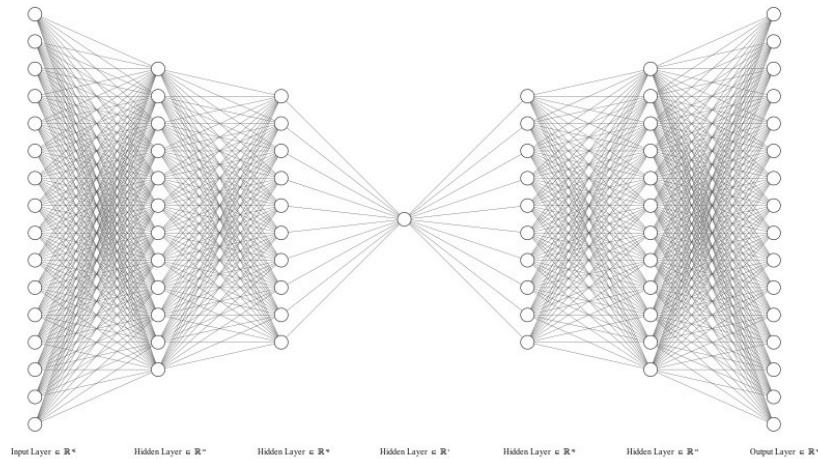


Figure 2.7: Architettura generale di un autoencoder: l'encoder comprime l'input in uno spazio latente, il decoder lo ricostruisce.

- **Variational Autoencoder (VAE):** combinano reti neurali e modelli probabilistici. Invece di apprendere una singola codifica per ogni input, apprendono una distribuzione probabilistica nello spazio latente. Questo li rende adatti per la generazione di nuovi dati realistici [2] (vd. figura 2.8c).
- **Convolutional Autoencoder:** utilizzano strati convoluzionali, rendendoli particolarmente adatti per dati strutturati spazialmente, come immagini [2] (vd. figura 2.8d).

### 2.5.5 Addestramento e regolarizzazione

L’addestramento degli **autoencoder** richiede strategie per evitare il fenomeno dell’**overfitting**, ossia quando il modello si adatta troppo ai dati di addestramento perdendo capacità generalizzativa. Tra le tecniche più comuni si annoverano:

- la regolarizzazione L1 e L2 sui pesi;
- l’uso di **Dropout**, ovvero la disattivazione casuale di neuroni durante l’addestramento;
- l’**early stopping**, ovvero l’interruzione dell’addestramento quando la prestazione sui dati di validazione inizia a peggiorare [7, 2].

## 2.6 Rilevamento di anomalie

### 2.6.1 Introduzione

Il rilevamento di anomalie (**anomaly detection** o **outlier detection**) è il processo mediante il quale si identificano osservazioni nei dati che si discostano significativamente dal comportamento atteso o comune. Tali osservazioni anomale possono indicare errori nei dati, eventi rari o potenziali minacce ad esempio, frodi finanziarie, guasti in sistemi industriali o accessi non autorizzati a sistemi informatici [2].

Secondo Géron, il rilevamento delle anomalie è particolarmente importante nei contesti in cui i dati anomali sono troppo rari per essere ben rappresentati nei set di addestramento supervisionati. In questi casi, è preferibile adottare approcci non supervisionati o semi-supervisionati [2].

### 2.6.2 Approcci principali

I metodi per il rilevamento di anomalie possono essere distinti in diverse categorie:

- Modelli basati su soglie: stabiliscono dei limiti statici o dinamici per considerare una misura come anomala. Sono semplici da implementare, ma poco flessibili.

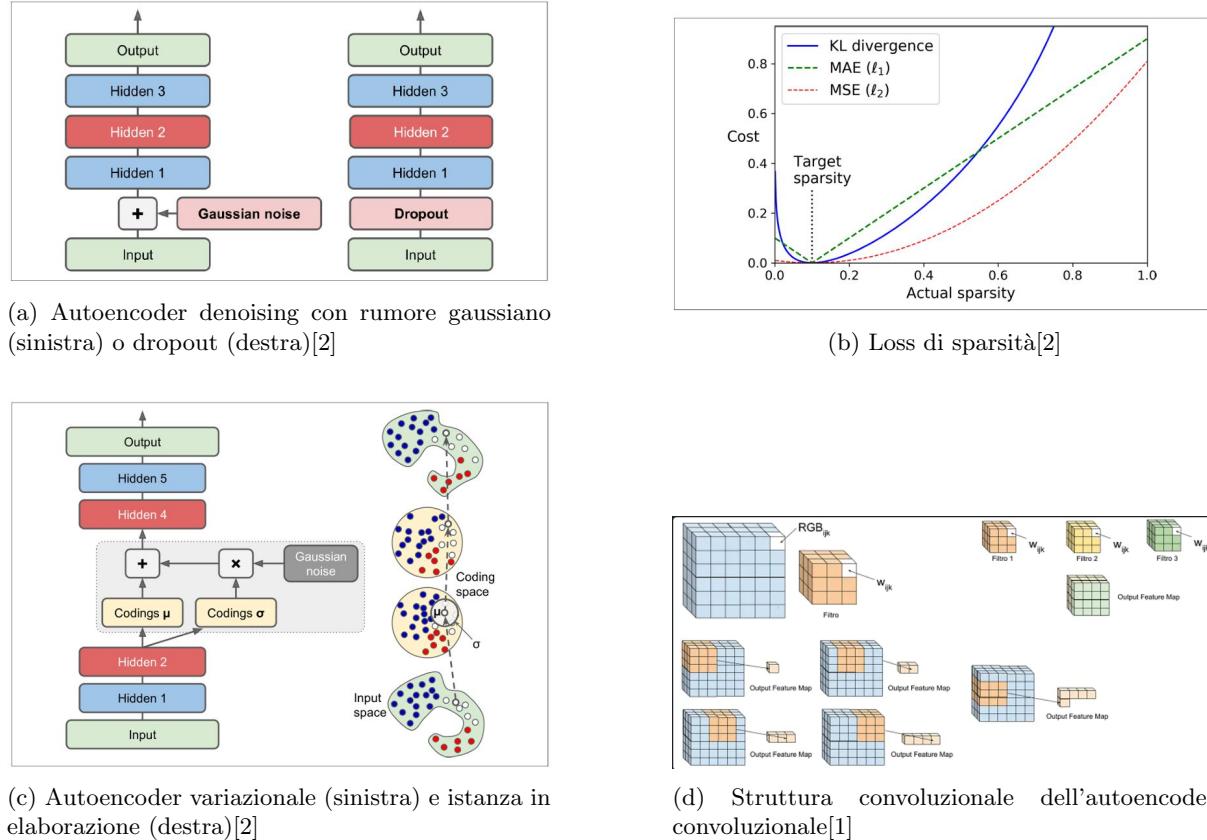


Figure 2.8: Principali varianti di autoencoder: denoising, sparse, variazionale e convoluzionale.

- Metodi statistici: assumono che i dati seguano una certa distribuzione (tipicamente gaussiana) e considerano anomali i punti con bassa probabilità a posteriori.
- Metodi basati su distanza o densità: identificano come anomali i punti distanti da altri o situati in regioni a bassa densità. Tra questi si trovano algoritmi come DBSCAN, kNN-based outlier detection, e Local Outlier Factor (LOF) [2].
- Modelli di apprendimento automatico: includono tecniche non supervisionate (come gli autoencoder), supervisionate (quando sono disponibili etichette di anomalie), e semi-supervisionate (quando si conoscono solo dati “normali”).

### 2.6.3 Autoencoder per il rilevamento di anomalie

Gli **autoencoder** si sono dimostrati particolarmente efficaci nel contesto del rilevamento di anomalie non supervisionato. L’idea è semplice ma potente: addestrare un **autoencoder** su dati “normali”, in modo tale che esso apprenda una rappresentazione compatta delle caratteristiche più ricorrenti nel dataset. Quando il modello è presentato con dati anomali, la ricostruzione risulterà più imprecisa, poiché tali dati non sono stati visti durante l’addestramento. Di conseguenza, un errore di ricostruzione elevato può essere utilizzato come segnale per identificare un’anomalia [7].

Secondo Chollet, “gli **autoencoder** possono essere efficacemente per il rilevamento di anomalie, in quanto imparano a ricostruire i dati normali con elevata precisione, mentre i dati anomali, non conformi alle distribuzioni apprese, risultano difficili da ricostruire” [7].

In pratica, una soglia sull’errore di ricostruzione (tipicamente la media quadratica degli errori) viene utilizzata per distinguere osservazioni normali da quelle anomale. La scelta della soglia è un aspetto critico e può essere definita empiricamente (ad esempio, mediante percentili) o tramite metodi basati su validazione.

### 2.6.4 Varianti di autoencoder per l'anomaly detection

Per migliorare la performance nel rilevamento delle anomalie, possono essere utilizzate diverse varianti degli autoencoder:

- **Denoising Autoencoder (DAE)**: allenati a ricostruire l'input originale da una versione corrotta, tendono a focalizzarsi sulle strutture più robuste del dato, rendendoli più efficaci nel generalizzare al di fuori del training set [2].
- **Sparse Autoencoder**: introducono una regolarizzazione sull'attivazione dei neuroni per costringere la rete a utilizzare solo una parte ridotta della capacità rappresentazionale. Questo induce il modello ad apprendere features più discriminative, utili per rilevare deviazioni dai pattern appresi [2].
- **Variational Autoencoder (VAE)**: attraverso una formulazione probabilistica dello spazio latente, sono in grado di misurare la “probabilità” che una nuova osservazione appartenga alla distribuzione dei dati normali. Punti con bassa probabilità sono considerati anomalie [2].

### 2.6.5 Misure di performance

Poiché le anomalie sono spesso eventi rari e sbilanciati rispetto ai dati normali, metriche come l'accuratezza non sono informative. È quindi preferibile l'uso di misure specifiche come la **precision**, la **recall**, la **F1-score** e l'**AUC-ROC** per valutare la qualità del rilevamento. Inoltre, la curva Precision-Recall è particolarmente utile in contesti sbilanciati [2].

## 2.7 Metriche di valutazione: Silhouette Score, Davies-Bouldin, Calinski-Harabasz

Nel contesto dell'apprendimento non supervisionato, e in particolare delle tecniche di clustering, è fondamentale disporre di metriche in grado di valutare la qualità dei raggruppamenti ottenuti. Poiché in questi casi non si dispone di etichette “vere” di riferimento, si ricorre a misure di coerenza interna e separazione tra i cluster. Le tre metriche più comuni sono il **Silhouette Score**, l'indice di **Davies-Bouldin** e l'indice di **Calinski-Harabasz** [2].

### 2.7.1 Silhouette Score

Il **Silhouette Score** misura quanto ogni punto dati è vicino al proprio cluster rispetto a quelli vicini. Formalmente, per ciascun punto  $i$  si definisce:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

dove:

- $a(i)$  è la distanza media di  $i$  dagli altri punti del proprio cluster;
- $b(i)$  è la distanza media dal punto  $i$  a tutti i punti del cluster più vicino diverso dal proprio.

Il valore di  $s(i)$  varia tra  $-1$  e  $1$ : valori vicini a  $1$  indicano che il punto è ben assegnato al proprio cluster; valori prossimi a  $0$  indicano sovrapposizione tra cluster; valori negativi indicano un'assegnazione probabilmente errata.

Come spiega Géron, il Silhouette Score può essere utilizzato per confrontare il risultato di un algoritmo su diverse configurazioni (es. diverso numero di cluster  $k$ ), e una sua media elevata è indicativa di una buona separazione tra cluster [2].

### 2.7.2 Indice di Davies-Bouldin (DBI)

Il **Davies-Bouldin Index** valuta la qualità del clustering sulla base di una combinazione tra la compattezza intra-cluster e la separazione inter-cluster. Per ciascun cluster  $i$  e per ogni altro cluster  $j \neq i$ , si calcola:

$$R_{ij} = \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$$

dove:

- $\sigma_i$  è la deviazione media (o varianza) dei punti del cluster  $i$  rispetto al suo centroide  $c_i$ ;
- $d(c_i, c_j)$  è la distanza tra i centroidi dei cluster  $i$  e  $j$ .

Il valore dell'indice è dato dalla media, sui cluster, del massimo  $R_{ij}$  per ciascun  $i$ . Valori più bassi indicano una migliore separazione tra i cluster. Come evidenziato in [2], questo indice è particolarmente utile quando si desidera una valutazione più rigorosa della separabilità geometrica dei cluster.

### 2.7.3 Indice di Calinski-Harabasz (CHI)

L'Indice di Calinski-Harabasz, noto anche come Variance Ratio Criterion, è calcolato come il rapporto tra la dispersione tra i cluster e quella all'interno dei cluster:

$$\text{CH} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \cdot \frac{n - k}{k - 1}$$

dove:

- $\text{Tr}(B_k)$  è la traccia della matrice di dispersione tra i cluster (inter-cluster variance);
- $\text{Tr}(W_k)$  è la traccia della matrice di dispersione intra-cluster;
- $n$  è il numero totale di campioni e  $k$  il numero di cluster.

Un valore elevato dell'indice indica che i cluster sono ben separati e compatti. Géron consiglia l'uso del CHI come metrica comparativa tra configurazioni alternative, in particolare quando il numero di cluster non è noto a priori [2].

# Chapter 3

## Dataset e preparazione dei dati

### 3.1 Origine dei dati (Fluctuo, POI, dati temporali e geografici)

#### 3.1.1 Dati sui Punti di Interesse (POI)

I dati relativi ai Punti di Interesse (**Points of Interest**, **POI**, figura 3.1<sup>1</sup>) utilizzati in questa analisi provengono dalla **Overture Maps Foundation**, un'iniziativa collaborativa open-source fondata da Meta, Microsoft, Amazon Web Services e TomTom, con lo scopo di sviluppare e mantenere una mappa mondiale libera, aperta e interoperabile<sup>2</sup>.

In particolare, si è fatto riferimento al dataset “Places” rilasciato il 19 febbraio 2025, disponibile pubblicamente tramite Amazon S3<sup>3</sup>.

Il dataset è fornito in formato “GeoParquet”, un formato *columnar* ad alte prestazioni progettato per la gestione di dati geospatiali su larga scala<sup>4</sup>. Ogni record rappresenta un’entità **POI**, comprendente:

- **Identificatore univoco** (**id**);
- **Geometria spaziale**, rappresentata in formato **WKT** o **GeoJSON** secondo lo schema di Overture<sup>5</sup>;
- **Categorie semantiche** (**categories**), organizzate secondo uno schema standardizzato, come ad esempio `overture:amenity:restaurant`;
- **Nomi multilingua** (**names**), con supporto a varianti locali;
- **Indirizzi strutturati** (**addresses**), con campi per via, città, codice postale, ecc.;
- **Valore di confidenza** (**confidence**), una metrica quantitativa della qualità e affidabilità del dato.

La struttura del dataset segue lo schema “Places”<sup>6</sup> definito all’interno della documentazione ufficiale di Overture.

Per questa analisi, i dati sono stati filtrati geograficamente per includere esclusivamente i **POI** localizzati nel territorio della città di Milano (Italia). Tale selezione è stata eseguita applicando una clausola SQL sul campo **addresses**, verificando la presenza della stringa “Milano” come valore di **locality**. I dati filtrati sono stati successivamente esportati nei formati **CSV** e **GeoJSON** per analisi geospatiali e visualizzazione.

<sup>1</sup><https://www.rivistageomedia.it/dati-geografici/overture-maps-foundation-e-la-cartografia-delle-infrastrutture-per-i-trasporti>

<sup>2</sup><https://docs.overturemaps.org>

<sup>3</sup>[s3://overturemaps-us-west-2/release/2025-02-19.0/theme=places/\\*/\\*](https://s3://overturemaps-us-west-2/release/2025-02-19.0/theme=places/*/*)

<sup>4</sup><https://parquet.apache.org/>

5<https://docs.overturemaps.org/schema/concepts/by-theme/places/>6<https://docs.overturemaps.org/schema/concepts/by-theme/places/>

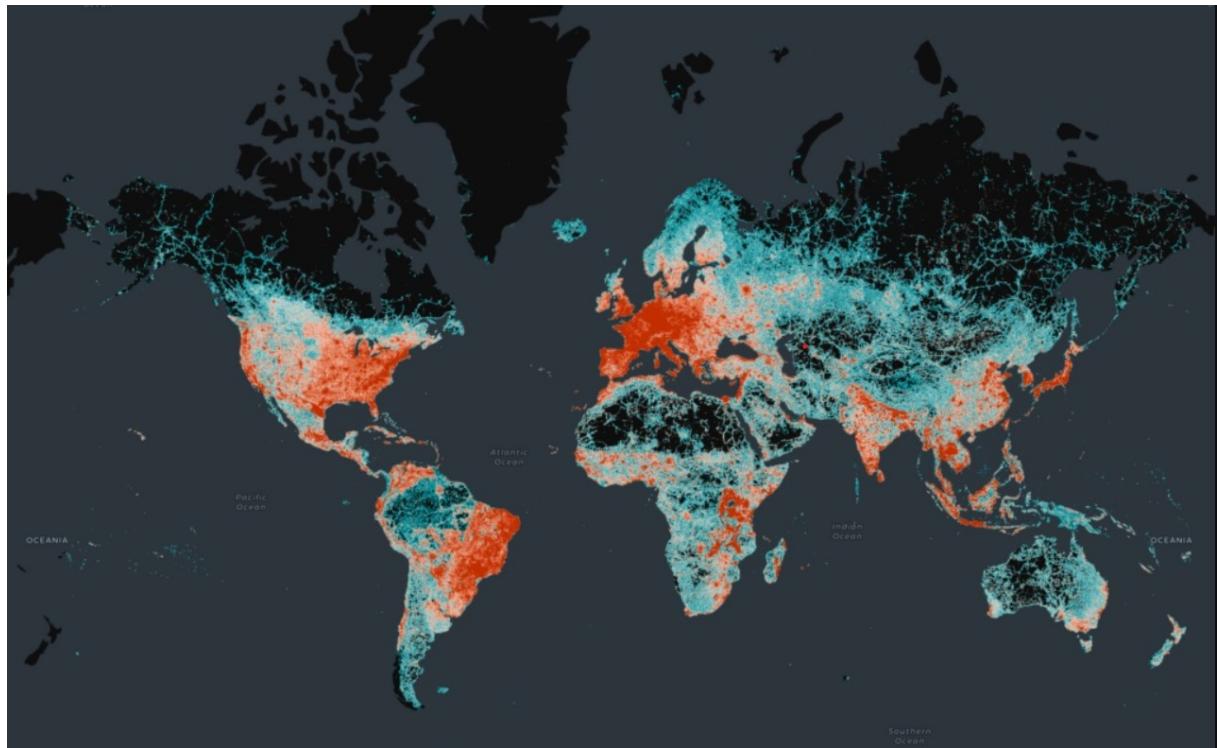


Figure 3.1: Cartografia delle infrastrutture per i trasporti generata a partire dai dati della *Overture Maps Foundation*.

### 3.1.2 Dati Fluctuo

I dati utilizzati in questo studio provengono da Fluctuo (vd. figura 3.2<sup>7</sup>), un aggregatore di dati specializzato nei servizi di mobilità condivisa acquisiti nel contesto delle attività dello Spoke 8 — MaaS and Innovative services del Centro Nazionale per la Mobilità SosTenibile(MOST), costituito grazie al “Piano nazionale di ripresa e resilienza (PNRR)—M4C2, investimento 1.4, “Potenziamento strutture di ricerca e creazione di “campioni nazionali di R&S” su alcune Key Enabling Technologies”.

Il dataset copre il periodo da settembre 2022 ad agosto 2024 e include circa 6.8 milioni di spostamenti. I dati sono organizzati in 15 file CSV mensili, ciascuno contenente dettagli a livello di singolo spostamento. Tuttavia, al fine di condurre un’analisi più approfondita, sono stati considerati solo i dati relativi all’anno 2023.

### 3.1.3 Dati di trasporto pubblico urbano

Per arricchire i flussi di mobilità urbana si è fatto uso inoltre delle informazioni relative ai dati di posizione delle fermate dei mezzi di trasporto pubblico della città di Milano utilizzando tre dataset:

- Linee di superficie (autobus, tram, ecc...)
- Linee metropolitane (Metro)
- Linee ferroviarie (Treni)

Ciò ha permesso di coprire diverse modalità di spostamento al fine di effettuare uno studio più completo. Tutti i dati dei mezzi di trasporto appartengono al comune di Milano con licenza Creative Commons Attribution 4.0 International (CC BY 4.0). <sup>8</sup>

<sup>7</sup><https://osservatoriosharingmobility.it/in-europa-ogni-secondo-si-fanno-5-viaggi-di-mobilita-condivisa/>

<sup>8</sup><https://creativecommons.org/licenses/by/4.0/>

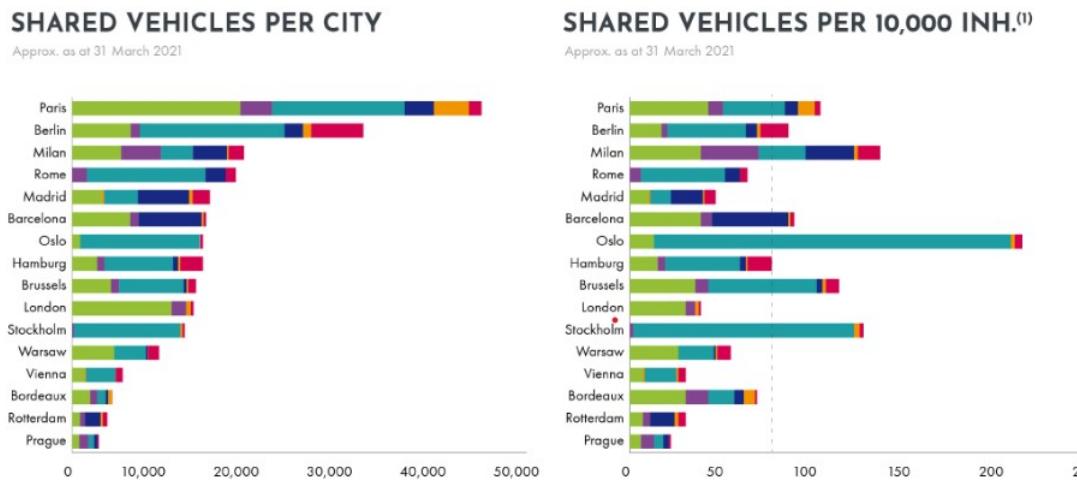


Figure 3.2: Azienda Fluctuo

© All rights reserved - © Fluctuo 2021

### Linee di superficie

Il dataset intitolato **ATM - Fermate linee di superficie urbane** (**Comune di Milano, 2024a**) (vd. figura 3.3)<sup>9</sup> contiene l'elenco delle fermate servite dalle linee di trasporto pubblico di superficie (tram, bus, filobus) gestite da ATM. Ogni record nel dataset è associato a:

- Un identificativo numerico (**id\_amat**)
- Il luogo della fermata (**ubicazione**)
- L'elenco delle linee di superficie che vi transitano (**linee**)
- La geometria del punto (**LONG\_X\_4326, LAT\_Y\_4326, Location**)

Questo dataset è stato aggiornato il 16 ottobre 2024 ed è disponibile nei formati **.csv, .geojson e .shp**.

### Linee metropolitane

Il secondo dataset impiegato è **ATM - Fermate linee metropolitane** (**Comune di Milano, 2024b**) (vd. figura 3.4)<sup>10</sup>, che fornisce la localizzazione delle stazioni della metropolitana milanese (linee M1-M5). Il dataset è formato nel seguente modo:

- Un identificativo numerico (**id**)
- Il nome della fermata (**nome**)
- L'elenco delle linee metropolitane che vi transitano (**linee**)
- La geometria del punto (**LONG\_X\_4326, LAT\_Y\_4326, Location**)

Questo dataset è stato aggiornato il 16 ottobre 2024 ed è disponibile nei formati **.csv, .geojson e .shp**.

### Linee ferroviarie

Il terzo dataset impiegato è Trasporto pubblico (vd. figura 3.5): **localizzazione delle stazioni ferroviarie** (**Comune di Milano, 2019**)<sup>11</sup>, che fornisce la localizzazione delle stazioni delle ferroviarie nel territorio milanese. Il dataset è costruito come segue:

- La città della stazione che nel nostro caso si tratta sempre di Milano (**Stazione**)
- Se la stazione è in superficie o sotterranea (**Ubicazione**)
- Quali sono le linee che passano per quella stazione (**Linee**)

<sup>9</sup><https://dati.comune.milano.it/en/dataset/ds534-atm-fermate-linee-di-superficie-urbane>

<sup>10</sup>[https://dati.comune.milano.it/en/dataset/ds535\\_atm-fermate-linee-metropolitane](https://dati.comune.milano.it/en/dataset/ds535_atm-fermate-linee-metropolitane)

<sup>11</sup>[https://dati.comune.milano.it/en/dataset/ds80\\_infogeo\\_stazioni\\_ferroviarie\\_localizzazione](https://dati.comune.milano.it/en/dataset/ds80_infogeo_stazioni_ferroviarie_localizzazione)

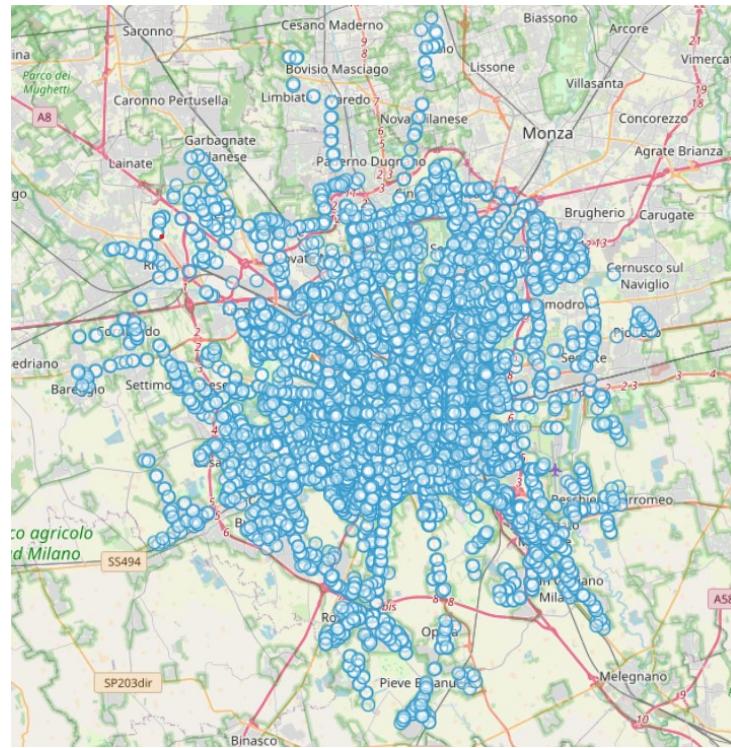


Figure 3.3: Mappa delle linee di Superficie

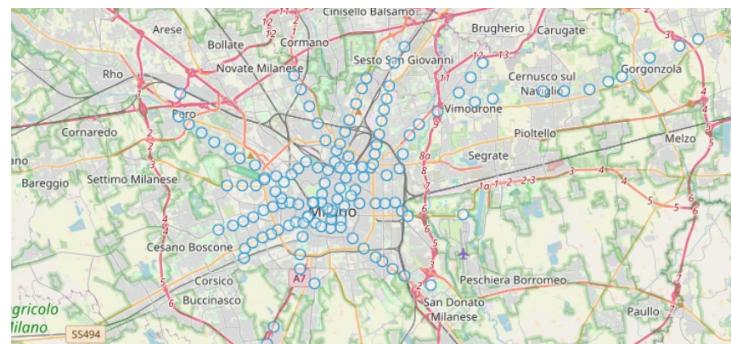


Figure 3.4: Mappa delle linee metropolitane

- Delle note indicative sui dati (**Note**)
- A che municipio appartiene (**Municipio**)
- Identificativo dei nuclei di identità locale (**id.nil**)
- A che nucleo di identità locale appartiene la stazione (**nil**)
- La geometria del punto (**LONG\_X\_4326**, **LAT\_Y\_4326**, **Location**)

Il dataset è stato aggiornato il 17 settembre 2024 ed è disponibile nei formati *.csv*, *.geojson* e *.json*.

## 3.2 Costruzione del tensore multidimensionale

Per analizzare efficacemente i dati di mobilità urbana, precedentemente descritti, abbiamo usufruito dell'utilizzo di tecniche di apprendimento non supervisionato. E' stato necessario trasformare tutti i dati contenuti nei dataset iniziali (*Fluctuo-viaggi*, *OvertureMaps-POI* e *trasporto pubblico urbano*) in una struttura dati adatta: un tensore multifunzionale.

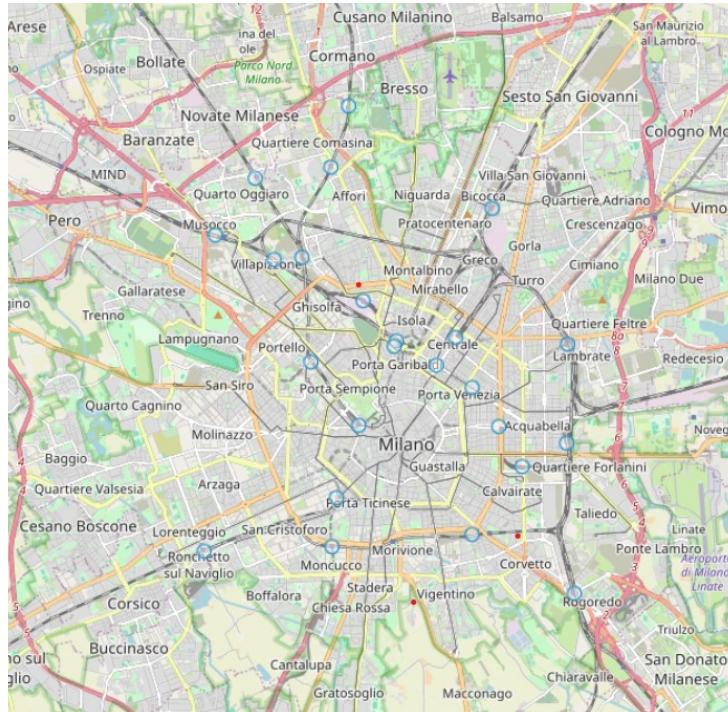


Figure 3.5: Mappa delle linee ferroviarie

### 3.2.1 Rappresentazione dei Viaggi

Ogni viaggio è stato rappresentato come un vettore di caratteristiche derivate da informazioni temporali, categoriali, spaziali e contestuali, con l’obiettivo di cogliere le correlazioni tra tutti questi dati.

Il tensore finale ottenuto dopo il processo di elaborazione può essere visto come una matrice di dimensione  $n \times m$  (con  $m = 62$ ), dove:

- $n$ : numero di righe, ciascuna rappresentante un singolo viaggio;
- $m$ : numero di colonne, ognuna delle quali rappresenta una caratteristica (feature) del viaggio.

Il tensore costruito riflette le diverse caratteristiche di un viaggio, che possiamo raggruppare come segue:

- **Caratteristiche temporali**
- **Caratteristiche modali**
- **Caratteristiche spaziali**
- **Caratteristiche contestuali**

#### Caratteristiche temporali

Il primo gruppo di caratteristiche si occupa di quando ha luogo un viaggio perciò sono state incluse nel tensore numerose variabili temporali, come l’ora e il minuto di inizio e fine, il mese e il giorno, nonché indicatori numerici e binari che rappresentano il giorno della settimana o se il viaggio ha avuto luogo nel weekend o in un giorno festivo.

Tutti questi dati sono stati ottenuti a partire dalle informazioni contenute nel dataset di Fluctuo, effettuando un processo di separazione dei dati di inizio e fine viaggio, originariamente presenti in un’unica colonna, in più colonne distinte. Questo ha permesso una suddivisione più strutturata e un controllo più efficiente sui singoli attributi del viaggio.

Queste informazioni permettono di distinguere, ad esempio, viaggi sistematici (come quelli pendolari) da quelli occasionali o legati al tempo libero. Inoltre, sono stati introdotti indicatori binari per i giorni festivi istituzionali e le vacanze scolastiche, in quanto tali eventi influenzano sensibilmente i pattern di spostamento e la quantità di viaggi in un determinato periodo.

### **Caratteristiche modali**

Il secondo gruppo di caratteristiche riguarda il mezzo di trasporto utilizzato: automobile, bicicletta, scooter elettrico o motorino.

Per ogni viaggio è stata creata una codifica binaria che indica la modalità principale, fondamentale per distinguere ad esempio da viaggi più rapidi e diretti a viaggi più lenti o soggetti a vincoli infrastrutturali

### **Caratteristiche spaziali**

Il terzo gruppo di caratteristiche riguarda dove vengono fatti questi viaggi, perciò vengono salvate le coordinate geografiche (latitudine e longitudine) del punto di partenza e di arrivo.

In aggiunta viene inclusa anche la distanza media stimata del viaggio, utile per capire micro-spostamenti o comportamenti anomali

### **Caratteristiche contestuali**

L'ultimo gruppo di caratteristiche comprende quelle contestuali, cioè quelle caratteristiche che arricchiscono l'informazione del viaggio con elementi dell'ambiente urbano circostante.

Per ogni punto di partenza ed arrivo vengono calcolati:

- Il numero di infrastrutture per il trasporto pubblico (ferrovie, metro, linee di superficie);
- Numero totale di punti di interesse (POI);
- Presenza o meno di POI appartenenti alle sei categorie funzionali principali:
  - turismo e intrattenimento,
  - ristorazione,
  - sport e cura della persona,
  - servizi,
  - attività commerciali,
  - istituzioni educative.

Queste caratteristiche consentono di inquadrare il viaggio nel suo contesto sociale e infrastrutturale.

## **3.3 Pulizia e trasformazione delle variabili**

La prima fase del lavoro si è concentrata sulla pulizia e trasformazione delle variabili contenute nel dataset in quanto la coerenza dei dati è un prerequisito fondamentale nello studio e nell'analisi dei dati

### **3.3.1 Gestione di valori mancanti o anomalie**

In primo luogo è stata effettuata un'analisi per controllare la presenza o meno di valori nulli o mancanti, *null* o *Nan* (*Not a Number*) e vengono fatti controlli rimuovendo tutti gli elementi avari:

- Durate negative o uguali a 0
- Distanze stimate inferiori a 10 metri o superiori a 100 chilometri
- Discrepanze incoerenti tra i valori di inizio e di fine del viaggio

### **3.3.2 Conversione e codifica delle variabili**

Dopo aver avuto la conferma che i dati siano integri, si è proceduto a trasformare le variabili in formati compatibili con gli input degli algoritmi di apprendimento automatico i quali richiedono tensori numerici

Questa conversione viene fatta sia per dati di inizio del viaggio sia per i dati di fine del viaggio

### Variabili categoriali

Le variabili (come il mezzo di trasporto o il giorno della settimana) sono state trasformate in una rappresentazione in codifica binaria in cui ogni categoria diventa una colonna che può assumere valori 0 o 1 andando così a creare le codifiche per:

- I giorni della settimana (start\_monday, end\_friday, ecc...)
- I tipi di mezzo utilizzati (car, bike, ecc...)
- Le categorie funzionali dei POI (eat\_and\_drinks\_start, ecc...)

### Variabili temporali e geografiche

Le variabili temporali continue (ora, minuto, giorno e mese) sono mantenute in formato numerico intero le quali verranno normalizzate in seguito (vd. Sezione 3.5)

Per quanto riguarda le coordinate geografiche (che verranno anch'esse normalizzate in seguito) si è preferito mantenerle in virgola mobile sia come input per il clustering sia per eventuali proiezioni geografiche per visualizzare i punti (vd. dalla sezione 5.12)

### Variabili numeriche continue

Le variabili continue come:

- estimated\_duration\_in\_mn
- actual\_duration\_mn
- estimated\_distance\_in\_meter
- numero di POI e infrastrutture

sono state mantenute in formato numerico intero o a virgola mobile e sottoposte a standardizzazione nella fase di normalizzazione (vedi Capitolo 4.5)

## 3.4 Integrazione dei dataset e operazioni preliminari

### 3.4.1 Operazioni preliminari

Dopo aver acquisito i dati da fonti eterogenee (viaggi, POI, trasporto pubblico) si è resa necessaria un'attenta fase di integrazione, volta a uniformare i formati, pulire i dati e arricchire il dataset principale

Innanzitutto come operazione iniziale i dataset vengono suddivisi con il carattere ‘;’ per facilitare la lettura e l'integrazione di tutti i dati

Le operazioni preliminari riguardano principalmente i dati del dataset Fluctuo e del dataset riguardante i **POI**.

### 3.4.2 Dati Fluctuo

Nel dataset Fluctuo tutte le features sono convertite da stringhe a formati corretti quali **datetime** o coordinate geospaziali.

#### Dati temporali

I dati temporali sono stati convertiti controllando i due formati **%Y-%m-%d %H:%M:%S.%f** e **%Y-%m-%d %H:%M:%S**, successivamente estraendo gli attributi temporali di mese, giorno, ora e minuto.

A partire dalla data di inizio e fine dei viaggi vengono ricavate diverse informazioni quali:

- la durata effettiva del viaggio che può essere messa a paragone con quella stimata;
- la presenza del viaggio in un giorno feriale o festivo;
- l'indicazione del viaggio se avviene durante il fine settimana;

- l'indicazione di presenza o meno di festività italiane tramite la libreria `holidays`<sup>12</sup>.

### Dati spaziali

Lo stesso lavoro di conversione viene effettuato sui dati spaziali convertendoli in valori numerici, ottenendo le coordinate e salvandole singolarmente come valori numerici a virgola mobile.

### Dati modali

Infine vengono anche convertiti i dati delle tipologie di veicolo creando una variabile booleana (`true` e `false`) per rappresentare o meno l'utilizzo di quel determinato veicolo.

### 3.4.3 Dati POI

La preparazione dei dati e l'ambiente di analisi sono stati configurati tramite Google Colab, sfruttando la compatibilità con Python, DuckDB e Google Drive per il salvataggio persistente dei dati.

La prima parte del notebook prevede l'installazione delle librerie necessarie:

- `duckdb` per interrogazioni SQL in locale e su file Parquet<sup>13</sup>
- `geopandas` per analisi spaziali;<sup>14</sup>
- `pandas` per la manipolazione dei dataframe;<sup>15</sup>
- `ipython-sql` per l'uso delle SQL magic in Jupyter/Colab;<sup>16</sup>
- `pyarrow` per il supporto Parquet.<sup>17</sup>

Successivamente è stato configurato l'ambiente DuckDB, caricando le estensioni:

- `spatial` per la gestione di dati geospaziali;
- `httpfs` per l'accesso diretto a file remoti (es. Amazon S3).

Infine, è stata stabilita una connessione al database DuckDB e definita la regione S3 (`us-west-2`) da cui leggere i file.

Il caricamento dei dati è avvenuto direttamente da S3<sup>18</sup>, senza necessità di download manuale.

Viene applicato un filtro sulla località Milano verificando che almeno un elemento nell'array `addresses` contenesse “country” = ‘IT’ e “locality” ILIKE ‘Milano’.

I risultati sono stati salvati in due formati:

- CSV, per l'analisi tabellare (campi: `id`, `geometry`, `names`, `categories`, `confidence`, `addresses`);
- GeoJSON, per la visualizzazione su mappa.

Una volta ottenuto il dataset filtrato, sono state analizzate le seguenti colonne:

- **categories**: un array JSON contenente etichette che descrivono la natura del POI. È stato analizzato per identificare la distribuzione delle categorie presenti a Milano. Le etichette sono state poi raggruppate in un insieme composto da 7 macro-categorie per favorirne la leggibilità e la comprensione<sup>19</sup>.
- **confidence**: un campo numerico che esprime la qualità del dato. Valori più alti indicano una maggiore affidabilità. Sono stati valutati i range di distribuzione per identificare eventuali dati rumorosi o non affidabili.

---

<sup>12</sup><https://github.com/vacanza/holidays>

<sup>13</sup><https://duckdb.org/>

<sup>14</sup><https://geopandas.org/en/stable/>

<sup>15</sup><https://pandas.pydata.org/>

<sup>16</sup><https://github.com/catherinedevlin/ipython-sql>

<sup>17</sup><https://arrow.apache.org/>

<sup>18</sup>[s3://overturemaps-us-west-2/release/2025-02-19.0/theme=places/\\*/\\*](https://overturemaps-us-west-2/release/2025-02-19.0/theme=places/*/*)

<sup>19</sup><https://docs.overturemaps.org/schema/concepts/by-theme/places/>

### 3.4.4 Integrazione nel dataset

Una volta finite tutte le operazioni preliminari viene unito il tutto per la creazione del tensore utilizzando la funzione `calcola_ball_tree`.

La funzione `calcola_ball_tree` utilizza la combinazione di BallTree<sup>20</sup> e distanza di Haversine<sup>21</sup>, ed è stata progettata per arricchire un dataset di spostamenti urbani mediante l'integrazione di informazioni contestuali relative al territorio.

Questa funzione associa a ciascun punto geolocalizzato una serie di informazioni riguardanti POI, mezzi di trasporto in un raggio di 200 metri di vicinanza dal punto stesso.

#### Struttura e funzionamento

L'approccio adottato si basa sull'utilizzo di strutture dati di tipo BallTree implementate con la metrica Haversine che permette di calcolare le distanze su una superficie sferica.

La funzione accetta in input:

- i dataset dei viaggi, dei POI e dei mezzi pubblici quali metro, linee di superficie e linee ferroviarie;
- le colonne contenenti latitudine e longitudine;
- un parametro `raggio_m` che indica il raggio di vicinanza in metri da considerare per ciascuna entità (in questo caso 200 metri).

Dopo aver convertito le coordinate geografiche dei punti in radianti, la funzione costruisce i rispettivi BallTree per ciascun dataset ausiliario. Ogni punto del dataset dei viaggi Fluctuo viene quindi confrontato, mediante ricerca spaziale, con le strutture create per verificare quali entità si trovano entro il raggio specificato.

Le features arricchite generate dalla funzione sono le seguenti:

- `ferrovia_colonna`: variabile binaria che indica se almeno una stazione ferroviaria è presente entro il raggio definito;
- `superficie_colonna`: numero di fermate del trasporto pubblico di superficie presenti nel raggio;
- `metro_colonna`: somma delle linee metropolitane distinte associate alle stazioni entro il raggio;
- `POI_colonna`: numero totale di punti di interesse rilevati nel raggio.

Inoltre, vengono generate sei variabili binarie, una per ciascuna macro-categoria di POI:

- `tourism_and_entertainment_col`
- `eat_and_drinks_col`
- `sport_and_care_col`
- `services_col`
- `businesses_col`
- `institutions_and_educations_col`

Tutte queste features vengono poi inserite nel tensore finale.

Questo processo viene effettuato sia per i punti di inizio del viaggio che per i punti di fine al fine di arricchire le informazioni riguardo allo scopo del viaggio o alle modalità di spostamento.

## 3.5 Normalizzazione

Dopo aver terminato la creazione del tensore possiamo passare finalmente all'esecuzione dell'autoencoder.

Prima di tutto procediamo alla preparazione dei dati per l'addestramento, cioè alla normalizzazione.

---

<sup>20</sup><https://scikit-learn/stable/modules/generated/sklearn.neighbors.BallTree.html>

<sup>21</sup><https://github.com/mapado/haversine/blob/main/haversine/haversine.py>

Per normalizzazione si intende la trasformazione delle features numeriche in un intervallo standard, per migliorare la convergenza del modello.

A questo scopo è stata implementata la funzione `load_and_normalize_data`:

Questa funzione tra le varie operazioni che esegue (vd. Sezione 4.1.2) applica la normalizzazione Min-Max alle features.

Trasforma le caratteristiche scalando ciascuna caratteristica in un intervallo specificato.

Questo stimatore scala e trasla ogni caratteristica individualmente in modo che rientri nell'intervallo specificato sul set di addestramento, ad esempio tra 0 e 1.

La trasformazione è definita come:

$$X_{\text{std}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad \Rightarrow \quad X_{\text{scaled}} = X_{\text{std}} \times (\max - \min) + \min$$

dove  $\min$ ,  $\max$  rappresentano i limiti dell'intervallo desiderato (`features_range`).

Questa trasformazione viene spesso utilizzata come alternativa alla normalizzazione a media zero e varianza unitaria.

Il `MinMaxScaler` non riduce l'effetto degli outlier, ma li scala linearmente entro un intervallo fisso: il punto dati più grande corrisponderà al valore massimo dell'intervallo e quello più piccolo al minimo.<sup>22</sup>

---

<sup>22</sup><https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

# Chapter 4

## Architettura e metodologia

### 4.1 Pipeline sperimentale

Dopo aver creato il tensore unendo le informazioni di tutti i dati ricavati dai dataset descritti nel capitolo precedente, ora si può iniziare la fase di utilizzo delle tecniche di apprendimento automatico. Durante il flusso di lavoro (**pipeline sperimentale**) verrà costruito, addestrato e testato un modello autoencoder al fine di individuare pattern ricorrenti o delle eventuali anomalie all'interno degli spostamenti della città di Milano con mezzi di mobilità in sharing.

L'intera pipeline può essere suddivisa nelle seguenti fasi principali (vd, figura 4.1):

#### 4.1.1 Caricamento, preparazione del dataset e normalizzazione

Il primo step consiste nel caricamento dei dati contenuti nel tensore precedentemente costruito, pre-elaborato e strutturato in modo da contenere informazioni di interesse.

- Il caricamento è stato effettuato con la libreria **Polars**, scelta per la sua efficienza nella gestione di grandi dataset.
- Il dataset include una colonna identificativa **id**, che viene separata dal resto dei dati prima della normalizzazione.

#### 4.1.2 Normalizzazione e conversione in tensori

Per garantire un apprendimento efficace da parte della rete neurale, i dati numerici sono stati normalizzati nel range [0, 1] tramite **MinMaxScaler** di **Scikit-learn** (vd. Sezione 3.5).

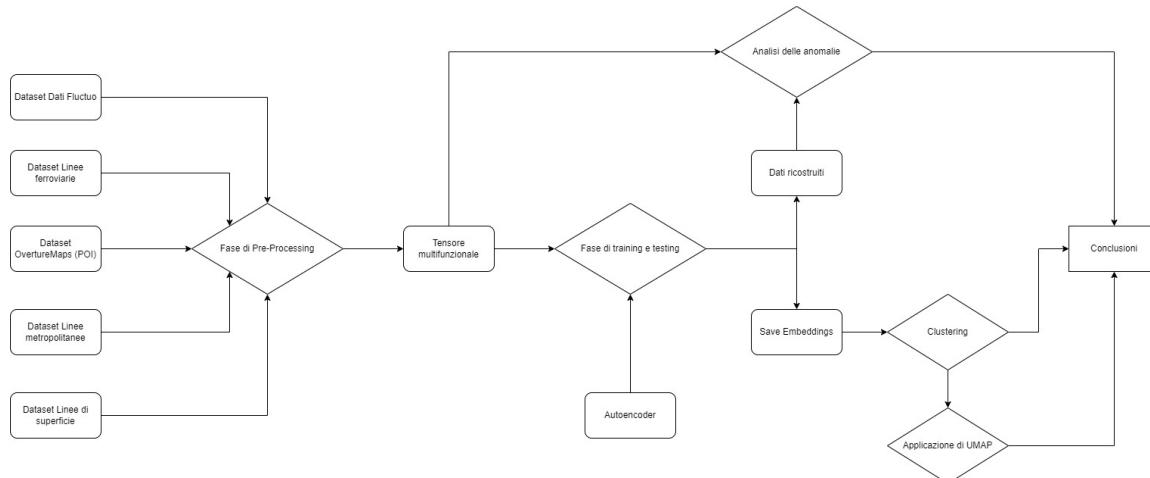


Figure 4.1: Autoencoder creato e utilizzato.

- Dopo la normalizzazione, i dati vengono convertiti in tensori PyTorch, per poter essere forniti direttamente all'autoencoder.
- Questo processo permette di mantenere la coerenza delle feature e facilita l'**training** della rete neurale.

#### 4.1.3 Progettazione dell'Autoencoder

La rete neurale autoencoder è stata progettata per apprendere una rappresentazione compressa dei dati. Essa svolge due compiti fondamentali:

- L'**encoder** trasforma i dati originali in un vettore di dimensione ridotta, chiamato *embeddings*.
- Il **decoder** tenta di ricostruire i dati originali a partire dagli *embeddings*.

L'autoencoder viene addestrato minimizzando l'errore di ricostruzione, cioè la differenza tra input e output.

#### 4.1.4 Training e Test del modello

Il **training** del modello è stato effettuato utilizzando PyTorch. Il dataset è stato diviso in due parti:

- **Training set** (80%): usato per aggiornare i pesi della rete.
- **Test set** (20%): usato per monitorare il modello ed evitare overfitting.

Durante il **training** sono stati monitorati i seguenti elementi:

- La **funzione di loss** monitorata sia con che senza la regolarizzazione L1/L2.
- L'**errore di ricostruzione medio**, utile per la fase di rilevamento delle anomalie.

Viene anche controllato l'**errore quadratrico assoluto (ASE)** per assicurarsi che il modello non vada in overfitting

#### 4.1.5 Rilevamento delle anomalie

Una volta addestrato il modello esso viene salvato e su di esso viene effettuato un rilevamento di possibili anomalie durante la ricostruzione dei dati. La metodologia si basa sull'analisi dell'errore di ricostruzione:

- Si calcola, per ogni istanza, la distanza tra i dati originali e quelli ricostruiti dall'autoencoder.
- Le osservazioni con un errore superiore a una certa soglia statistica (es. media +  $2\sigma$ ) vengono identificate come *anomalie*.

Viene effettuata un'analisi su questi osservando:

- Qual è la feature più difficile da ricostruire
- Qual è la percentuale di viaggi considerati anomalie

Questa sezione serve a valutare l'efficacia del modello nella ricostruzione delle features.

#### 4.1.6 Estrazione e salvataggio degli embedding

Una volta finita l'analisi sul modello, vengono estratti gli embedding (vettori ridotti) generati dal livello di bottleneck. Questi embedding rappresentano una proiezione compressa di tutti i viaggi, per poi essere salvati in un nuovo file .csv per essere riutilizzati nelle fasi successive.

#### 4.1.7 Clustering sugli embedding

Sui vettori di embeddings ottenuti è stato applicato un algoritmo di clustering (K-Means), con l'obiettivo di raggruppare insieme spostamenti con caratteristiche simili.

- Il numero ottimale di cluster è stato selezionato a seconda di cosa volessimo che venisse rappresentato (vd. Sezione 5.4 e 5.7)

- Ogni spostamento viene quindi associato a un cluster, e questa informazione viene unita al dataset originale per analisi successive.

#### 4.1.8 Riduzione dimensionale per la visualizzazione

Per consentire la *visualizzazione bidimensionale* degli embedding e dei cluster, sono stati applicati metodi di riduzione della dimensionalità:

- PCA (**P**rincipal **C**omponent **A**nalysis): utile per ridurre rumore e linearità nei dati.
- UMAP (**U**niform **M**anifold **A**pproximation and **P**rojection): particolarmente efficace nel preservare la struttura locale e globale dei dati.

Queste tecniche hanno permesso di creare grafici 2D chiari, dove è possibile osservare la separazione tra gruppi di spostamenti.

(vd. tabella 4.1 per un riassunto del workflow generale della pipeline)

| Fase | Descrizione  | Tecnologie                             |
|------|--|--|
| 1    | Caricamento dei dati dal dataset originale, in formato CSV, e conversione in un DataFrame. | Polars                                 |
| 2    | Normalizzazione dei dati e conversione in tensori per il <b>training</b> .                 | MinMaxScaler, PyTorch                  |
| 3    | Progettazione e definizione dell'architettura dell'autoencoder.                            | Reti neurali feedforward               |
| 4    | <b>training</b> del modello con funzioni di loss e <b>test</b> su un set separato.         | Adam optimizer, L1/L2 loss, Early stop |
| 5    | Rilevamento delle anomalie basato sull'errore di ricostruzione dell'autoencoder.           | Reconstruction Error                   |
| 6    | Salvataggio degli embedding generati dall'encoder per l'analisi successiva.                | CSV, PyTorch                           |
| 7    | Applicazione del clustering sugli embedding per rilevare gruppi strutturati nei dati.      | K-Means                                |
| 8    | Riduzione dimensionale per visualizzazione e interpretazione dei cluster.                  | PCA, UMAP                              |

Table 4.1: Sintesi della pipeline sperimentale adottata

## 4.2 Progettazione dell'Autoencoder

L'autoencoder rappresenta il cuore dell'approccio basato sull'apprendimento non supervisionato per la riduzione dimensionale e l'analisi strutturale dei dati di mobilità. Il modello è stato sviluppato utilizzando PyTorch, e ottimizzato attraverso diverse fasi sperimentali, con l'obiettivo di apprendere una rappresentazione compressa (embedding) dei dati in input.

### 4.2.1 Architettura e iperparametri

L'autoencoder per quest'analisi è stato strutturato nel seguente modo:

Due componenti principali strutturati in modo simmetrico, entrambe sono implementate utilizzando il modulo `nn.Sequential` di Pytorch.

#### Encoder

L'encoder ha il compito di comprimere i dati originali ad alta dimensionalità in uno spazio latente (chiamato **bottleneck**). Questo processo è fondamentale per l'estrazione degli **embeddings** che rappresentano l'informazione contenuta nei dati

La struttura è così definita:

1. Un primo livello che riduce la dimensione originale in uno spazio ridotto
2. Un secondo livello che riduce ulteriormente la dimensionalità
3. Un terzo livello che riduce ulteriormente la dimensionalità e genera gli **embeddings**, ovvero la rappresentazione compressa dei dati

Ogni livello è seguito da una funzione di attivazione non lineare, necessaria per introdurre non linearità all'interno della rete

### Decoder

Il decoder ha la funzione opposta dell'encoder, cioè a partire dallo spazio latente ricostruire il dato originale

La struttura del decoder è simmetrica rispetto a quella dell'encoder, partendo dalla dimensione degli **embeddings** fino a tornare alla dimensione originale

### Considerazioni architetturali

La dimensione del **bottleneck** è cruciale: se troppo grande, l'autoencoder rischia di non riuscire a generalizzare le informazioni, se troppo piccola può perdere informazioni importanti

**Iperparametri principali:** L'efficacia dell'autoencoder dipende in larga misura dalla scelta degli iperparametri, i quali determinano le capacità della rete senza incorrere in overfitting o underfitting:

- **n\_epochs** = 100: Numero massimo di epoche di **training**, se il modello non migliora viene attuato un meccanismo per fermare il processo di **training** (early stopping)
- **batch\_size** = 64: Dimensione dei batch utilizzati per il **training**
- **learning\_rate** =  $1 \times 10^{-3}$ : Tasso di apprendimento dell'ottimizzatore, questo valore controlla quanto velocemente i pesi della rete vengono aggiornati.
- **hidden\_1** = 48, **hidden\_2** = 24: Dimensioni dei due layer dell'encoder e del decoder.
- **bottleneck\_size** = 12: Dimensione dello spazio latente e quindi degli **embeddings**
- **activation\_fn** = `nn.Tanh`: Funzione di attivazione utilizzata alla fine di ogni layer
- **L1** = 0.00001, **L2** =  $1 \times 10^{-5}$ : Coefficienti usati nella funzione di loss:
  - Regolarizzazione L1 (Lasso): aggiunge una penalità sui pesi del modello<sup>1</sup>
  - Regolarizzazione L2 (Ridge): Riduce i pesi più grandi ma non li azzera<sup>2</sup>
- **Optimizer** = `Adam`: Adatta i tassi di apprendimento per ciascun peso della rete<sup>3</sup>.
- **Scheduler** = `ReduceLROnPlateau`: monitora la loss. Se non si osservano miglioramenti per 5 epoche consecutive, il **learning rate** viene ridotto di un fattore prefissato (es. dimezzato), con un limite minimo pari a  $1 \times 10^{-5}$ .

Questi iperparametri sono stati scelti attraverso una serie di esperimenti e di test valutando le performance, l'errore di ricostruzione e loss (vd. figura 4.2).

#### 4.2.2 Creazione dei set di training e test

Dopo la normalizzazione dei dati e la creazione del tensore `torch` creiamo il tensore per il **training** e per la fase di **test**:

- **Training set**: 80% del totale dei dati.
- **test set**: 20% dei dati.

---

<sup>1</sup><https://www.diariodunanalista.it/posts/regolarizzazione-l1-vs-l2-nel-machine-learning-differenze/>

<sup>2</sup><https://www.diariodunanalista.it/posts/regolarizzazione-l1-vs-l2-nel-machine-learning-differenze/>

<sup>3</sup><https://datamasters.it/blog/introduzione-all'algoritmo-di-ottimizzazione-adam/>

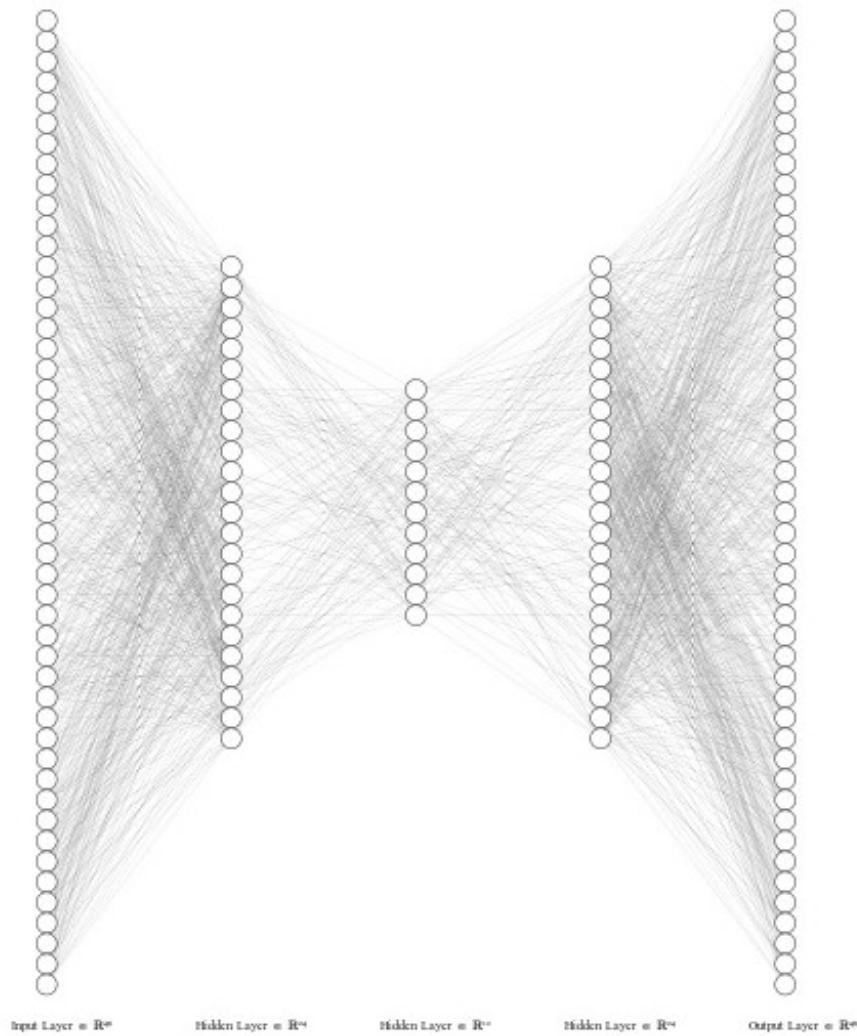


Figure 4.2: Autoencoder creato e utilizzato.

### 4.2.3 Training e Test del modello

Il **training** dell'autoencoder è stato effettuato tramite un ciclo supervisionato che include sia una fase di **training** sia una fase di test, al fine di monitorare la capacità di generalizzazione del modello e prevenire fenomeni di overfitting o underfitting. Il metodo utilizzato, `train_autoencoder_with_test`, implementa una pipeline robusta e modulare che consente di:

- Calcolare la perdita di ricostruzione (*loss*) su base `batch`;
- Applicare regolarizzazione L1 opzionale (in aggiunta alla L2 già presente nell'ottimizzatore);
- Eseguire un test ad ogni epoca per controllare l'errore su un set separato dai dati di **training**;
- Applicare `early stopping` e adattamento del **learning rate** tramite lo scheduler.

**Training** Ad ogni epoca, il modello viene posto in modalità `train()` e addestrato su mini-batch forniti dal `train_loader`. Per ciascun batch:

- Viene calcolata la ricostruzione dell'input.
- Si calcola la `loss` di ricostruzione.
- Viene applicata anche una penalizzazione L1 sui pesi del modello (escludendo i bias);
- Si eseguono i passi di backpropagation e aggiornamento dei pesi tramite l'ottimizzatore Adam;

- Viene calcolato anche l'errore quadratico medio (MSE) per ogni batch, utile per analisi successive. La `loss` media di `training` e il MSE vengono salvati per ogni epoca.

**Test** Al termine di ogni epoca, il modello viene valutato su un `test` set separato:

- Si calcola la loss di `test` su ogni batch del `test_loader`;
- Si calcola il MSE medio;
- Le metriche vengono aggregate e memorizzate in liste per il tracciamento.

**Early stopping** La funzione include un meccanismo di *early stopping* basato sulla mancata riduzione della `test loss`. Se la `loss` non migliora oltre una soglia prefissata ( $10^{-5}$ ) per 10 epoche consecutive, il `training` si interrompe anticipatamente, prevenendo l'overfitting.

**Salvataggio della loss** Le perdite di `training` e `test` per ogni epoca vengono salvate su file CSV, nel formato `epoch,train_loss,val_loss`, per permettere successive visualizzazioni e analisi.

**Valutazione finale e ASE** Al termine del `training`, il modello viene applicato all'intero dataset per generare una versione ricostruita dei dati. Inoltre, viene calcolata una metrica di *Average Squared Error* (ASE) sia sul set di `training` che su quello di `test`, tramite la formula:

$$\text{ASE} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2$$

dove  $x_i$  è il dato originale e  $\hat{x}_i$  la sua ricostruzione.

Se la differenza tra l'ASE di `test` e l'ASE di `training` è superiore al 10% allora il modello ha alte probabilità di overfitting

### 4.3 Rilevamento delle anomalie

Una volta completata la fase di `training` dell'autoencoder, il modello è stato utilizzato per effettuare la ricostruzione degli stessi dati originali normalizzati. Per rilevare le anomalie l'autoencoder, dopo aver appreso la struttura dei dati, dovrebbe essere in grado di ricostruirli con un errore minimo. Al contrario le anomalie tendono ad avere errori di ricostruzione sensibilmente maggiori.

Per ciascuna osservazione è stato quindi calcolato l'errore di ricostruzione come la media dei quadrati delle differenze tra i valori originali e quelli ricostruiti. A partire dalla distribuzione di questi errori è stata definita una soglia statistica per discriminare le anomalie:

$$\text{soglia} = \mu + 3 \cdot \sigma$$

dove  $\mu$  è la media e  $\sigma$  la deviazione standard degli errori di ricostruzione. Tutte le osservazioni il cui errore supera questa soglia sono state etichettate come anomalie.

Il processo di rilevamento è stato implementato in modo tale da:

- Calcolare e salvare gli errori di ricostruzione;
- Etichettare ogni osservazione con una variabile binaria `is_anomaly` che assume valore 1 per le anomalie e 0 altrimenti;
- Salvare i risultati in un file CSV esportabile per analisi successive.

Per una comprensione più approfondita delle cause delle anomalie, è stata condotta un'analisi mirata alle sole istanze anomale. In particolare, per ciascuna variabile originaria è stato calcolato lo scarto medio assoluto tra valore originale e valore ricostruito, al fine di individuare le features maggiormente responsabili delle anomalie.

## 4.4 Salvataggio degli embedding e valutazione del modello

Per valutare le prestazioni dell'autoencoder e ottenere una rappresentazione compressa dei dati, sono stati implementati due metodi distinti: uno per la valutazione del modello e uno per il salvataggio degli embedding.

### 4.4.1 Valutazione del modello

Dopo l'addestramento, il modello è stato valutato confrontando i dati originali con quelli ricostruiti. A tal fine sono state utilizzate tre metriche di errore:

- **Mean Squared Error (MSE)**: misura la media dei quadrati delle differenze tra i valori originali e quelli ricostruiti;
- **Mean Absolute Error (MAE)**: calcola la media del valore assoluto degli errori;
- **R<sup>2</sup> Score**: rappresenta la varianza spiegata, ovvero quanto bene la ricostruzione riesce a spiegare la variabilità dei dati originali.

Il modello viene posto in modalità valutazione per disattivare il calcolo dei gradienti e migliorare l'efficienza computazionale.

### 4.4.2 Salvataggio degli embedding

Per ottenere una rappresentazione degli `embeddings` gli input normalizzati sono passati attraverso l'autoencoder. Il risultato è una matrice a dimensione ridotta, in cui ogni riga rappresenta un'osservazione nello spazio latente

Questi embedding sono stati salvati in un file CSV denominato `embedding.csv`, in cui ciascuna colonna rappresenta una dimensione dello spazio latente (es. `dim_0`, `dim_1`, ...). Tale rappresentazione è stata poi utilizzata per le fasi successive di clustering e riduzione dimensionale.

## 4.5 Clustering sugli embedding

Dopo aver ottenuto gli `embedding` è stata eseguita una fase di clustering per identificare insiemi omogenei di spostamenti all'interno dello spazio latente al fine di studiare i comportamenti principali dei viaggi. A tale scopo, è stato utilizzato l'algoritmo **K-Means**, una tecnica di clustering non supervisionato che ha l'obiettivo di partizionare il dataset in  $k$  gruppi, minimizzando la distanza intra-cluster.

### 4.5.1 Metodo di clustering

Il clustering è stato applicato direttamente sugli `embedding` prodotti dal modello addestrato, che rappresentano una versione compressa ma informativa degli spostamenti. La funzione `cluster_only` esegue le seguenti operazioni:

1. Inizializza l'algoritmo **K-Means** con un numero di cluster specificato (`n_clusters`).
2. Applica il clustering agli `embedding`, ottenendo un'etichetta intera per ciascun punto.
3. Calcola le principali metriche di valutazione della bontà del clustering:
  - **Silhouette Score**: Misura la coesione e separazione dei cluster.
  - **Davies-Bouldin Score**: Valuta la compattezza e separazione dei cluster (valori più bassi indicano clustering migliore).
  - **Calinski-Harabasz Index**: Misura la dispersione *intra* e *inter* cluster.
4. Salva le etichette di clustering in un file CSV, una colonna per ogni configurazione di  $k$ .

## 4.6 Tecniche di riduzione dimensionale applicate

Infine nell'ultima fase del lavoro viene applicata una riduzione dimensionale per due finalità principali:

- Visualizzazione esplorativa dei cluster generati tramite K-Means, su uno spazio di 2 o 3 dimensioni;
- Valutazione qualitativa della separabilità dei cluster nello spazio latente compresso generato dall'auto-encoder.

#### 4.6.1 UMAP (Uniform Manifold Approximation and Projection)

Per la riduzione dimensionale è stato scelto **UMAP**, un algoritmo di riduzione di dimensionalità non lineare. Prima viene applicata la funzione di clustering e successivamente è stato applicato **UMAP** con le seguenti configurazioni:

- `n_components` = 2 o 3, a seconda della visualizzazione desiderata;
- `random_state` = 42 per garantire la riproducibilità;
- impostazioni standard per `n_neighbors` e `min_dist`.

I motivi della scelta di UMAP sono i seguenti:

- **Scalabilità**: UMAP è significativamente più veloce e scalabile rispetto a t-SNE su dataset di grandi dimensioni (oltre 300.000 campioni);
- **Preservazione della topologia**: consente di mantenere la coerenza nella disposizione dei cluster, rendendo l'interpretazione visiva più affidabile;
- **Flessibilità**: supporta la proiezione anche in 3D, facilitando l'esplorazione spaziale della distribuzione dei dati compresi.

#### 4.6.2 Esclusione di t-SNE

**t-SNE** (t-Distributed Stochastic Neighbor Embedding) è stato inizialmente considerato, ma successivamente escluso a causa della sua scarsa efficienza computazionale su dataset di grandi dimensioni. In particolare, t-SNE:

- richiede un tempo computazionale elevato, non compatibile con dataset contenenti centinaia di migliaia di osservazioni;
- non è facilmente parallelizzabile;
- mostra variabilità nei risultati tra esecuzioni diverse, anche con `random_state` fissato.

Pertanto, pur producendo risultati visivamente interessanti, t-SNE non è stato ritenuto adeguato per l'analisi sistematica condotta.

#### 4.6.3 Esclusione di PCA

Anche **PCA** (Principal Component Analysis) è stato testato come metodo alternativo. Tuttavia, essendo una tecnica lineare, PCA non è risultata efficace nel catturare la complessità non lineare degli embedding generati dall'autoencoder. Le principali limitazioni osservate sono:

- perdita di informazione significativa nella proiezione su 2D o 3D;
- ridotta capacità di separazione visiva tra i cluster, rispetto a UMAP.

# Chapter 5

## Risultati sperimentali

In questo capitolo analizzeremo i risultati ottenuti dal modello e dalla clusterizzazione avvenuta in seguito. Riportiamo gli iperparametri descritti nel capitolo precedente:

- `n_epochs = 100`
- `batch_size = 64`
- `learning_rate = 1 × 10-3`
- `hidden_1 = 48, hidden_2 = 24`
- `bottleneck_size = 12`
- `activation_fn = nn.Tanh`
- `Optimizer = Adam`
- `Scheduler = ReduceLROnPlateau`

### 5.1 Andamento del training (MSE e loss)

Questa sezione analizza in dettaglio l'andamento del training dell'autoencoder, concentrandosi maggiormente sull'andamento della loss e del **Mean Squared Error** (MSE). Verrà prima fatta un'analisi generale delle due metriche e successivamente un'analisi sulle ultime 20 epoche, per osservare nel dettaglio il comportamento del modello in fase avanzata. I risultati riportati fanno riferimento:

1. Addestramento con regolarizzazione L1 e L2 impostate a  $1 \times 10^{-5}$
2. Addestramento con regolarizzazione L1 e L2 impostate a 0, quindi senza regolarizzazione

#### 5.1.1 Addestramento con Regolarizzazione L1/L2

##### Andamento della loss

Possiamo notare dal grafico relativo alla loss sull'intero training (vd. figura 5.1a) un comportamento coerente e stabile. La loss decresce progressivamente fin dalle prime epoche, segno che il modello impara da subito una rappresentazione dei dati. Possiamo notare come successivamente si stabilizzi, segno che il modello continua ad apprendere.

**Zoom sulle ultime 20 epoche** : Dal comportamento delle ultime 20 epoche (vd.figura 5.1b) possiamo notare come il modello tenda comunque ad apprendere ma la differenza tra le loss sia minima, questo è segno di stabilità. Il modello non cambia ma affina le sue capacità di ricostruzione.

Risulta evidente come la loss non si annulli o stabilizzi troppo precocemente, questo è segno che il modello non è sovradestrato.

L'adozione di una regolarizzazione abbia contribuito ad evitare instabilità nella fase finale del training

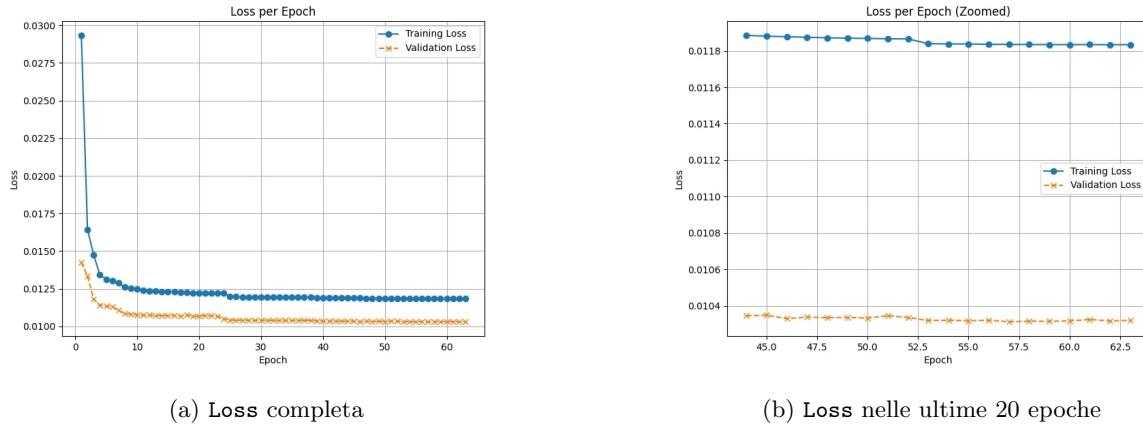


Figure 5.1: Loss di addestramento con regolarizzazione L1/L2

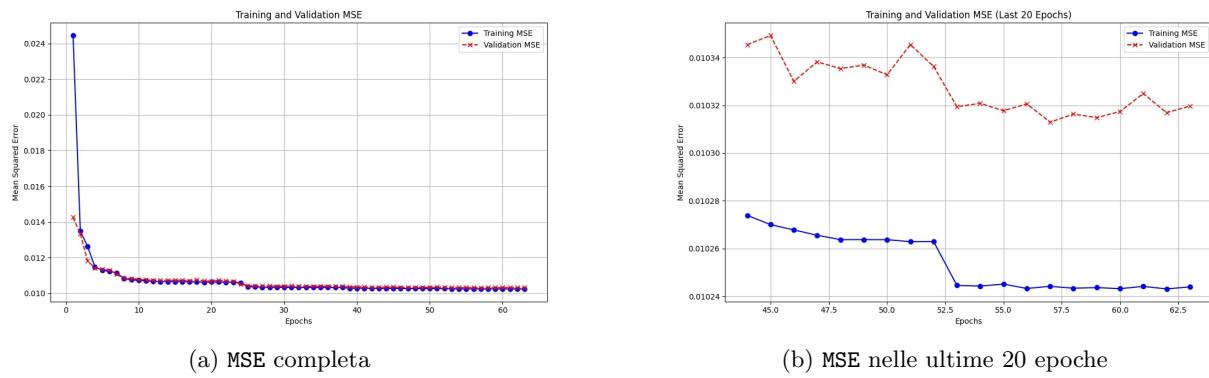


Figure 5.2: MSE di addestramento con regolarizzazione L1/L2

### Andamento del MSE

Il grafico del MSE (vd. figura 5.2a) sull'intero training conferma quanto osservato nella loss. Si rileva una decrescita netta iniziale, seguita da una diminuzione costante, coerente con l'obiettivo di minimizzare l'errore di ricostruzione.

**Zoom sulle ultime 20 epoche** Nel dettaglio, le ultime 20 epoche (vd. figura 5.2b) mostrano un andamento piatto, con variazioni spesso dell'ordine di pochi millesimi. Questo indica che il modello ha raggiunto una rappresentazione stabile, con un margine di miglioramento minimo.

Possiamo notare come nelle ultime 20 epoche (vd. figura 5.2b) sia mostrato un'andamento con variazioni minime con l'ordine di pochi millesimi

Si può constatare che non vi sono episodi di overfitting, il che conferma l'efficacia della regolarizzazione.

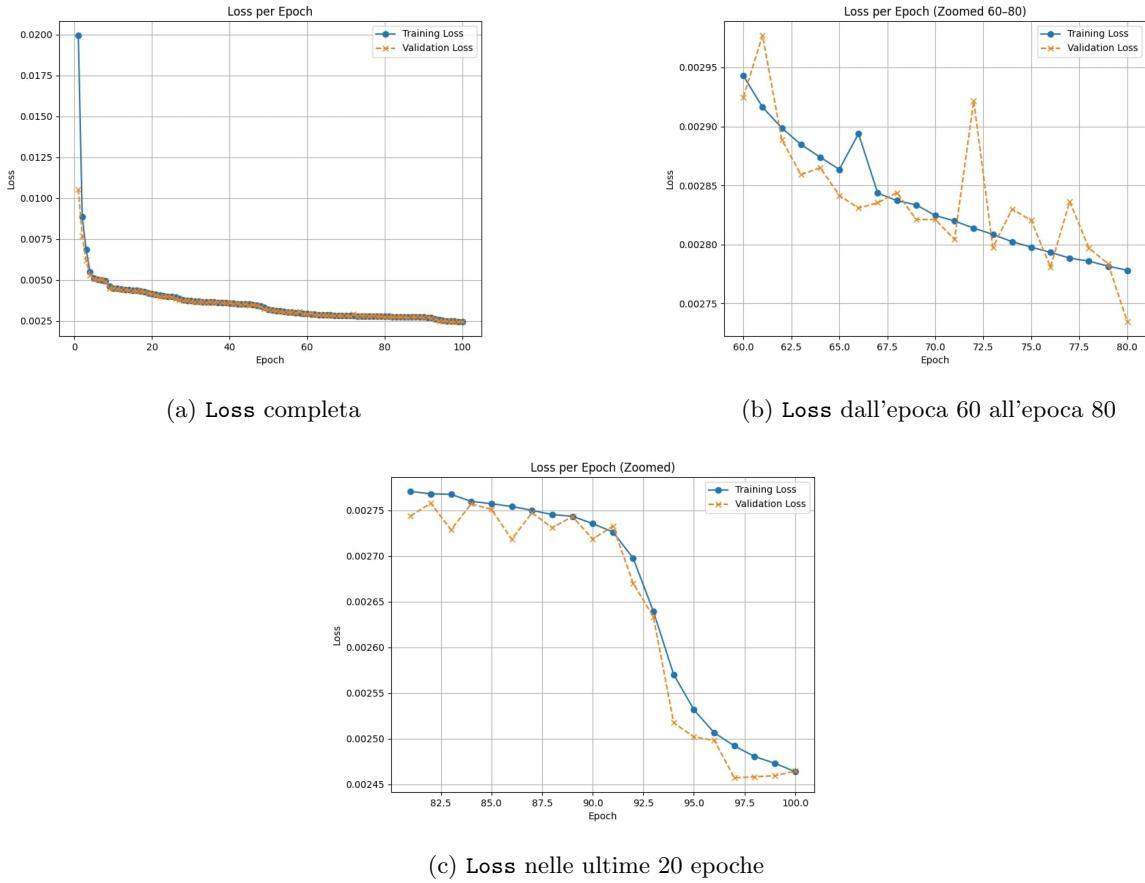
### 5.1.2 Addestramento senza regolarizzazione L1/L2

Nel secondo esperimento, il modello autoencoder è stato addestrato senza regolarizzazione ( $L_1 = L_2 = 0$ ), con l'obiettivo di osservare l'andamento naturale della loss e della MSE senza vincoli.

#### Andamento della loss

L'andamento della loss totale (vd. figura 5.3a) risulta regolare e decrescente nel tempo, senza oscillazioni particolari nel complesso.

La discesa iniziale è piuttosto rapida, seguita da un appiattimento progressivo.

Figure 5.3: `loss` di addestramento con regolarizzazione L1/L2

**Zoom sulle epoche 60–80** Nel grafico (vd. figura 5.3b), che mostra l’andamento della loss tra l’epoca 60 e l’epoca 80, si nota una componente più irregolare e leggermente oscillante, specialmente nella curva della loss di test. Questo effetto è dovuto alla mancanza di regolarizzazione, che consente al modello maggiore libertà di adattarsi ai dati, potenzialmente anche al rumore. L’assenza di una penalizzazione sulla complessità dei pesi permette una discesa continua, ma meno controllata, rendendo il modello più suscettibile a piccoli sbalzi nella curva di test.

**Zoom sulle ultime 20 epoche** Il grafico (vd. figura 5.3c), che si concentra sulle ultime 20 epoche, evidenzia un chiaro rallentamento nella discesa della loss, con miglioramenti sempre più lievi. Questo comportamento indica che il modello si sta progressivamente avvicinando a una fase di stabilizzazione, in cui la capacità di apprendimento si riduce sensibilmente. La relativa stabilità osservata lascia intendere che non vi siano segnali evidenti di overfitting; tuttavia, senza vincoli di regolarizzazione, è possibile che il modello stia iniziando a memorizzare particolarità del dataset di training, a discapito della capacità di generalizzazione. Un simile rallentamento è tipico quando il modello esaurisce le informazioni utili a migliorare ulteriormente la propria performance, senza interventi esterni o modifiche alla configurazione.

### Andamento dell’MSE

L’andamento dell’MSE, come visibile nel grafico (vd. figura 5.4a), rispecchia la traiettoria della `loss`: una decrescita regolare e piuttosto fluida, senza oscillazioni anomale né brusche variazioni. Il calo iniziale è molto rapido, segno che il modello apprende rapidamente le principali strutture informative dei dati. Successivamente, la curva rallenta e si avvicina a un valore di convergenza, coerentemente con quanto osservato nella loss.

**Zoom sulle epoche 60–80** Nel grafico (vd. figura 5.4b), che mostra l’andamento tra le epoche 60 e 80. l’MSE mostra una variazione dei valori più marcata, soprattutto nella curva del test-set. Si osservano alcune

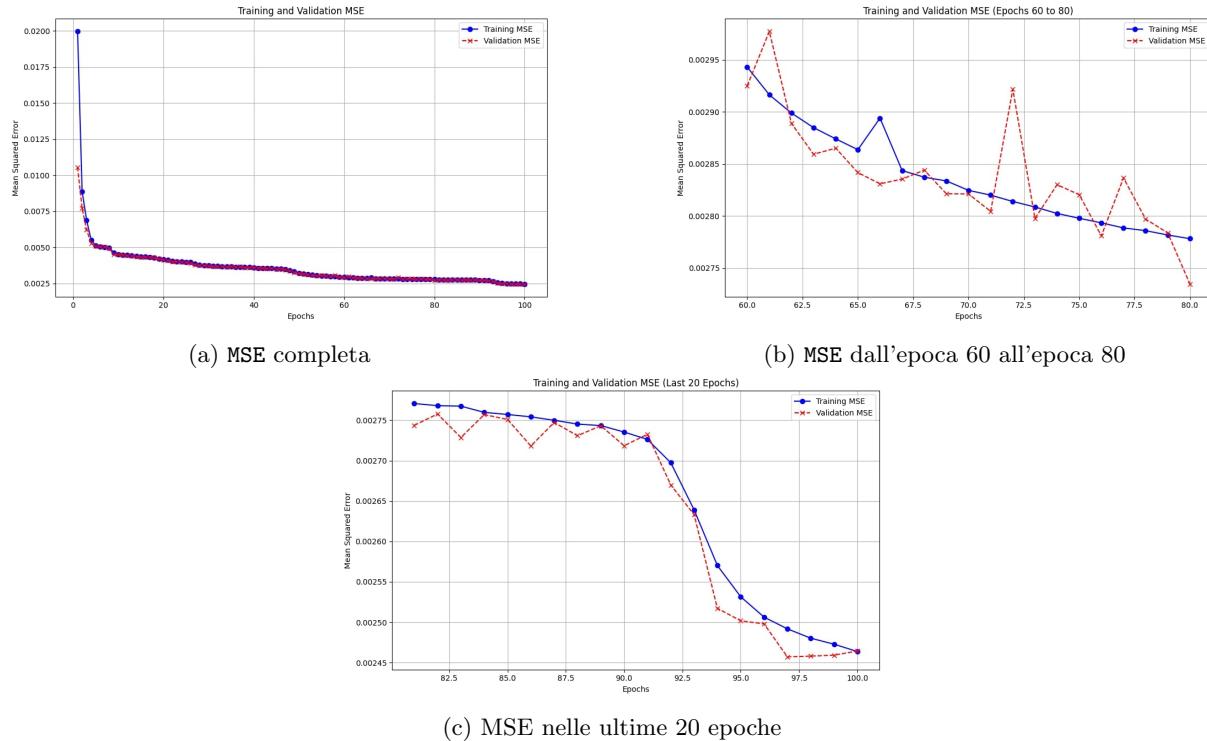


Figure 5.4: MSE di addestramento con regolarizzazione L1/L2

oscillazioni che indicano una fase di ottimizzazione meno fluida. Tuttavia, non si tratta di instabilità grave: il trend generale rimane decrescente, e sia la curva di training che quella di test mostrano una riduzione progressiva dell'errore.

L'assenza di regolarizzazione in questo intervallo si traduce in un modello che può adattarsi con maggiore flessibilità ai dati, ma che al tempo stesso può iniziare a cogliere rumore o dettagli meno generali, producendo piccole oscillazioni nell'errore di ricostruzione.

**Zoom sulle ultime 20 epoche** Il grafico (vd. figura 5.4c), che rappresenta le ultime 20 epoche, non mostra un vero plateau statico: anzi, rispetto alla loss, l'MSE continua a decrescere in modo netto, soprattutto tra l'epoca 80 e l'epoca 90, con un calo ancora visibile nel test del MSE. Dopo l'epoca 90, le curve si avvicinano a una fase più stabile, ma non ancora completamente saturata.

Questo suggerisce che il modello, pur non regolarizzato, continua ad apprendere e migliorare leggermente la qualità della ricostruzione anche in questa fase avanzata. Tuttavia, l'assenza di un meccanismo di regolarizzazione può implicare che questi miglioramenti si ottengano a scapito della generalizzazione, ovvero a vantaggio della precisione sui dati di training ma non necessariamente su nuovi dati.

### 5.1.3 Spiegazione della loss e correlazione della regolarizzazione L1/L2

L'analisi complessiva dei risultati ottenuti evidenzia che l'inserimento dei termini di regolarizzazione L1 e L2 rappresenta un compromesso strategico tra accuratezza della ricostruzione e capacità di generalizzazione del modello.

Nel caso senza regolarizzazione ( $L1 = 0, L2 = 0$ ), il modello riesce a minimizzare efficacemente l'errore di ricostruzione sul training set, ottenendo valori molto bassi di MSE e loss, ma al costo di una maggiore complessità dei pesi. Questa libertà può portare a overfitting, soprattutto in contesti rumorosi o in presenza di pattern non rappresentativi dell'intero dominio.

Al contrario, l'utilizzo della regolarizzazione L1/L2 comporta una perdita controllata in accuratezza (i valori di loss e MSE tendono a essere leggermente più alti), ma garantisce modelli più robusti e meno dipendenti dal dataset di partenza. In particolare:

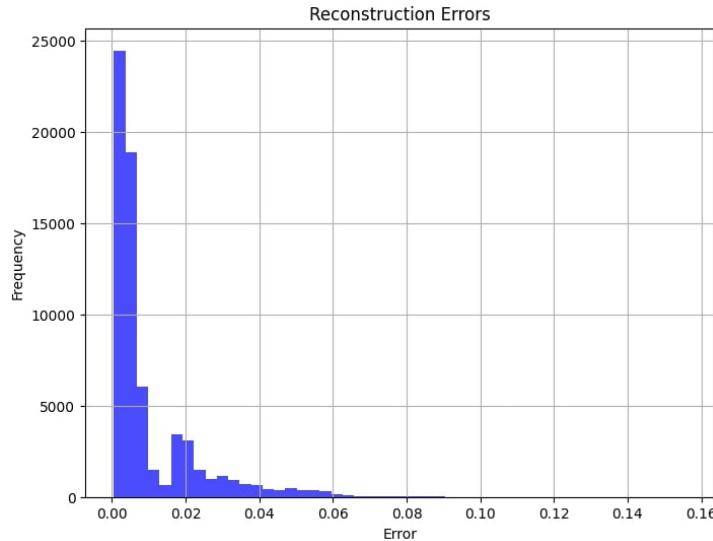


Figure 5.5: Distribuzione completa degli errori di ricostruzione.

- La regolarizzazione L1 favorisce una selezione automatica delle features e induce una maggiore sparsità nei pesi, utile per scoprire rappresentazioni più interpretabili;
- La regolarizzazione L2 contribuisce a evitare pesi estremi, promuovendo una distribuzione più omogenea e stabile nei parametri appresi;
- La presenza della regolarizzazione L1 e L2 dimostra il suo effetto di controllo sulla complessità dei pesi durante l'addestramento. In genere, senza regolarizzazione, i pesi ottimizzati sul training tendono a essere più piccoli rispetto a quelli osservati in fase di test, un comportamento tipico nell'addestramento dei modelli di apprendimento automatico, che può portare a fenomeni di overfitting. Con l'introduzione della regolarizzazione, invece, i pesi del modello durante il training risultano generalmente più elevati, poiché il modello è costretto a bilanciare l'errore di ricostruzione con la penalizzazione sui pesi, riducendo così il rischio di overfitting e migliorando la generalizzazione.

Alla luce dei risultati ottenuti durante la fase di training e testing, l'utilizzo della regolarizzazione ha dimostrato di essere la scelta migliore. Permette infatti di ottenere rappresentazioni più pulite e meno sensibili al rumore, migliorando la qualità dei risultati nelle fasi successive della pipeline analitica.

## 5.2 Ricostruzione e visualizzazione dell'errore

Dopo l'addestramento dell'autoencoder, è stata valutata la sua capacità di apprendere una rappresentazione compatta dei dati e di ricostruirli con elevata fedeltà. A tal fine, sono state calcolate le principali metriche di errore di ricostruzione:

- MSE (Mean Squared Error): 0.010319
- MAE (Mean Absolute Error): 0.041036
- R<sup>2</sup> Score (Coefficient of Determination): 0.594163

Il Mean Squared Error e il Mean Absolute Error indicano un buon livello medio di accuratezza nella ricostruzione, mentre il R<sup>2</sup> Score suggerisce che circa il 59% della varianza presente nei dati originali è spiegata dal modello di ricostruzione. Sebbene tale valore non sia prossimo all'unità, risulta comunque significativo, considerando l'alta complessità e variabilità intrinseca nei comportamenti di mobilità in sharing.

Per un'analisi più dettagliata, è stata generata la distribuzione degli errori di ricostruzione sull'intero dataset (vd. figura 5.5). La distribuzione presenta un unico picco con la maggior parte degli errori concentrati su valori contenuti, mentre la coda asimmetrica verso destra evidenzia la presenza di un numero limitato di osservazioni con errori di ricostruzione elevati.

| #  | Feature                           | Errore Medio Assoluto |
|----|-----------------------------------|-----------------------|
| 1  | eat_and_drinks_start              | 0.441516              |
| 2  | services_start                    | 0.431922              |
| 3  | businesses_start                  | 0.428486              |
| 4  | eat_and_drinks_end                | 0.426903              |
| 5  | sport_and_care_start              | 0.425464              |
| 6  | businesses_end                    | 0.411856              |
| 7  | institutions_and_educations_start | 0.411555              |
| 8  | services_end                      | 0.409655              |
| 9  | institutions_and_educations_end   | 0.402438              |
| 10 | sport_and_care_end                | 0.402318              |

Table 5.1: Top 10 features con il maggiore errore assoluto medio nelle osservazioni con errore di ricostruzione elevato

### 5.3 Identificazione delle osservazioni con errore di ricostruzione elevato

Uno degli obiettivi principali nell'utilizzo dell'`autoencoder` è l'identificazione delle osservazioni per cui la ricostruzione presenta un errore significativamente alto, potenzialmente indicativo di comportamenti atipici o non ben rappresentati dal modello.

#### 5.3.1 Definizione della soglia di anomalia

La selezione delle osservazioni con ricostruzione meno accurata è stata effettuata utilizzando il **Mean Squared Error** tra le features originali e quelle ricostruite. Le osservazioni con un errore superiore a una soglia statistica definita come:

$$\text{soglia} = \mu + 3 \cdot \sigma$$

dove  $\mu$  è la media e  $\sigma$  la deviazione standard degli errori di ricostruzione, sono state classificate come dati ad errore elevato. Nel nostro caso:

- **Soglia definita:** 0.049182
- **Numero di osservazioni con errore elevato:** 1814 su 66.870 (2.71%)

Questa soglia assume che il modello sia stato addestrato principalmente su dati “normali”, per cui un errore elevato identifica osservazioni con caratteristiche più complesse, rare o rumorose.

#### 5.3.2 Analisi delle features con errore elevato

Per ogni feature è stato calcolato l'*errore assoluto medio* tra i valori originali e quelli ricostruiti, al fine di identificare le variabili che maggiormente contribuiscono all'errore di ricostruzione elevato. Le Top 10 features con il più alto errore medio assoluto sono riportate nella **Tabella 5.1**.

Come evidenziato (vd. figura 5.6), gli errori maggiori si concentrano su features legate a valori inconsueti nelle vicinanze di *Point of Interest* (POI), in particolare relativi a ristorazione, servizi, attività commerciali, istituzioni educative e centri sportivi. Tali errori sono soprattutto associati a variabili che descrivono l'inizio e la fine dello spostamento.

#### 5.3.3 Visualizzazione nello spazio latente

Per esplorare ulteriormente la natura degli errori, è stata applicata una riduzione dimensionale mediante UMAP alle differenze assolute tra le features originali e quelle ricostruite. Come mostrato (vd. figura 5.7), la distribuzione nello spazio latente evidenzia la presenza di gruppi distinti, suggerendo diverse categorie di osservazioni con difficoltà di ricostruzione.

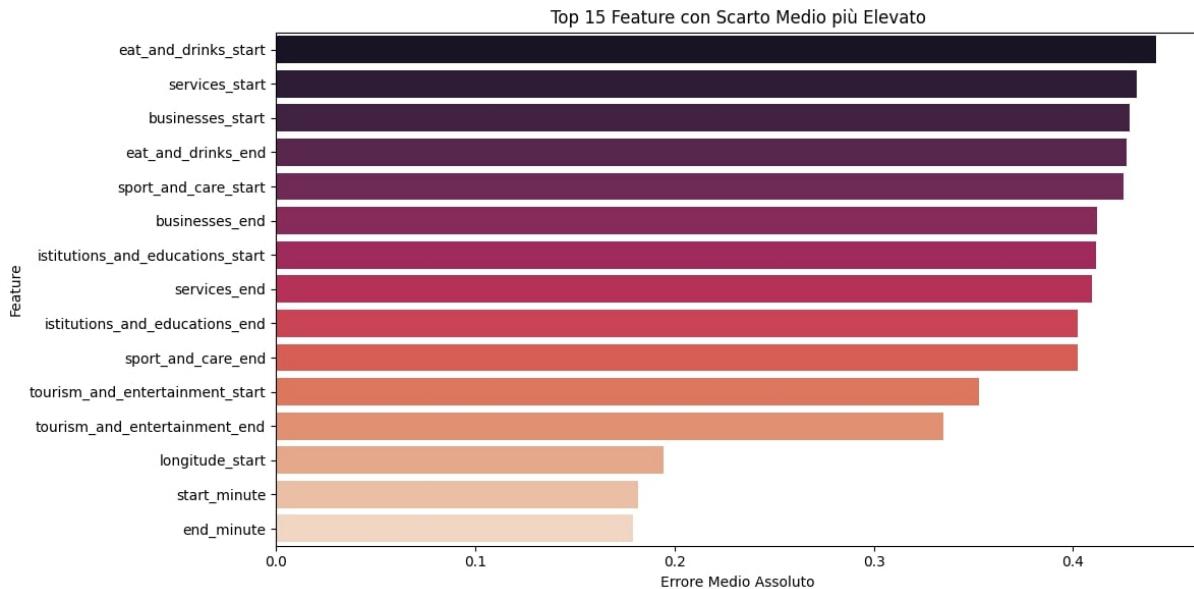


Figure 5.6: features con il maggior errore medio assoluto tra valori originali e ricostruiti.

## 5.4 Valutazione tramite metriche geometriche

In una fase preliminare dell’analisi, si è cercato di valutare le prestazioni del clustering mediante metriche geometriche standard, con l’obiettivo di individuare un valore ottimale del numero di cluster  $k$ . Le metriche adottate sono state:

- **Silhouette Score**, che misura la coesione intra-cluster e la separazione inter-cluster;
- **Davies-Bouldin Score**, che valuta la compattezza e la separabilità dei cluster (valori più bassi sono preferibili);
- **Calinski-Harabasz Score**, che esprime il rapporto tra la dispersione tra i cluster e quella interna ai cluster (valori più alti sono migliori).

L’analisi è stata condotta applicando l’algoritmo **K-means** su embeddings di dimensione 12 ottenuti tramite un autoencoder addestrato con i seguenti iperparametri:

- `n_epochs = 100`
- `batch_size = 64`
- `learning_rate = 1 × 10-3`
- `hidden_1 = 48, hidden_2 = 24`
- `bottleneck_size = 12`
- `activation_fn = nn.Tanh`
- `Optimizer = Adam`
- `Scheduler = ReduceLROnPlateau`

Nella Tabella (vd tabella 5.2) sono riportati i valori delle metriche per diversi valori di  $k$ , da 12 a 2000. Come si può osservare, le metriche geometriche mostrano un trend complesso: il **Silhouette Score** decresce all’aumentare di  $k$ , indicando una minore coesione e separazione tra i cluster, mentre il **Davies-Bouldin Score** migliora leggermente per  $k$  elevati, suggerendo una maggiore compattezza relativa. Il **Calinski-Harabasz Score**, invece, mostra un picco intorno a  $k = 50$ , ma tende a diminuire significativamente per  $k$  superiori.

Tuttavia, questi valori non forniscono un’indicazione univoca per la scelta del numero di cluster, motivo per cui si è optato per un approccio più orientato all’interpretabilità. In particolare, sono stati selezionati diversi valori di  $k$  per esplorare le diverse configurazioni e le diverse modalità:

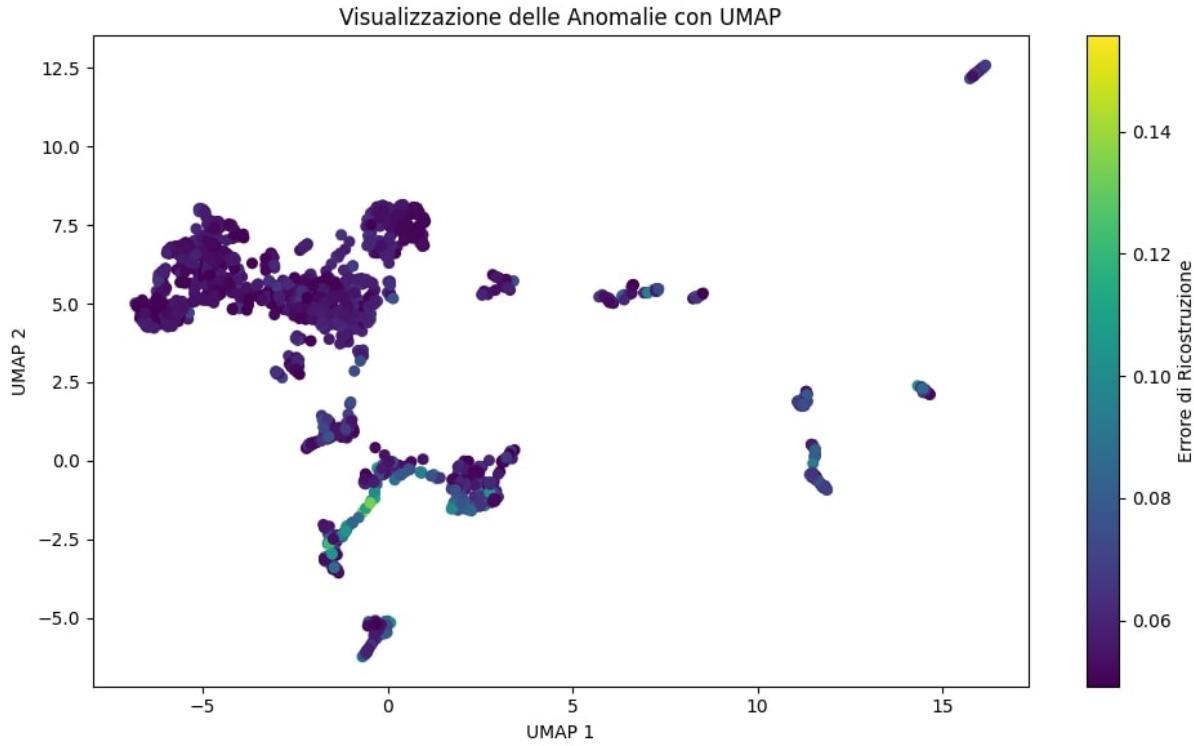


Figure 5.7: Distribuzione delle osservazioni con errore elevato nel piano UMAP.

| Valore di $k$ | Silhouette-Score | Davies-Bouldin-Score | Calinski-Harabasz-Score |
|---------------|------------------|----------------------|-------------------------|
| 12            | 0.3932           | 0.9880               | 152356.44               |
| 14            | 0.4026           | 0.9986               | 156528.48               |
| 50            | 0.3553           | 1.2918               | 177078.59               |
| 100           | 0.2849           | 1.2202               | 137520.62               |
| 500           | 0.2748           | 1.1599               | 87557.89                |
| 1000          | 0.2687           | 1.1464               | 68459.0                 |
| 2000          | 0.2877           | 1.1199               | 54088.91                |

Table 5.2: Valori di valutazione di metriche geometriche

- $k = 14$ : per cogliere eventuali pattern ricorrenti tra i giorni della settimana;
- $k = 50$ : per identificare strutture temporali più raffinate o suddivisioni nei comportamenti di spostamento;
- $k = 500$ : per analizzare nel dettaglio le variazioni su base giornaliera;
- $k = 1000$  e  $k = 2000$ : per catturare informazioni su base spaziale, come la suddivisione delle aree urbane o la prossimità a specifici punti di interesse (POI).

Questo approccio ha permesso di valutare l'efficacia del clustering non solo in termini di compattezza e separabilità, ma anche in relazione alla qualità e alla significatività dei cluster ottenuti dal punto di vista interpretativo.

## 5.5 Clustering dei dati

Dopo aver completato la fase di **training** e compressione dei dati tramite l'autoencoder è possibile procedere con l'analisi dei dati tramite le tecniche di clustering, utilizzate per individuare gruppi di spostamenti con caratteristiche simili.

Il clustering è stato condotto a partire dalle rappresentazioni dello spazio latente ottenuto dall'auto-

`encoder`. Questi valori sono chiamati embeddings e costituiscono una codifica compatta dei dati originali e ne preservano le caratteristiche informative più rilevanti.

È stato effettuato uno studio tramite clustering in due modalità. entrambe le modalità hanno utilizzato lo stesso algoritmo di clustering, ovvero K-Means, con un valore di  $k$  cluster variabile:

- Come prima modalità di analisi, è stato effettuato un clustering con un numero fisso di cluster pari a  $k = 12$ , seguito da una riduzione di dimensionalità tramite UMAP (**Uniform Manifold Approximation and Projection**), una tecnica non lineare utile per proiettare dati ad alta dimensionalità in spazi bidimensionali o tridimensionali. In questo caso, UMAP è stato applicato allo spazio latente ottenuto dopo il clustering, consentendo una visualizzazione sia in 2D che in 3D dei gruppi individuati.
- Come seconda modalità di analisi, è stato effettuato clustering su dati ad alta dimensionalità senza applicare UMAP successivamente, lasciando i dati nel loro spazio originale. In questo caso, il numero di cluster  $k$  è stato variato al fine di esplorare differenti granularità e pattern nei dati. I valori di  $k$  utilizzati durante la sperimentazione vanno da valori bassi come  $k = 14$ , a valori intermedi come  $k = 100$ , fino a valori molto alti come  $k = 2000$ , in modo da catturare differenti caratteristiche e strutture latenti nei dati.

### 5.5.1 Ottenimento degli embeddings

Dopo aver completato la fase di training i dati sono stati compressi negli `embeddings`, che rappresentano ciascun viaggio in uno spazio latente ridotto.

Il modello a partire dai dati originari riduce i dati in uno spazio latente pari alla dimensione data dalla variabile `bottleneck_size` che è stata impostata a 12.

La funzione utilizzata per generare gli `embeddings`, chiamata `save_embeddings`, consente di elaborare i dati e di ridurli nello spazio latente desiderato.

## 5.6 Clustering su dati ridotti con UMAP

Poiché lo spazio degli `embeddings` risultava comunque ad alta dimensionalità (12 dimensioni) per una rappresentazione grafica diretta, si è scelto, in una prima analisi esplorativa, di applicare UMAP. In particolare, sono state adottate due proiezioni distinte:

- Una proiezione in **2D** (vd. figura 5.8a)
- Una proiezione in **3D** (vd. figura 5.8b)

È stata scelta la proiezione sia in 2 dimensioni che in 3 al fine di verificare se la riduzione in due dimensioni comportasse una perdita informativa rispetto alla proiezione in tre dimensioni. Possiamo notare come non vi sia perdita di informazione tramite il controllo della colonna `kmeans_label_UMAP2D` che verifica la somiglianza tra cluster delle due dimensioni. Sono stati selezionati 2 cluster per effettuare un'analisi dei dati e al fine di comprendere eventuali pattern ricorrenti o differenze significative tra i viaggi.

Nell'analisi dei cluster vengono analizzate 5 caratteristiche principali, al fine di strutturare delle conclusioni su quel determinato cluster:

1. **Dimensione del cluster:** indica il numero di spostamenti presenti e la loro percentuale rispetto al totale (334.353).
2. **Caratteristiche temporali degli spostamenti:** analizza giorni e fasce orarie degli spostamenti, durata stimata vs effettiva (inclusi viaggi >24h, se presenti), e distanza media percorsa per identificare pattern temporali ricorrenti.
3. **Mezzi di trasporto utilizzati:** mostra la distribuzione percentuale dei mezzi in sharing (bici, monopattini, scooter, auto), evidenziando prevalenze, assenze o anomalie legate al contesto urbano.
4. **Geolocalizzazione degli spostamenti:** descrive la distanza tra origine e destinazione (media, mediana, deviazione standard) e la distribuzione spaziale nelle zone di Milano (es. centro, periferia, assi principali).

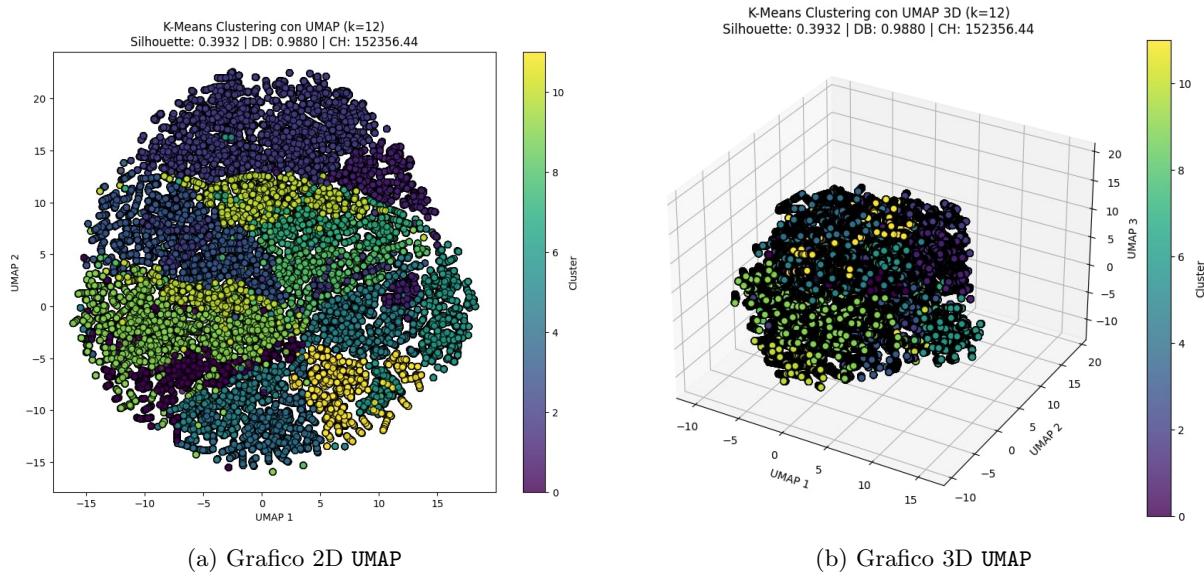


Figure 5.8: Visualizzazione dei dati tramite UMAP in 2D e 3D

**5. Caratteristiche delle zone di partenza e arrivo:** definisce la tipologia delle aree (residenziali, commerciali, ecc.), la presenza di POI e trasporti pubblici (metro, ferrovie), e ricorrenze nei flussi tra zone.

**Conclusione:** riassume i principali risultati del cluster e aiuta a capire che tipo di utenti (es. pendolari, studenti, turisti) lo compongono e come usano la mobilità in città.

Ricapitolando, l'analisi dei cluster tramite algoritmo **k-means**, è stata svolta su dati ricostruiti da un modello con i seguenti iperparametri:

- `n_epochs = 100`
- `batch_size = 64`
- `learning_rate =  $1 \times 10^{-3}$`
- `hidden_1 = 48, hidden_2 = 24`
- `bottleneck_size = 12`
- `activation_fn = nn.Tanh`
- `Optimizer = Adam`
- `Scheduler = ReduceLROnPlateau`

e con un valore di  $k$ -cluster pari a 12 ridotti a 2 o 3 dimensioni.

La riduzione di dimensionalità è stata effettuata sia in 2D che in 3D per visualizzare meglio la struttura dei dati. Dalle analisi è emerso che i cluster ottenuti nelle due proiezioni sono molto simili e non si perde informazione passando da 3 a 2 dimensioni. Per questo motivo, le analisi successive sono state svolte utilizzando la proiezione bidimensionale, più semplice da interpretare.

### 5.6.1 Introduzione

L'analisi approfondita di due cluster, il Cluster 2 (17.04% degli spostamenti) e il Cluster 9 (13.44%), ha permesso di evidenziare due profili distinti di mobilità urbana in sharing all'interno della città di Milano. Sebbene entrambi si sviluppino prevalentemente in aree centrali e semicentrali, in giorni feriali e in presenza di un'elevata densità di servizi, le loro differenze strutturali sono marcate e significative.

Il Cluster 2 è dominato dall'utilizzo di scooter elettrici, mezzi agili e adatti a spostamenti brevi, rapidi e concentrati nelle ore centrali della giornata. Gli spostamenti sono brevi (media distanza di circa 2.17 km) e avvengono quasi esclusivamente nei giorni feriali, con una durata stimata vicina a quella effettiva. Gli

utenti sembrano prediligere un uso funzionale e quotidiano del mezzo, probabilmente per commissioni, brevi spostamenti urbani o attività lavorative distribuite. Le aree attraversate sono dense di POI, con una forte presenza di servizi, istituzioni educative e locali ricreativi, suggerendo una mobilità diffusa e integrata con le funzioni urbane.

Il Cluster 9, al contrario, è composto esclusivamente da spostamenti in automobile, con tratti temporali e spaziali molto più dilatati. Le durate effettive sono nettamente superiori a quelle stimate (in media oltre 100 minuti contro una stima di 8), le distanze percorse sono più che doppie (circa 5.5 km), e l'uso del mezzo è spesso prolungato nel tempo, con frequenti viaggi che si estendono oltre la mezzanotte o su più giorni. La distribuzione settimanale è fortemente concentrata tra lunedì e giovedì, senza attività nel weekend, a conferma della natura lavorativa e strutturata di questi spostamenti. Anche qui, le aree interessate sono dense e centrali, ma l'uso dell'automobile suggerisce esigenze di comfort, trasporto di oggetti o persone, o semplicemente maggiore autonomia rispetto ai vincoli della micro-mobilità.

I risultati evidenziano con chiarezza che è il mezzo di trasporto stesso a modellare l'intero comportamento di mobilità: lo scooter abilita uno stile dinamico, leggero e flessibile su tratte brevi; l'automobile permette uno spostamento più lungo, strutturato e spesso frammentato nel tempo. Questa osservazione è cruciale: al netto di variabili esterne come il giorno della settimana, la zona o la densità urbana, sono le caratteristiche intrinseche del veicolo a determinare la forma e la funzione dello spostamento.

Queste evidenze suggeriscono che una progettazione efficace della mobilità urbana in sharing, sia in termini di offerta che di regolamentazione, dovrebbe tenere conto non solo della domanda e della localizzazione, ma soprattutto del profilo d'uso implicito nei diversi tipi di mezzo. L'integrazione tra scooter e auto, se ben calibrata, può coprire esigenze molto diverse e complementari, riducendo sovrapposizioni e aumentando l'efficienza complessiva del sistema.

## 5.6.2 Cluster 2

### Dimensioni del cluster

Il cluster 2 comprende 56.988 spostamenti equivalenti al 17.04% dell'intero dataset, contenente complessivamente 334.353 spostamenti.

### Caratteristiche temporali degli spostamenti

L'analisi temporale degli spostamenti all'interno di questo cluster rivela un insieme di comportamenti regolari e strutturati, con alcune anomalie significative che meritano una riflessione più approfondita. Esaminiamo le principali dimensioni temporali, considerando i giorni e le fasce orarie in cui si concentrano i viaggi, le durate stimate ed effettive, nonché eventuali pattern ricorrenti. In tabella sono riportate le informazioni temporali riguardo al cluster (vd. tabella 5.3)

**Fasce orarie** L'orario medio di partenza degli spostamenti è compreso nel primo pomeriggio: le medie indicano un avvio attorno alle 13 : 38, con una variabilità significativa (STD = 6.36 ore), che abbraccia quindi sia la mattina che le ore serali. Il 25° percentile si colloca alle ore 9 : 00, la mediana alle 15 : 00 e il 75° percentile alle 19 : 00: ciò suggerisce che il cluster inglobi spostamenti distribuiti tra la tarda mattinata e il tardo pomeriggio, ma con una tendenza centrale marcata tra metà e fine giornata.

Anche l'orario di fine viaggio è coerente con questo schema (media 13 : 37, mediana 15 : 00, con simile distribuzione quartilica), confermando che gli spostamenti hanno una durata contenuta entro la giornata, fatta eccezione per pochi outlier discussi in seguito. L'assenza di concentrazioni mattutine all'alba e serali dopo le 20 indica che non si tratta né di spostamenti pendolari da/per il lavoro a orari classici né di mobilità notturna.

**Durata stimata vs effettiva** Una delle anomalie più evidenti riguarda il confronto tra la durata stimata (*estimated duration mn*) e quella effettiva (*actual duration mn*). La durata stimata si attesta su una media di circa 8.85 minuti, con un range interquartile molto ristretto (da 5 a 12 minuti), suggerendo che la maggior parte dei viaggi sia progettata per essere breve e rapida.

Tuttavia, la durata effettiva è significativamente più elevata, con una media di 30.43 minuti e, soprattutto, una deviazione standard estremamente alta (160.54 minuti). Questo valore è dovuto alla presenza di alcuni casi eccezionali, evidenziati dal massimo registrato di oltre 7124 minuti (quasi 5 giorni), che suggerisce una

| Variabile             | Media | STD    | Min  | 25%   | Mediana | 75%   | Max     |
|-----------------------|-------|--------|------|-------|---------|-------|---------|
| estimated_duration_mn | 8.85  | 5.80   | 0.0  | 5.00  | 8.00    | 12.00 | 59.00   |
| actual_duration_mn    | 30.43 | 160.54 | 2.63 | 10.15 | 15.05   | 29.44 | 7124.63 |
| start_hour            | 13.65 | 6.36   | 0.0  | 9.00  | 15.00   | 19.00 | 23.00   |
| start_minute          | 30.68 | 17.32  | 0.0  | 14.00 | 30.0    | 45.00 | 59.00   |
| start_day             | 15.65 | 8.79   | 1.00 | 8.00  | 15.00   | 23.00 | 31.00   |
| start_month           | 6.72  | 3.39   | 1.00 | 4.00  | 7.00    | 10.0  | 12.00   |
| end_hour              | 13.61 | 6.49   | 0.0  | 9.00  | 15.00   | 19.00 | 23.00   |
| end_minute            | 30.44 | 17.35  | 0.0  | 14.00 | 29.00   | 44.00 | 59.00   |
| end_day               | 15.66 | 8.79   | 1.00 | 8.00  | 15.00   | 23.00 | 31.00   |
| end_month             | 6.73  | 3.39   | 1.00 | 4.00  | 7.00    | 10.0  | 12.00   |
| weekday_num_start     | 3.20  | 1.38   | 1.00 | 2.00  | 3.00    | 4.00  | 6.00    |
| weekday_num_end       | 3.22  | 1.38   | 1.00 | 2.00  | 3.00    | 4.00  | 7.00    |
| start_monday          | 0.16  | 0.37   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_tuesday         | 0.17  | 0.37   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_wednesday       | 0.23  | 0.42   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_thursday        | 0.21  | 0.41   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_friday          | 0.24  | 0.43   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_saturday        | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_sunday          | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_monday            | 0.15  | 0.37   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_tuesday           | 0.17  | 0.37   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_wednesday         | 0.23  | 0.42   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_thursday          | 0.21  | 0.41   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_friday            | 0.24  | 0.43   | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_saturday          | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_sunday            | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_weekend         | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_weekend           | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_public_holiday  | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_public_holiday    | 0.0   | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| holiday_start         | 0.0   | 0.2    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| holiday_end           | 0.0   | 0.2    | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |

Table 5.3: Statistiche descrittive delle caratteristiche temporali del cluster 2

durata anomala. In queste circostanze è possibile che l'orario di fine risulti apparentemente antecedente all'orario di inizio, se si considera solo l'orario (senza data) oppure a causa di errori nel recupero dei timestamp. In alternativa, questi valori potrebbero rappresentare veicoli parcheggiati per lunghi periodi senza terminare formalmente il viaggio — fenomeno tipico nei servizi di mobilità condivisa.

Escludendo tali outlier, la mediana della durata effettiva è comunque quasi doppia rispetto alla stima iniziale (15.05 vs 8 minuti), indicando una sottostima sistematica o una frequente inefficienza nello svolgimento del viaggio, forse legata a congestione, attese o interruzioni.

**Pattern ricorrenti** Il cluster è caratterizzato da spostamenti brevi ma relativamente dilazionati nel tempo. La distribuzione oraria e giornaliera suggerisce un uso non impulsivo né continuativo dei mezzi, ma piuttosto occasionale e concentrato in fasce centrali della giornata. L'assenza nel weekend lo distingue nettamente da altri cluster più ricreativi, mentre la differenza tra durata stimata ed effettiva rivela problemi di puntualità o uso prolungato non previsto.

Inoltre, considerando che lo scarto tra orari di inizio e fine è spesso ridotto, ma con valori estremi elevati, possiamo ipotizzare l'esistenza di due sottogruppi all'interno del cluster: uno maggioritario, con viaggi brevi e coerenti con la pianificazione, e uno minoritario, caratterizzato da lunghi stalli o uso improprio dei mezzi.

| Variabile    | Media | STD  | Min | 25%  | Mediana | 75%  | Max  |
|--------------|-------|------|-----|------|---------|------|------|
| Car          | 0.0   | 0.0  | 0.0 | 0.0  | 0.0     | 0.0  | 0.0  |
| Bike         | 0.18  | 0.39 | 0.0 | 0.0  | 0.0     | 0.0  | 1.00 |
| Motorscooter | 0.0   | 0.0  | 0.0 | 0.0  | 0.0     | 0.0  | 0.0  |
| Scooter      | 0.82  | 0.39 | 0.0 | 1.00 | 1.00    | 1.00 | 1.00 |

Table 5.4: Statistiche descrittive sull'utilizzo dei veicoli del cluster 2

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 2168.55   | 1432.79  | 101.00    | 1133.75   | 1840.0    | 2854.00   | 15324.00  |
| latitude_start       | 45.469852 | 0.015359 | 45.433190 | 45.458290 | 45.469400 | 45.481310 | 45.524620 |
| longitude_start      | 9.192369  | 0.023790 | 9.084430  | 9.176110  | 9.193415  | 9.209770  | 9.249310  |
| latitude_end         | 45.469912 | 0.015396 | 45.431740 | 45.458280 | 45.469740 | 45.481310 | 45.525380 |
| longitude_end        | 9.192224  | 0.023877 | 9.084410  | 9.176000  | 9.193260  | 9.209800  | 9.278630  |

Table 5.5: Caratteristiche geospatiali del cluster 2

### Mezzi di trasporto utilizzati

La distribuzione dei mezzi in sharing utilizzati nel cluster 2 (vd. tabella 5.4) mostra una netta predominanza dello scooter elettrico (81.6%), seguito da una quota decisamente inferiore di automobili in sharing (18.4%), mentre biciclette e motorini a benzina risultano assenti.

Questa ripartizione è significativa per almeno due motivi:

- Il forte ricorso agli scooter elettrici evidenzia una preferenza per mezzi agili, flessibili, e adatti a spostamenti rapidi su brevi distanze, specialmente nelle ore centrali della giornata e in contesti urbani densi.
- L'assenza di biciclette suggerisce che il cluster non rappresenta una mobilità *dolce o sostenibile* legata allo sport o al tempo libero, mentre l'assenza di motorini a benzina riflette l'evoluzione dell'offerta dei servizi di sharing, sempre più orientata all'elettrico.

La presenza minoritaria dell'auto potrebbe indicare che, sebbene alcuni spostamenti richiedano maggiore comfort o copertura, il contesto e la durata favoriscono mezzi meno ingombranti e più economici.

### Geolocalizzazione degli spostamenti

Dal punto di vista spaziale (vd. tabella 5.5), la distanza tra punto di partenza e destinazione ha una media di circa 2.17 km, con una mediana inferiore a 2 km, e una deviazione standard di 1.43 km. Questo indica una concentrazione di spostamenti su tratte brevi, ma con una discreta variabilità: accanto a spostamenti estremamente locali, ve ne sono alcuni che coprono aree più estese.

La geolocalizzazione (media delle coordinate) è centrata attorno a latitudine 45.4698 e longitudine 9.1923, corrispondente all'area centrale di Milano, tra Porta Venezia, Centrale e Città Studi. Tuttavia, la dispersione dei dati mostra che gli spostamenti non si limitano a una singola zona, ma interessano tutto il tessuto urbano centrale e semicentrale, con punte in zone come **Navigli, Porta Romana, Isola e Bicocca**.

Questa distribuzione suggerisce che il cluster copre una mobilità intra-urbana, spesso intra-quartiere, ma con possibilità di coprire tratte anche di 3-4 km, magari in assenza di alternative efficienti di trasporto pubblico o nei momenti meno serviti.

### Caratteristiche delle zone di partenza e arrivo

Le aree di partenza e arrivo risultano ricche di punti di interesse (POI), con una media di circa 1034 POI all'origine e 1023 alla destinazione. La presenza diffusa di POI suggerisce che gli spostamenti avvengono prevalentemente in aree dense e multifunzionali, con un'elevata offerta di servizi, esercizi commerciali e funzioni pubbliche.

Più nel dettaglio, le categorie prevalenti nei luoghi di partenza e arrivo sono (vd. tabella 5.6):

| Variabile                         | Media   | STD    | Min | 25%    | Mediana | 75%     | Max     |
|-----------------------------------|---------|--------|-----|--------|---------|---------|---------|
| num_ferrovie_start                | 0.07    | 0.25   | 0.0 | 0.0    | 0.0     | 0.0     | 1.00    |
| num_superficie_start              | 4.42    | 2.88   | 0.0 | 2.00   | 4.00    | 6.00    | 16.00   |
| num_metro_start                   | 0.36    | 0.62   | 0.0 | 0.0    | 0.0     | 1.00    | 3.00    |
| num_POI_start                     | 1034.50 | 726.48 | 0.0 | 510    | 864     | 1362    | 4608    |
| tourism_and_entertainment_start   | 0.997   | 0.05   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| eat_and_drinks_start              | 0.99    | 0.10   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| sport_and_care_start              | 0.995   | 0.07   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| services_start                    | 0.99    | 0.08   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| businesses_start                  | 0.996   | 0.06   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| institutions_and_educations_start | 0.99    | 0.08   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| num_ferrovie_end                  | 0.07    | 0.25   | 0.0 | 0.0    | 0.0     | 0.0     | 1.00    |
| num_metro_end                     | 0.36    | 0.61   | 0.0 | 0.0    | 0.0     | 1.00    | 3.00    |
| num_superficie_end                | 4.41    | 2.87   | 0.0 | 2.00   | 4.00    | 6.00    | 18.00   |
| num_POI_end                       | 1023.12 | 709.19 | 0.0 | 504.00 | 870.0   | 1356.00 | 4566.00 |
| tourism_and_entertainment_end     | 0.996   | 0.06   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| eat_and_drinks_end                | 0.99    | 0.11   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| sport_and_care_end                | 0.993   | 0.08   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| services_end                      | 0.99    | 0.08   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| businesses_end                    | 0.99    | 0.07   | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |
| institutions_and_educations_end   | 0.99    | 0.097  | 0.0 | 1.0    | 1.0     | 1.0     | 1.0     |

Table 5.6: Caratteristiche delle zone di partenza e arrivo cluster 2

- Eat & Drinks e Tourism & Entertainment quasi sempre presenti (> 98% dei casi), indicando che gli spostamenti sono spesso legati a ristorazione, locali, attività culturali e tempo libero.
- Seguono Services e Businesses, anch'essi con frequenza prossima al 99%, a conferma dell'impronta funzionale degli spostamenti.
- institutions and Education sono presenti nel 99% dei casi, evidenziando una forte componente legata a scuole, università o enti pubblici.

La rete di trasporto pubblico è discretamente rappresentata: in media sono presenti 4.4 linee di superficie e 0.36 stazioni metro nei pressi della partenza, con valori simili all'arrivo. Tuttavia, la presenza molto bassa di stazioni ferroviarie (0.068) indica che gli spostamenti avvengono all'interno del tessuto urbano e non sono intermodali con treni regionali o nazionali.

### Conclusione

Il cluster 2 rappresenta una componente centrale e strutturale della mobilità urbana in sharing a Milano, caratterizzata da:

- Spazi urbani centrali o semicentrali, densi di servizi e attrazioni;
- Spostamenti brevi, efficienti e frequenti, concentrati in giorni feriali e nelle ore centrali della giornata;
- Una netta preferenza per gli scooter elettrici, che coniugano flessibilità, velocità e semplicità d'uso;
- Origine e destinazione in aree ad alta densità di POI, con una forte presenza di attività lavorative, scolastiche e ricreative.

Nel complesso, questo cluster riflette il comportamento di utenti urbani dinamici, probabilmente giovani o adulti attivi, che utilizzano lo sharing come alternativa rapida e flessibile al trasporto pubblico, soprattutto per gestire spostamenti quotidiani, commissioni o impegni tra le diverse polarità della città.

### 5.6.3 Cluster 9

#### Dimensioni del cluster

Il cluster 9 contiene 44.954 spostamenti, che rappresentano circa il 13.44% dell'intero dataset di 334.353 spostamenti.

### Caratteristiche temporali degli spostamenti

L'analisi temporale di questo cluster (vd. tabella 5.7) mostra un quadro complesso e piuttosto interessante che merita una disamina articolata per cogliere pattern, anomalie e interpretazioni plausibili.

**Distribuzione dei giorni della settimana e presenza nei fine settimana** Gli spostamenti si concentrano prevalentemente nei giorni feriali, con una netta predominanza di lunedì, mercoledì e giovedì. La frequenza di inizio viaggio è molto alta in questi tre giorni, rispettivamente intorno 31.5%, 33.5% e 31.2%, mentre gli altri giorni mostrano valori marginali o quasi nulli, con venerdì, sabato e domenica quasi assenti. In particolare, i dati indicano praticamente nessun viaggio iniziato o terminato nel weekend o nei giorni festivi, suggerendo un cluster fortemente legato a attività lavorative o istituzionali, tipiche della settimana lavorativa.

**Fasce orarie di partenza e arrivo** L'analisi degli orari di partenza evidenzia una media intorno alle 14 : 29 con una deviazione standard di circa 6.2 ore per l'ora e 17.4 minuti per i minuti. La mediana dell'ora di inizio è a circa 14 : 29, confermando che la maggior parte degli spostamenti inizia nel primo pomeriggio.

Gli orari di arrivo, invece, risultano mediamente alle 13 : 48, con deviazioni standard simili. La mediana di arrivo (ora 15, minuti 30) è leggermente più alta, ma comunque in linea con l'interpretazione di viaggi che si concludono nel primo pomeriggio del giorno successivo o di giorni vicini.

L'apparente anomalia per cui l'orario medio di arrivo risulti antecedente all'orario medio di partenza suggerisce che molti spostamenti o durano più di un giorno, ovvero attraversano la mezzanotte o si prolungano per più giorni oppure sono concentrati maggiormente nella mattinata (plausibile a causa della STD dei viaggi di partenza elevati). Questo dato è coerente con la forte discrepanza osservata tra la durata stimata (media circa 8 minuti) e la durata effettiva (media circa 104 minuti, con una deviazione standard molto ampia), indicando la presenza di viaggi pendolari con soste intermedie o pause prolungate, oppure viaggi di lunga durata.

Questi risultati sono supportati anche dall'analisi delle date di partenza e arrivo: la mediana del giorno di partenza è 16 e quella del giorno di arrivo 16, con una deviazione standard attorno a 9 giorni, confermando una variazione temporale significativa nei dati. I mesi medi di partenza e arrivo sono entrambi intorno a luglio (7.25), suggerendo una distribuzione stagionale coerente.

Infine, l'analisi dei giorni della settimana indica che la maggior parte degli spostamenti inizia e termina tra lunedì e giovedì, con poche partenze o arrivi nel weekend, il che rafforza l'ipotesi di spostamenti principalmente lavorativi o pendolari.

**Durata stimata vs durata effettiva** La durata stimata media è di 8.17 minuti con mediana a 8 minuti, un valore molto basso e con scarto contenuto (deviazione standard di 4.6 minuti). Questo indica che la previsione o il modello di durata stimata considera prevalentemente spostamenti brevi o veloci.

Al contrario, la durata effettiva media è estremamente alta, pari a circa 103.94 minuti, con una deviazione standard molto ampia (309.42 minuti) e un massimo che supera le 4319.73 minuti (circa 3 giorni). La mediana è invece 40.07 minuti, molto più bassa della media ma ancora di gran lunga superiore alla stima, e il 75° percentile a 65.07 minuti. Questo indica che la maggior parte degli spostamenti effettivi supera abbondantemente la durata stimata, con alcuni casi molto lunghi che alzano la media.

Questa differenza suggerisce la presenza di spostamenti con tempi di attesa, soste o ritardi non previsti nel modello di durata stimata, oppure viaggi particolarmente lunghi (anche di più giorni) che fanno aumentare la variabilità e la media.

**Distanza e durata: pattern temporali** Non disponiamo della distanza media percorsa direttamente, ma la breve durata stimata combinata con la lunga durata effettiva e gli orari anomali fa ipotizzare un cluster che comprende:

- Spostamenti brevi in termini di distanza (pochi minuti stimati), ma che in realtà si protraggono molto nel tempo per motivi di soste, pause o attività intermedie.

| Variabile             | Media  | STD    | Min  | 25%   | Mediana | 75%   | Max     |
|-----------------------|--------|--------|------|-------|---------|-------|---------|
| estimated_duration_mn | 8.17   | 4.60   | 0.0  | 5.0   | 8.0     | 11.0  | 34.0    |
| actual_duration_mn    | 103.94 | 309.42 | 2.47 | 29.87 | 40.07   | 65.07 | 4319.73 |
| start_hour            | 14.15  | 6.21   | 0.0  | 9.0   | 15.0    | 19.0  | 23.0    |
| start_minute          | 28.99  | 17.41  | 0.0  | 14.0  | 29.0    | 44.0  | 59.0    |
| start_day             | 15.94  | 8.86   | 1.0  | 8.0   | 16.0    | 23.0  | 31.0    |
| start_month           | 7.25   | 3.38   | 1.0  | 4.0   | 8.0     | 10.0  | 12.0    |
| end_hour              | 13.80  | 6.66   | 0.0  | 9.0   | 15.0    | 19.0  | 23.0    |
| end_minute            | 29.21  | 17.49  | 0.0  | 14.0  | 30.0    | 45.0  | 59.0    |
| end_day               | 15.91  | 8.86   | 1.0  | 8.0   | 16.0    | 23.0  | 31.0    |
| end_month             | 7.25   | 3.39   | 1.0  | 4.0   | 8.0     | 10.0  | 12.0    |
| weekday_num_start     | 2.66   | 1.24   | 1.0  | 1.0   | 3.0     | 4.0   | 6.0     |
| weekday_num_end       | 2.72   | 1.22   | 1.0  | 1.0   | 3.0     | 4.0   | 6.0     |
| start_monday          | 0.315  | 0.465  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| start_tuesday         | 0.033  | 0.178  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_wednesday       | 0.335  | 0.472  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| start_thursday        | 0.312  | 0.463  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| start_friday          | 0.001  | 0.032  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_saturday        | 0.004  | 0.059  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_sunday          | 0.0    | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_monday            | 0.298  | 0.457  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| end_tuesday           | 0.029  | 0.167  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_wednesday         | 0.333  | 0.471  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| end_thursday          | 0.339  | 0.473  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |
| end_friday            | 0.0    | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_saturday          | 0.001  | 0.031  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_sunday            | 0.0    | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_weekend         | 0.004  | 0.059  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| end_weekend           | 0.001  | 0.031  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_public_holiday  | 0.0    | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_public_holiday    | 0.0    | 0.0    | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| holiday_start         | 0.023  | 0.151  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| holiday_end           | 0.024  | 0.152  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |

Table 5.7: Statistiche descrittive delle caratteristiche temporali del cluster 9

- Possibile presenza di viaggi iniziati nel pomeriggio di un giorno e conclusi il giorno successivo o anche oltre, che giustificano il fatto che l'orario di fine è più basso rispetto all'orario di inizio e la durata effettiva è molto estesa.

**Pattern orari e riflessioni sul calendario** Il fatto che i viaggi si concentrino quasi esclusivamente tra lunedì e giovedì, senza spostamenti nel weekend e praticamente nessun viaggio il venerdì, può riflettere:

- Un comportamento lavorativo molto regolare e “strutturato”, forse legato a impieghi con orari rigidi.
- Eventuali assenze o riduzione di mobilità in corrispondenza del weekend o venerdì, suggerendo che il cluster rappresenti lavoratori o utenti con ritmi settimanali particolari.

I dati sulle festività sono nulli, confermando che questo cluster non registra viaggi in giorni festivi o durante le vacanze, rafforzando l'ipotesi di attività legate a routine lavorative.

### Mezzi di trasporto utilizzati

Il cluster mostra una fortissima predominanza dell'automobile (vd. tabella 5.8) in sharing, utilizzata nel 100% degli spostamenti. Nessun viaggio è stato compiuto con biciclette, monopattini o scooter. Questa unicità del mezzo impiegato è molto significativa: si tratta di un cluster che rappresenta esclusivamente la mobilità condivisa su quattro ruote.

La totale assenza di altri mezzi indica un comportamento utente specifico, probabilmente orientato a spostamenti più comodi, protetti dalle intemperie, o su distanze che renderebbero scomoda o inefficiente la micro-mobilità. Potrebbe coinvolgere utenti con esigenze di trasporto regolari e mediamente impegnative, come tragitti casa-lavoro o casa-scuola, dove la presenza di bagagli o passeggeri potrebbe rendere l'auto preferibile.

### Geolocalizzazione degli spostamenti

La distanza media tra origine e destinazione si aggira intorno ai 5.5 km, con una mediana di circa 5 km, valori che indicano un utilizzo all'interno dei confini urbani ma non limitato a quartieri circoscritti. La deviazione standard di oltre 3.3 km conferma una certa varietà negli itinerari.

| Variabile    | Media | STD | Min | 25% | Mediana | 75% | Max |
|--------------|-------|-----|-----|-----|---------|-----|-----|
| Car          | 1.0   | 0.0 | 1.0 | 1.0 | 1.0     | 1.0 | 1.0 |
| Bike         | 0.0   | 0.0 | 0.0 | 0.0 | 0.0     | 0.0 | 0.0 |
| Motorscooter | 0.0   | 0.0 | 0.0 | 0.0 | 0.0     | 0.0 | 0.0 |
| Scooter      | 0.0   | 0.0 | 0.0 | 0.0 | 0.0     | 0.0 | 0.0 |

Table 5.8: Statistiche descrittive sull'utilizzo dei veicoli del cluster 9

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 5549.62   | 3389.74  | 101.00    | 2979.0    | 5007.0    | 7680.0    | 26146.0   |
| latitude_start       | 45.472492 | 0.023763 | 45.358300 | 45.4000   | 45.471610 | 45.489920 | 45.585330 |
| longitude_start      | 9.185039  | 0.035430 | 9.068340  | 9.159430  | 9.185570  | 9.210930  | 9.291890  |
| latitude_end         | 45.472469 | 0.023914 | 45.358320 | 45.453970 | 45.471720 | 45.489800 | 45.574530 |
| longitude_end        | 9.185263  | 0.035763 | 9.071030  | 9.159502  | 9.186080  | 9.211010  | 9.291900  |

Table 5.9: Caratteristiche geospaziali del cluster 9

Dal punto di vista spaziale, la latitudine e longitudine medie sia di partenza che di arrivo coincidono quasi perfettamente (rispettivamente 45.4725°N, 9.185°E), localizzando gli spostamenti in una zona centrale di Milano, verosimilmente tra Porta Venezia, Porta Romana e Porta Garibaldi, ovvero l'anello semicentrale interno alla circonvallazione. Tuttavia, il range di coordinate mostra che i movimenti si estendono anche verso sud e ovest (minimo 45.358°, zona Barona/Rozzano), a indicare una certa copertura metropolitana ampia, pur restando nei limiti cittadini.

La presenza media di POI nelle aree di partenza e arrivo è elevata (oltre 595 POI medi per area), con punte massime che raggiungono valori molto elevati. Ciò riflette la presenza di numerose attrazioni e servizi, tipiche delle aree centrali o dei nodi strategici di interscambio urbano (vd. tabella 5.9).

### Caratteristiche delle zone di partenza e arrivo

Le zone di partenza e arrivo presentano un profilo urbano articolato, ma con una spiccata densità di servizi. Le categorie più rappresentate includono (vd tabella 5.10):

- Servizi e attività economiche, con quasi la totalità degli spostamenti che originano e terminano in aree con disponibilità di servizi (94%).
- Punti di interesse legati a turismo e intrattenimento, ristorazione, attività sportive e istituzioni educative sono presenti in media nel 92-96% dei casi, evidenziando una centralità funzionale e sociale delle aree coinvolte.

Anche la presenza di mezzi pubblici è significativa, con una media di circa 3.7 linee di superficie e una presenza non trascurabile di ferrovie e metro. Questo suggerisce che gli spostamenti avvengano in aree ben servite, ma per motivi legati a comodità, trasferimento porta a porta o flessibilità temporale, l'utente opta comunque per l'auto in sharing.

L'elevata concentrazione di servizi e POI sia all'origine che alla destinazione evidenzia un pattern interessante: questi spostamenti iniziano e terminano in aree dense, ma sono effettuati in auto piuttosto che con mezzi pubblici, forse per motivi legati a privacy, urgenza, o trasporto multiplo.

### Conclusione

Il cluster 9 rappresenta una componente rilevante della mobilità urbana in sharing, con tratti distintivi che lo differenziano chiaramente da altri profili. In particolare, si evidenziano:

- Una netta prevalenza nei giorni feriali, soprattutto tra lunedì e giovedì, e una quasi totale assenza nel weekend, suggerendo una mobilità associata ad attività lavorative, scolastiche o istituzionali.
- Una concentrazione degli spostamenti nel primo pomeriggio, accompagnata da una certa variabilità oraria, che può indicare flessibilità negli orari o la presenza di interruzioni e soste prolungate.

| Variabile                         | Media  | STD    | Min | 25%   | Mediana | 75%    | Max    |
|-----------------------------------|--------|--------|-----|-------|---------|--------|--------|
| num_ferrovie_start                | 0.032  | 0.175  | 0.0 | 0.0   | 0.0     | 0.0    | 1.0    |
| num_superficie_start              | 3.69   | 2.55   | 0.0 | 2.0   | 3.0     | 5.0    | 17.0   |
| num_metro_start                   | 0.186  | 0.440  | 0.0 | 0.0   | 0.0     | 0.0    | 3.0    |
| num_POI_start                     | 595.14 | 503.81 | 0.0 | 222.0 | -462.0  | -846.0 | 4530.0 |
| tourism_and_entertainment_start   | 0.970  | 0.172  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| eat_and_drinks_start              | 0.926  | 0.262  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| sport_and_care_start              | 0.941  | 0.236  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| services_start                    | 0.947  | 0.225  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| businesses_start                  | 0.948  | 0.221  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| institutions_and_educations_start | 0.941  | 0.236  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| num_ferrovie_end                  | 0.034  | 0.180  | 0.0 | 0.0   | 0.0     | 0.0    | 1.0    |
| num_metro_end                     | 0.187  | 0.436  | 0.0 | 0.0   | 0.0     | 0.0    | 0.0    |
| num_superficie_end                | 3.71   | 2.59   | 0.0 | 2.0   | 3.0     | 5.0    | 17.0   |
| num_POI_end                       | 596.37 | 509.16 | 0.0 | 222.0 | 462.0   | 846.0  | 4554.0 |
| tourism_and_entertainment_end     | 0.965  | 0.183  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| eat_and_drinks_end                | 0.921  | 0.270  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| sport_and_care_end                | 0.936  | 0.245  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| services_end                      | 0.941  | 0.235  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| businesses_end                    | 0.942  | 0.233  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |
| institutions_and_educations_end   | 0.934  | 0.249  | 0.0 | 1.0   | 1.0     | 1.0    | 1.0    |

Table 5.10: Caratteristiche delle zone di partenza e arrivo cluster 9

- Una marcata discrepanza tra durata stimata e durata effettiva, con tempi effettivi mediamente più elevati e una deviazione standard significativa, indice di un utilizzo frazionato o prolungato del mezzo.
- La presenza di viaggi che si estendono su più giorni o che iniziano e terminano in date differenti, suggerendo comportamenti di mobilità pendolare flessibile o usi non lineari del servizio.

Nel complesso, il cluster restituisce un’immagine di mobilità strutturata ma non rigida, spesso associata a esigenze professionali gestite con una certa elasticità. Questo profilo risulta particolarmente interessante per individuare eventuali criticità nell’efficienza operativa del servizio e per sviluppare strategie finalizzate a ridurre i tempi di inattività e migliorare la rotazione del parco veicoli.

#### 5.6.4 Analisi comparativa dei cluster 2 e 9

L’analisi ha messo in evidenza due gruppi particolarmente significativi per coerenza interna e rilevanza comportamentale:

- Il cluster 2 con il 17.04% degli spostamenti totali
- Il cluster 9 con il 13.44 degli spostamenti totali

Sebbene siano molto simili in quanto occupano percentuali simili del dataset totale, i due cluster descrivono profili diversi sia nelle modalità di uso sia nelle caratteristiche spazio-temporali.

#### Differenza nei mezzi di trasporto

La distinzione più evidente riguarda il mezzo di trasporto utilizzato in prevalenza:

- Cluster 2 è dominato degli scooter elettrici (81.1%), con una presenza marginale di auto
- Cluster 9 è composto esclusivamente da automobili in car sharing.

Questa differenza riflette due approcci distinti alla mobilità urbana: nel primo caso, una scelta orientata all’agilità, alla rapidità e al costo contenuto; nel secondo, una preferenza per comfort, protezione e flessibilità, anche a fronte di una maggiore durata o distanza.

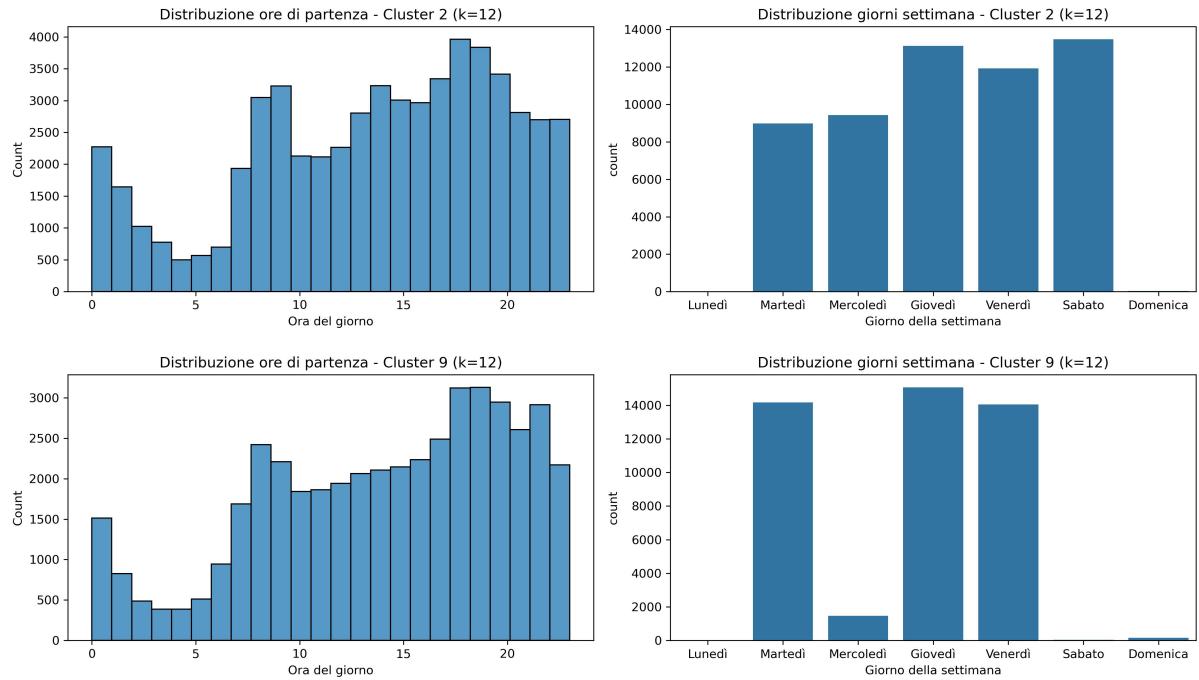


Figure 5.9: Distribuzione temporale per i cluster 2 e 9 ( $k = 12$ ).

### Distribuzione temporale

Sul piano temporale, entrambi i cluster si concentrano prevalentemente nei giorni feriali, ma con sfumature differenti:

- Cluster 2 mostra una leggera prevalenza il mercoledì e il venerdì, con una distribuzione oraria centrata attorno al primo pomeriggio.
- Cluster 9 si concentra invece su lunedì e venerdì, con una fascia oraria più ampia e frequente estensione oltre la mezzanotte.

La durata effettiva dei viaggi offre un altro spunto rilevante: mentre in Cluster 2 la durata stimata coincide quasi perfettamente con quella effettiva (circa 9 minuti), Cluster 9 mostra un forte scarto tra stima ed effettivo (durata media oltre 100 minuti), suggerendo un uso prolungato o flessibile del mezzo (vd. figura 5.9).

### Geografia degli spostamenti

Dal punto di vista spaziale:

- Gli spostamenti del Cluster 2 si mantengono brevi (circa 2.17 km), all'interno di aree centrali e semicentrali come Porta Venezia, Isola e Navigli, a testimonianza di una mobilità urbana intra-quartiere o tra poli vicini.
- Cluster 9, invece, pur con origini e destinazioni simili (zone centrali come Porta Romana o Brera), presenta distanze medie maggiori (oltre 5.5 km), che suggeriscono spostamenti inter-quartiere o su tratte più estese.

### Contesto funzionale

Entrambi i cluster si svolgono in aree ricche di POI (punti di interesse): uffici, servizi, ristoranti, istituzioni educative. Tuttavia, l'interpretazione funzionale dei due comportamenti è diversa:

- Cluster 2 riflette un profilo “dinamico e quotidiano”, probabilmente legato a studenti, giovani professionisti o lavoratori autonomi, che prediligono spostamenti brevi e frequenti.
- Cluster 9 appare invece più “strategico e flessibile”, utilizzato forse da lavoratori che necessitano di autonomia, famiglie, o utenti che affrontano tratte più impegnative o gestiscono attività distribuite

su più sedi.

### Conclusione

L'analisi comparativa tra il Cluster 2 e il Cluster 9 mette in evidenza come le differenze emerse in termini di tempi di utilizzo, durata effettiva, distanze percorse e tipologia di aree attraversate non siano casuali, bensì profondamente influenzate dal mezzo di trasporto prevalente all'interno di ciascun cluster.

In particolare:

- Il Cluster 2, dominato dagli scooter elettrici, mostra spostamenti tipicamente brevi e concentrati in aree centrali e densamente servite. Le durate effettive sono molto vicine a quelle stimate, le distanze ridotte e i tempi di utilizzo si concentrano nelle ore centrali della giornata. Questo comportamento riflette le caratteristiche intrinseche dello scooter: un mezzo agile, adatto a spostamenti rapidi su tratte brevi e per singoli utenti.
- Al contrario, il Cluster 9, costituito esclusivamente da automobili in car sharing, presenta spostamenti più lunghi, flessibili e variabili, sia in termini temporali che spaziali. La durata effettiva dei viaggi è significativamente superiore a quella stimata, con frequenti estensioni oltre la mezzanotte, e le distanze percorse risultano mediamente più che doppie rispetto a quelle del cluster scooter. Tali comportamenti sono coerenti con l'utilizzo dell'automobile, che consente maggiore autonomia, comfort, possibilità di trasporto di passeggeri o oggetti, e una minore dipendenza dalle condizioni ambientali.

Pertanto, si può concludere che le caratteristiche dei viaggi osservati in ciascun cluster siano fortemente modellate dal tipo di veicolo utilizzato, più che da variabili socio-temporali o geografiche esterne. È il mezzo stesso a determinare il perimetro di utilizzo, l'intensità temporale e la natura funzionale degli spostamenti, influenzando di conseguenza anche la distribuzione dei punti di interesse toccati.

## 5.7 Analisi per numero di cluster (14, 50, 100, 500, 1000, 2000)

L'analisi è stata condotta applicando l'algoritmo di clustering K-Means sull'insieme di dati già codificati (**encoded data**), ottenuti dalla fase di **training** avvenuta precedentemente. La base di dati è sempre la stessa, si sono variati i valori di  $k$  per riuscire a catturare al meglio più informazioni possibili sugli spostamenti:

- $k = 14$ : Il clustering con un numero ridotto di cluster produce una segmentazione molto simile a quella osservata tramite UMAP.
- $k = 50$ : Aumentando  $k$ , emergono pattern più specifici. La caratteristica principale che viene catturata è la distribuzione degli spostamenti in funzione del giorno della settimana. In particolare, si iniziano a distinguere chiaramente i cluster che rappresentano spostamenti tipici del weekend, dei giorni feriali e dei giorni festivi o di vacanza. Le caratteristiche temporali, quindi, diventano una delle principali direttive di separazione tra i gruppi.
- $k = 100$ : Con una segmentazione più fine, inizia a delinearsi una distinzione anche tra diverse fasce orarie della giornata. I cluster riflettono ora differenze tra spostamenti mattutini, pomeridiani, serali o notturni, suggerendo una strutturazione temporale più complessa nei comportamenti osservati.
- $k = 500$ : A questa soglia si comincia a intravedere una chiara suddivisione spaziale dei cluster. Le coordinate geografiche iniziano a incidere maggiormente sulla struttura dei gruppi, e si possono identificare almeno tre grandi categorie di cluster:
  1. Cluster distribuiti in modo sparso su tutto il territorio (spostamenti occasionali o non ricorrenti).
  2. Cluster concentrati all'interno della città di Milano.
  3. Cluster che rappresentano spostamenti periferici o fuori dai confini urbani.
- $k = 1000$ : Con una granularità ancora maggiore, si assiste a una raffinazione della suddivisione spaziale. I cluster iniziano a riflettere in modo più preciso la struttura urbana della città, differenziando tra:
  1. Spostamenti sparsi.

| Variabile              | Media | STD  | Min | 25% | Mediana | 75% | Max |
|------------------------|-------|------|-----|-----|---------|-----|-----|
| kmeans_labels_umap2D   | 2.0   | 0.02 | 1.0 | 2.0 | 2.0     | 2.0 | 8.0 |
| kmeans_labels_NoUmap2D | 2.0   | 0.02 | 1.0 | 2.0 | 2.0     | 2.0 | 8.0 |

Table 5.11: Corrispondenza del cluster 2 con e senza l'utilizzo di UMAP

| Variabile              | Media | STD | Min | 25% | Mediana | 75% | Max |
|------------------------|-------|-----|-----|-----|---------|-----|-----|
| kmeans_labels_umap2D   | 9.0   | 9.0 | 9.0 | 9.0 | 9.0     | 9.0 | 9.0 |
| kmeans_labels_NoUmap2D | 9.0   | 9.0 | 9.0 | 9.0 | 9.0     | 9.0 | 9.0 |

Table 5.12: Corrispondenza del cluster 9 con e senza l'utilizzo di UMAP

2. Spostamenti all'interno della città di Milano.
3. Spostamenti all'interno della circonvallazione (zona centrale).
4. Spostamenti nelle aree periferiche.
5. Spostamenti al di fuori dei confini cittadini, includendo i comuni e i paesi limitrofi.

- $k = 2000$  Con un numero così elevato di cluster, è possibile osservare una suddivisione ancora più dettagliata delle categorie di spostamento precedentemente individuate. In particolare, emergono le seguenti tipologie:
  1. Spostamenti distribuiti in modo sparso sul territorio.
  2. Spostamenti interni alla città di Milano.
  3. Spostamenti concentrati all'interno della circonvallazione, ovvero nella zona centrale della città.
  4. Spostamenti localizzati nelle aree periferiche di Milano.
  5. Spostamenti che si estendono oltre i confini cittadini, coinvolgendo i comuni e i paesi limitrofi.
  6. Spostamenti caratteristici delle aree a Sud, Sud-Est ed Est.
  7. Spostamenti nell'area Nord.
  8. Spostamenti nell'area Nord-Ovest.

## 5.8 Analisi con $k = 14$

I cluster ottenuti con  $k = 14$  mostrano una forte corrispondenza con quelli individuati durante l'analisi preliminare effettuata mediante la tecnica di riduzione dimensionale UMAP. Sebbene non vi sia una corrispondenza perfetta tra i due insiemi di cluster, la somiglianza è comunque significativa.

Questa discrepanza può essere attribuita principalmente al fatto che nell'analisi con UMAP era stato scelto un numero di cluster pari a  $k = 12$ . La differenza nel numero di cluster ha inevitabilmente influenzato la segmentazione dei dati, portando a una diversa granularità nella rappresentazione delle strutture latenti.

Le tabelle 5.11 e 5.12 mostrano la corrispondenza tra i cluster ottenuti con e senza l'uso di UMAP. Come si può osservare, i valori sono molto simili, a conferma che la riduzione dimensionale non ha modificato in modo significativo la struttura dei cluster.

## 5.9 Analisi con $k = 50$

Nel contesto della partizione a  $k = 50$ , emergono pattern più specifici nella struttura dei dati, con una forte separazione basata sulle caratteristiche temporali dei viaggi (vd. figura 5.13 e figura 5.14). In particolare, si distinguono cluster che riflettono comportamenti legati ai giorni feriali, al weekend e alle festività. In questo scenario, i cluster 38 e 39 rappresentano due casi peculiari che meritano un'analisi approfondita.

| Variabile             | Media | STD    | Min  | 25%  | Mediana | 75%   | Max     |
|-----------------------|-------|--------|------|------|---------|-------|---------|
| estimated_duration_mn | 9.20  | 5.07   | 0.0  | 5.0  | 8.0     | 12.0  | 50.0    |
| actual_duration_mn    | 33.14 | 119.57 | 3.95 | 15.0 | 20.02   | 25.30 | 3415.27 |
| start_hour            | 14.28 | 5.93   | 0.0  | 9.0  | 15.0    | 19.0  | 23.0    |
| start_minute          | 28.70 | 17.30  | 0.0  | 14.0 | 29.0    | 44.0  | 59.0    |
| start_day             | 7.96  | 4.35   | 1.0  | 5.0  | 8.0     | 12.0  | 16.0    |
| start_month           | 7.22  | 3.38   | 1.0  | 4.0  | 9.0     | 10.0  | 12.0    |
| end_hour              | 14.57 | 5.97   | 0.0  | 10.0 | 16.0    | 19.0  | 23.0    |
| end_minute            | 29.29 | 17.28  | 0.0  | 15.0 | 30.0    | 45.0  | 59.0    |
| end_day               | 7.97  | 4.35   | 1.0  | 5.0  | 8.0     | 12.0  | 16.0    |
| end_month             | 7.22  | 3.38   | 1.0  | 4.0  | 9.0     | 10.0  | 12.0    |
| weekday_num_start     | 3.0   | 0.0    | 3.0  | 3.0  | 3.0     | 3.0   | 3.0     |
| weekday_num_end       | 3.01  | 0.12   | 3.0  | 3.0  | 3.0     | 3.0   | 5.0     |
| start_monday          | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_tuesday         | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_wednesday       | 1.0   | 0.0    | 1.0  | 1.0  | 1.0     | 1.0   | 1.0     |
| start_thursday        | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_friday          | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_saturday        | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_sunday          | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_monday            | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_tuesday           | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_wednesday         | 0.99  | 0.09   | 1.0  | 1.0  | 1.0     | 1.0   | 1.0     |
| end_thursday          | 0.006 | 0.008  | 0.0  | 0.0  | 0.0     | 0.0   | 1.0     |
| end_friday            | 0.001 | 0.044  | 0.0  | 0.0  | 0.0     | 0.0   | 1.0     |
| end_saturday          | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_sunday            | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_weekend         | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_weekend           | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| start_public_holiday  | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| end_public_holiday    | 0.0   | 0.0    | 0.0  | 0.0  | 0.0     | 0.0   | 0.0     |
| holiday_start         | 0.024 | 0.16   | 0.0  | 0.0  | 0.0     | 0.0   | 1.0     |
| holiday_end           | 0.024 | 0.16   | 0.0  | 0.0  | 0.0     | 0.0   | 1.0     |

Table 5.13: Statistiche descrittive delle caratteristiche temporali del cluster 38

### 5.9.1 Introduzione

L'analisi dei cluster 38 e 39 mostra che aumentare il numero di cluster a 50 è stata una scelta utile, perché ha permesso di individuare comportamenti particolari che con meno cluster sarebbero stati mescolati insieme e difficili da riconoscere.

Per esempio, il cluster 38 evidenzia situazioni in cui il servizio viene usato in modo scorretto o problematico (come viaggi lasciati aperti o chiusi male), quindi può essere utile per capire dove intervenire e migliorare le regole di utilizzo. Il cluster 39, invece, raggruppa utenti che usano il servizio soprattutto per svago, magari nel weekend o durante il tempo libero.

Questa divisione non è solo interessante dal punto di vista dei dati, ma può aiutare anche a prendere decisioni pratiche, come pianificare interventi di manutenzione, fare campagne informative, o proporre incentivi per usare meglio il servizio.

Tutti questi aspetti vengono approfonditi e spiegati nell'analisi presentata di seguito.

### 5.9.2 Cluster 38: viaggi anomali durante i giorni feriali

Il cluster 38 è caratterizzato da viaggi che avvengono esclusivamente il mercoledì, come evidenziato dai valori costanti delle variabili `weekday_num_start` e `weekday_num_end` (entrambe pari a 3). Tuttavia, ciò che distingue maggiormente questo cluster è la discrepanza tra la durata stimata e quella reale del viaggio: la durata stimata (`estimated_duration_mn`) è di circa 9.2 minuti in media, mentre la durata effettiva (`actual_duration_mn`) è di ben 33.14 minuti, con una deviazione standard molto elevata (119.57) e un valore massimo estremo che supera le 56 ore.

Tali caratteristiche suggeriscono un comportamento anomalo, probabilmente riconducibile a una dimenticanza nella chiusura del noleggio da parte dell'utente. Considerando che in questi dati non sono presenti automobili, ma solo biciclette e motorini, è plausibile che i mezzi siano stati lasciati attivi o parcheggiati senza corretta chiusura del viaggio. La coincidenza tra giorno di inizio e fine rafforza questa ipotesi, indicando che il viaggio viene prolungato artificialmente nella registrazione del sistema.

| Variabile             | Media | STD   | Min  | 25%   | Mediana | 75%   | Max     |
|-----------------------|-------|-------|------|-------|---------|-------|---------|
| estimated_duration_mn | 9.24  | 6.42  | 0.0  | 5.0   | 9.0     | 13.0  | 47.0    |
| actual_duration_mn    | 26.81 | 70.65 | 4.30 | 14.85 | 15.08   | 29.31 | 2279.83 |
| start_hour            | 11.55 | 7.45  | 0.0  | 3.0   | 13.0    | 18.0  | 23.0    |
| start_minute          | 30.66 | 17.24 | 0.0  | 14.0  | 29.0    | 44.0  | 59.0    |
| start_day             | 7.76  | 3.73  | 1.0  | 5.0   | 9.0     | 11.0  | 15.0    |
| start_month           | 6.33  | 3.23  | 1.0  | 4.0   | 5.0     | 9.0   | 12.0    |
| end_hour              | 11.68 | 7.43  | 0.0  | 4.0   | 13.0    | 18.0  | 23.0    |
| end_minute            | 30.79 | 17.18 | 0.0  | 14.0  | 29.0    | 44.0  | 59.0    |
| end_day               | 7.78  | 3.71  | 1.0  | 5.0   | 9.0     | 11.0  | 15.0    |
| end_month             | 6.33  | 3.22  | 1.0  | 4.0   | 5.0     | 9.0   | 12.0    |
| weekday_num_start     | 6.99  | 0.11  | 6.0  | 7.0   | 7.0     | 7.0   | 7.0     |
| weekday_num_end       | 7.0   | 0.0   | 7.0  | 7.0   | 7.0     | 7.0   | 7.0     |
| start_monday          | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_tuesday         | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_wednesday       | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_thursday        | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_friday          | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| start_saturday        | 0.01  | 0.11  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| start_sunday          | 0.99  | 0.11  | 0.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| end_monday            | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_tuesday           | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_wednesday         | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_thursday          | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_friday            | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_saturday          | 0.0   | 0.0   | 0.0  | 0.0   | 0.0     | 0.0   | 0.0     |
| end_sunday            | 1.0   | 0.0   | 1.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| start_weekend         | 1.0   | 0.0   | 1.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| end_weekend           | 1.0   | 0.0   | 1.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| start_public_holiday  | 0.99  | 0.11  | 0.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| end_public_holiday    | 1.0   | 0.0   | 1.0  | 1.0   | 1.0     | 1.0   | 1.0     |
| holiday_start         | 0.25  | 0.43  | 0.0  | 0.0   | 0.0     | 0.0   | 1.0     |
| holiday_end           | 0.25  | 0.43  | 0.0  | 0.0   | 0.0     | 1.0   | 1.0     |

Table 5.14: Statistiche descrittive delle caratteristiche temporali del cluster 39

### 5.9.3 Cluster 39: spostamenti concentrati nei giorni festivi e weekend

Il cluster 39 rappresenta un comportamento diametralmente opposto rispetto al cluster 38. Qui, i viaggi avvengono quasi esclusivamente la domenica, in corrispondenza di giornate festive e di vacanza, come confermano le variabili `start_sunday`, `end_sunday`, `start_public_holiday` e `end_public_holiday`, tutte con media prossima o pari a 1.0.

Dal punto di vista temporale, gli spostamenti iniziano tipicamente in tarda mattinata (media dell'ora di inizio pari a 11.55) e si concludono nel primo pomeriggio. La durata effettiva è mediamente di 26.81 minuti, più lunga di quanto stimato (9.24 minuti), ma con una varianza decisamente inferiore rispetto al cluster 38, indicando una maggiore regolarità nei comportamenti.

Questo cluster riflette dunque una modalità d'uso legata al tempo libero, coerente con quanto ci si aspetta in un contesto urbano durante il weekend: spostamenti più lunghi della media (ma non anomali) e associati a giornate in cui non si lavora. L'associazione con festività (pubbliche e religiose) è particolarmente forte, a testimonianza di una fruizione del servizio orientata al tempo libero o a eventi straordinari

### 5.9.4 Confronto e interpretazione

I due cluster si collocano agli estremi di un asse interpretativo che va dall'anomalia tecnica (cluster 38 5.13) all'uso tipico in giornate speciali (cluster 39 5.14)(vd. figura 5.10).

## 5.10 Analisi con $k = 100$

Aumentando il numero di cluster ( $k$ ) all'interno di K-means, (vd. figura 5.15 e figura 5.11), possiamo notare un maggior raggruppamento di dati simili. Anche in questa configurazione, Con il valore di  $k$  impostato a 100, le caratteristiche più considerate sono ancora quelle temporali, ma, a differenza di quanto osservato con  $k = 50$ , emerge una maggiore attenzione all'orario. Questo conferma che il fattore temporale è uno dei più rilevanti nella mobilità urbana in sharing.

Tra tutti i cluster ottenuti, il Cluster 20 risulta particolarmente interessante per il suo comportamento molto specifico e coerente. Gli spostamenti che rientrano in questo gruppo avvengono esclusivamente la domenica o durante le festività, sempre nelle prime ore del mattino: tra l'0 : 00 e le 4 : 00, con un picco

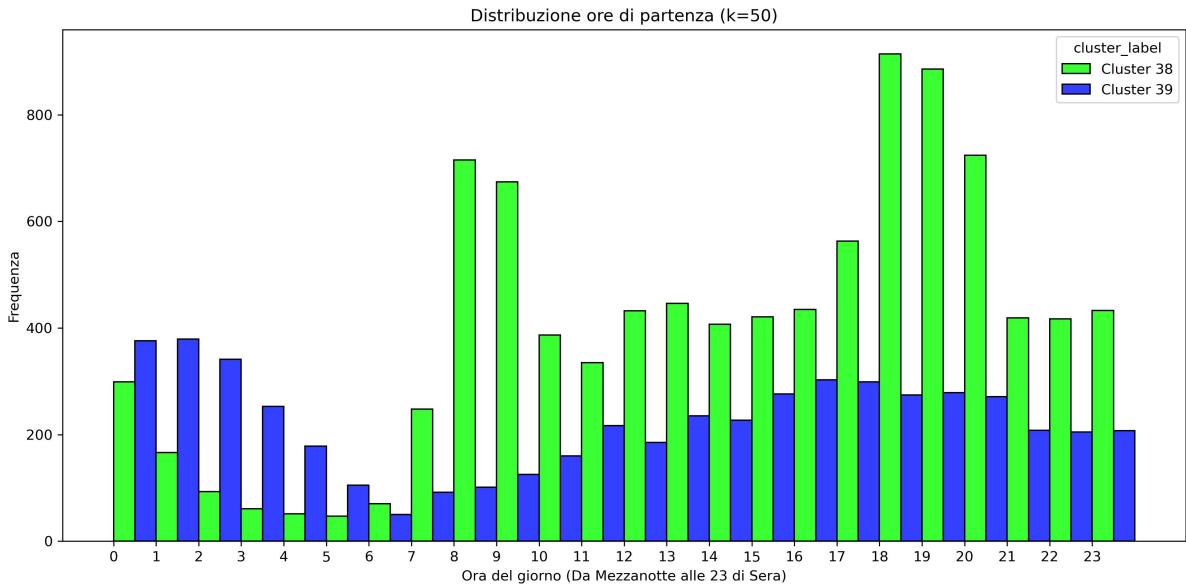


Figure 5.10: Distribuzione temporale per i cluster 38 e 39 ( $k = 50$ ).

tra le 0 : 00 e le 2 : 00. Non ci sono differenze significative tra stagioni o mesi, il che suggerisce che si tratta di un comportamento costante durante tutto l'anno.

Gli utenti di questo cluster utilizzano i mezzi in modo piuttosto diverso dalla media: fanno viaggi notturni con percorrenze brevi, ma durate molto più lunghe. In alcuni casi, il veicolo viene lasciato fermo per ore o riutilizzato prima di terminare il viaggio, con outlier che arrivano anche a durare oltre 13 ore. Questo tipo di utilizzo fa pensare a rientri da serate o eventi notturni, dove il mezzo viene parcheggiato per lungo tempo o usato in modo frazionato.

In sintesi, il Cluster 20 evidenzia un tipo di mobilità legato al tempo libero notturno del fine settimana, che non era visibile con configurazioni a minor granularità. Questo dimostra che aumentare il numero di cluster può aiutare a individuare comportamenti specifici e meno frequenti, utili per capire meglio l'uso reale del servizio e migliorare la sicurezza, la gestione della flotta e l'organizzazione nelle ore notturne e nei giorni festivi.

### 5.10.1 Cluster 20: distribuzione settimanale e festiva

Tutti gli spostamenti associati a questo cluster avvengono di domenica, come indicato dalla variabile `weekday_num_start` con media pari a 7.0 e dalla variabile `start_sunday` pari a 1.0. Tutte le altre variabili legate ai giorni della settimana risultano nulle. Anche `start_weekend` e `end_weekend` assumono valore 1.0, confermando che il cluster è esclusivamente associato al fine settimana. Inoltre, la totalità degli spostamenti ricade in giornate festive, come evidenziato da `start_public_holiday` e `end_public_holiday` (entrambi pari a 1.0), suggerendo una forte sovrapposizione tra la domenica e le festività pubbliche.

### 5.10.2 Cluster 20: distribuzione mensile e giornaliera

Dal punto di vista stagionale, gli spostamenti sono distribuiti lungo tutto l'anno in maniera equilibrata: la mediana del mese è pari a 6 (giugno), con un intervallo che si estende da gennaio (minimo 1) a dicembre (massimo 12). Anche la distribuzione giornaliera (variabili `start_day` e `end_day`) mostra valori medi e mediani coerenti (mediana pari a 14.0), segnalando una certa regolarità nella collocazione mensile e giornaliera dei viaggi.

### 5.10.3 Cluster 20: fascia oraria

Gli orari di inizio e fine degli spostamenti si concentrano prevalentemente durante la notte e nelle prime ore del mattino (vd. figura 5.11). L'orario medio di partenza è 2.42, con una mediana pari a 2.0, mentre l'orario medio di arrivo è leggermente più alto (2.83), ma con la stessa mediana. L'intervallo interquartile

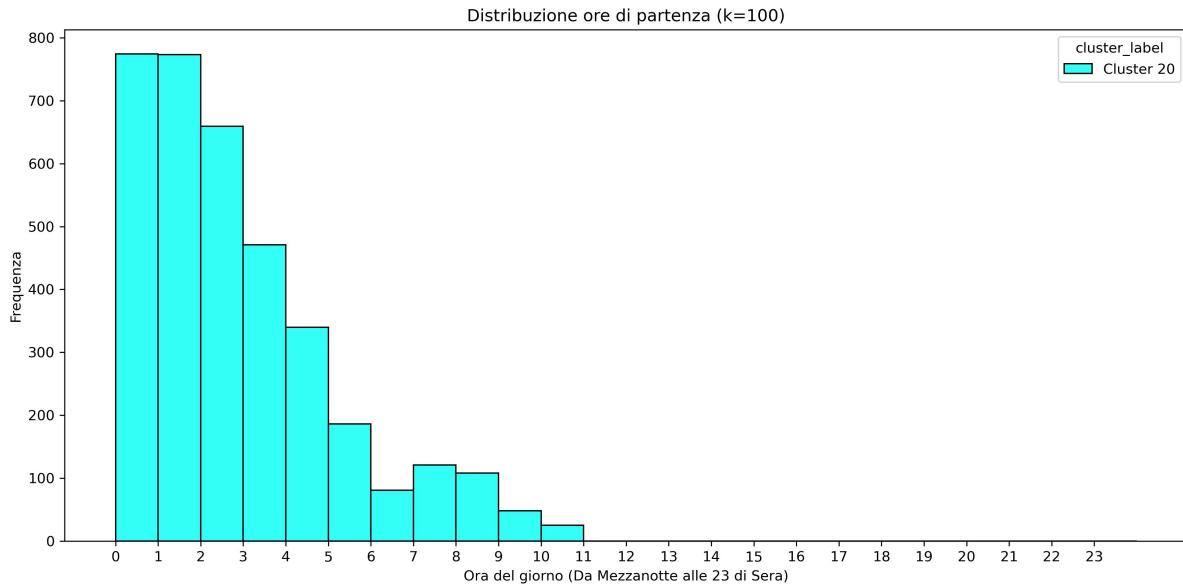


Figure 5.11: Distribuzione delle ore di partenza per il cluster 20 ( $k = 100$ ). La maggior parte degli spostamenti avviene tra mezzanotte e le prime ore del mattino, con un picco tra le 0 : 00 e le 2 : 00.

degli orari di inizio va da 1 : 00 a 4 : 00, con valori massimi che possono raggiungere le 10 : 00 per le partenze e le 14 : 00 per gli arrivi. I minuti risultano distribuiti in modo uniforme, con mediane pari a 27 per l'inizio e 29 per la fine dello spostamento, indicando una finestra temporale ben definita e concentrata nelle ore notturne.

#### 5.10.4 Cluster 20: durata degli spostamenti

La durata stimata degli spostamenti è breve, con una media di 10.28 minuti e una mediana di 9.0. Tuttavia, la durata effettiva è significativamente più elevata, con una media di 25.36 minuti e una mediana di 15.15 minuti. La deviazione standard (42.43 minuti) e il valore massimo (809.98 minuti, circa 13.5 ore) indicano la presenza di outlier o utilizzi anomali del mezzo, probabilmente dovuti a lunghe soste o a viaggi frammentati.

#### 5.10.5 Cluster 20: interpretazione complessiva

Il cluster 20 rappresenta un comportamento di mobilità chiaramente ritagliato sulla domenica e sulle festività pubbliche, con spostamenti che avvengono prevalentemente nelle prime ore del giorno. Questo cluster riflette viaggi legati al tempo libero notturno, al rientro da attività ricreative o a eventi occasionali svolti nella notte tra sabato e domenica. L'elevata differenza tra durata stimata e reale suggerisce anche la possibilità di lunghi parcheggi o soste tra l'inizio e la fine del viaggio.

### 5.11 Analisi con $k = 500$

Per analizzare i comportamenti degli utenti della mobilità condivisa a Milano, abbiamo usato l'algoritmo K-means con  $k = 500$ . Questo ci ha permesso di dividere tutti gli spostamenti in gruppi (cluster) simili tra loro, in base a quando avvengono, dove si svolgono e che mezzo viene usato. In particolare, abbiamo scelto di concentrarci su tre cluster: il numero 50, il 56 e il 237. Anche se diversi per quantità di spostamenti, questi cluster sono interessanti perché rappresentano tre modi molto diversi di muoversi in città.

- Il cluster 50 riguarda spostamenti brevi fatti in bicicletta nelle prime ore del lunedì mattina, soprattutto tra le 3 e le 10. Gli utenti partono da zone residenziali e arrivano vicino a stazioni ferroviarie o fermate della metropolitana, come Lambrate, Porta Garibaldi o Romolo. Questo comportamento ci fa pensare a persone che vanno al lavoro o a scuola e che usano la bici come primo mezzo, per poi prendere i mezzi pubblici. È quindi un esempio chiaro di mobilità intermodale, molto utile per capire come le biciclette in sharing si integrano col trasporto pubblico.

| Variabile             | Media | STD   | Min | 25%   | Mediana | 75%   | Max    |
|-----------------------|-------|-------|-----|-------|---------|-------|--------|
| estimated.duration_mn | 10.28 | 6.35  | 0.0 | 6.0   | 9.0     | 14.0  | 47.0   |
| actual.duration_mn    | 25.36 | 42.43 | 4.6 | 14.85 | 15.15   | 29.86 | 809.98 |
| start.hour            | 2.42  | 2.29  | 0.0 | 1.0   | 2.0     | 4.0   | 10.0   |
| start.minute          | 29.80 | 17.30 | 0.0 | 12.0  | 27.0    | 44.0  | 59.0   |
| start.day             | 14.73 | 8.57  | 1.0 | 9.0   | 14.0    | 22.0  | 30.0   |
| start.month           | 6.58  | 3.29  | 1.0 | 4.0   | 6.0     | 10.0  | 12.0   |
| end.hour              | 2.83  | 2.38  | 0.0 | 1.0   | 2.0     | 4.0   | 14.0   |
| end.minute            | 30.40 | 17.02 | 0.0 | 15.0  | 29.0    | 44.0  | 59.0   |
| end.day               | 14.73 | 8.57  | 1.0 | 9.0   | 14.0    | 22.0  | 30.0   |
| end.month             | 6.58  | 3.28  | 1.0 | 4.0   | 6.0     | 10.0  | 12.0   |
| weekday_num_start     | 7.0   | 0.0   | 7.0 | 7.0   | 7.0     | 7.0   | 7.0    |
| weekday_num_end       | 7.0   | 0.0   | 7.0 | 7.0   | 7.0     | 7.0   | 7.0    |
| start.monday          | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.tuesday         | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.wednesday       | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.thursday        | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.friday          | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.saturday        | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| start.sunday          | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| end.monday            | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.tuesday           | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.wednesday         | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.thursday          | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.friday            | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.saturday          | 0.0   | 0.0   | 0.0 | 0.0   | 0.0     | 0.0   | 0.0    |
| end.sunday            | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| start.weekend         | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| end.weekend           | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| start.public_holiday  | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| end.public_holiday    | 1.0   | 0.0   | 1.0 | 1.0   | 1.0     | 1.0   | 1.0    |
| holiday_start         | 0.12  | 0.33  | 0.0 | 0.0   | 0.0     | 0.0   | 1.0    |
| holiday_end           | 0.12  | 0.33  | 0.0 | 0.0   | 0.0     | 0.0   | 1.0    |

Table 5.15: Statistiche descrittive delle caratteristiche temporali del cluster 20

- Il cluster 237 è molto più piccolo (solo 71 spostamenti), ma interessante perché mostra un comportamento diverso: gli spostamenti avvengono la domenica sera e notte, con qualche coda al lunedì mattina. Sono fatti tutti in auto in sharing, partendo e arrivando fuori dal centro città, in zone come Segrate, Pioltello o vicino alla stazione Cascina Gobba. Questo tipo di mobilità può indicare rientri da attività ricreative o lavorative, in orari in cui i mezzi pubblici sono meno disponibili. Anche se raro, è importante perché rappresenta spostamenti extraurbani o periferici che spesso non vengono considerati nelle analisi classiche.
- Il cluster 56 comprende invece spostamenti urbani brevi fatti con scooter in sharing nel pomeriggio e nella prima serata del lunedì, tra le 15 : 10 e le 20 : 30. Le zone coinvolte sono centrali, come Porta Venezia, Buenos Aires e la Stazione Centrale. Questo comportamento è probabilmente legato al rientro dal lavoro o a brevi tragitti verso negozi, servizi o appuntamenti serali. Il cluster è interessante soprattutto per la fascia oraria che rappresenta: un momento della giornata molto attivo, in cui lo scooter viene scelto per la sua rapidità e comodità nel traffico.

### 5.11.1 Cluster 50: spostamenti pendolari mattutini con bicicletta

Il cluster 50 (vd. tabella 5.16) comprende 542 spostamenti, concentrati nelle prime ore del lunedì mattina, in particolare nella fascia oraria compresa tra le 3 : 00 e le 10 : 00. Gli utenti percorrono brevi distanze, generalmente inferiori ai 2 km, utilizzando biciclette in sharing. Le destinazioni mostrano una notevole regolarità spaziale, tendendo a concentrarsi presso stazioni ferroviarie, fermate della metropolitana o altri nodi di interscambio modale, come ad esempio le aree intorno a Lambrate, Porta Garibaldi o Romolo.

La regolarità temporale e spaziale di questi spostamenti suggerisce una mobilità pendolare altamente strutturata, finalizzata al raggiungimento di mezzi pubblici per la prosecuzione del tragitto. In tal senso, la bicicletta funge da primo segmento di un percorso intermodale. Questo comportamento è coerente con una crescente integrazione tra mobilità dolce e trasporto pubblico, in linea con le politiche urbane di sostenibilità.

### 5.11.2 Cluster 237: mobilità extraurbana o periferica domenicale

Il cluster 237 (vd. tabella 5.17) raccoglie un numero ridotto di spostamenti (71 su un totale di 334.353), con una percentuale trascurabile in termini quantitativi, ma di particolare interesse qualitativo. Gli

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 2263.27   | 1579.18  | 116.0     | 1086.25   | 1970.0    | 3216.0    | 8124.0    |
| latitude_start       | 45.478327 | 0.015482 | 45.433790 | 45.459013 | 45.470495 | 45.481725 | 45.509430 |
| longitude_start      | 9.190285  | 0.024844 | 9.126110  | 9.171015  | 9.190295  | 9.208702  | 9.242780  |
| latitude_end         | 45.469257 | 0.015451 | 45.433660 | 45.458137 | 45.467875 | 45.481217 | 45.507500 |
| longitude_end        | 9.190068  | 0.023304 | 9.127520  | 9.174397  | 9.190425  | 9.206330  | 9.245440  |

Table 5.16: Caratteristiche geospaziali del cluster 50

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 8398.20   | 4604.64  | 478.0     | 5177.50   | 8213.0    | 11831.0   | 17682.0   |
| latitude_start       | 45.477272 | 0.039844 | 45.358890 | 45.454750 | 45.476910 | 45.512205 | 45.534030 |
| longitude_start      | 9.188983  | 0.060990 | 9.086980  | 9.130935  | 9.193020  | 9.231780  | 9.290390  |
| latitude_end         | 45.473647 | 0.028640 | 45.417470 | 45.451590 | 45.475890 | 45.495430 | 45.529410 |
| longitude_end        | 9.188006  | 0.041527 | 9.084860  | 9.157075  | 9.191940  | 9.225700  | 9.264780  |

Table 5.17: Caratteristiche geospaziali del cluster 237

spostamenti si verificano quasi esclusivamente nella fascia serale della domenica, dalle ore 17 circa, e notturna della domenica, alle ore 1, con qualche viaggio compreso nella mattinata del lunedì seguente, con una durata complessiva spesso molto prolungata. Le origini e le destinazioni si collocano al di fuori del centro urbano, in particolare nella zona est della città e oltre i confini comunali, in aree come Segrate, Pioltello o l'intorno della stazione di Cascina Gobba.

Tutti gli spostamenti in questo cluster avvengono mediante auto in sharing, un mezzo che garantisce autonomia, comfort e flessibilità in orari in cui l'offerta di trasporto pubblico è ridotta. Le distanze percorse sono generalmente comprese tra gli 5 e i 12 km, con una durata effettiva del noleggio spesso superiore a quella strettamente necessaria per il tragitto, probabilmente a causa della congestione del traffico urbano. Congestione plausibile considerando la fascia oraria di utilizzo tipica del rientro alla propria dimora dopo una giornata di lavoro. Questo pattern è interpretabile come rappresentativo di una mobilità ricreativa o sociale, presumibilmente effettuata da giovani, o di viaggi di ritorno post giornate lavorative.

### 5.11.3 Cluster 56: mobilità urbana serale con scooter

Il cluster 56 (vd. tabella 5.18) comprende 774 spostamenti, distribuiti con regolarità durante il pomeriggio e la prima serata del lunedì, in particolare tra le 15 : 10 e le 20 : 30. Gli spostamenti avvengono esclusivamente mediante scooter in sharing e coprono distanze brevi, generalmente inferiori ai 2 km. Le aree coinvolte si concentrano nella zona centro-orientale della città, tra Porta Venezia, Buenos Aires e la Stazione Centrale, ovvero in un tessuto urbano denso, caratterizzato dalla compresenza di attività lavorative, commerciali e residenziali.

Il profilo temporale e spaziale suggerisce che tali spostamenti corrispondano alla fase di rientro dal lavoro o a brevi tragitti verso servizi e attività serali. L'utilizzo dello scooter in questa fascia oraria e in tale contesto urbano riflette l'esigenza di rapidità, agilità nel traffico e possibilità di parcheggio agevolato, tipiche degli utenti che operano in contesti ad alta densità e con tempistiche rigide.

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 2057.41   | 1340.57  | 129.0     | 1147.25   | 1717.0    | 2602.50   | 10452.0   |
| latitude_start       | 45.469208 | 0.015225 | 45.435800 | 45.458320 | 45.469035 | 45.479842 | 45.524620 |
| longitude_start      | 9.192548  | 0.024201 | 9.084430  | 9.176783  | 9.194785  | 9.210897  | 9.244680  |
| latitude_end         | 45.469086 | 0.015541 | 45.435590 | 45.456553 | 45.46990  | 45.481025 | 45.515780 |
| longitude_end        | 9.193769  | 0.024415 | 9.120430  | 9.176707  | 9.195495  | 9.213082  | 9.245020  |

Table 5.18: Caratteristiche geospaziali del cluster 56

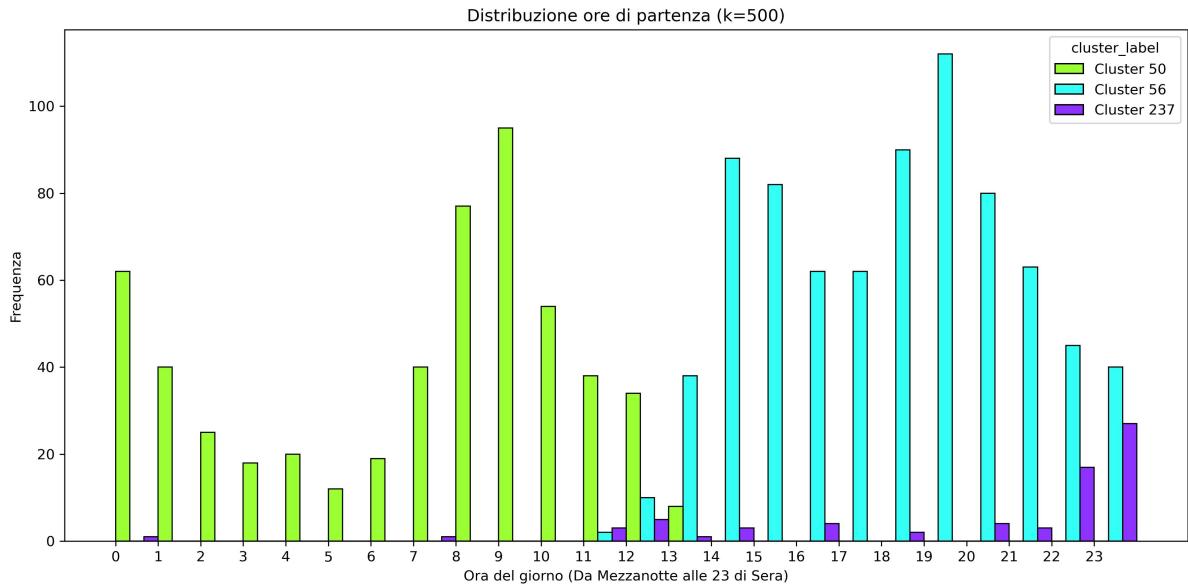


Figure 5.12: Distribuzione temporale per i cluster 50, 56 e 237 ( $k = 500$ ).

#### 5.11.4 Confronto tra i cluster

I tre cluster analizzati evidenziano differenti logiche di mobilità urbana, distinguibili secondo variabili temporali, spaziali e modali. Il cluster 237 si differenzia per l'adozione dell'auto in contesto extraurbano e in orari serali, rappresentando una mobilità meno frequente ma significativa in termini di autonomia e assenza di alternative. Il cluster 50, al contrario, evidenzia una mobilità regolare e intermodale, associata all'inizio della settimana lavorativa e alla necessità di integrazione con il trasporto pubblico. Il cluster 56 infine rappresenta una mobilità urbana interna, flessibile e ad alta frequenza, compatibile con esigenze di spostamento serali e brevi (vd. figura 5.12).

La segmentazione ottenuta attraverso il clustering consente di individuare con precisione comportamenti ricorrenti, facilitando la lettura della domanda di mobilità in funzione del contesto urbano, delle infrastrutture disponibili e degli stili di vita (vd. figura 5.13). Questa tipologia di analisi rappresenta un utile supporto per la pianificazione urbana e per l'ottimizzazione dei servizi di mobilità condivisa, in un'ottica di efficienza, equità e sostenibilità.

## 5.12 Analisi con $k = 1000$

In questa sezione analizziamo la mobilità urbana con un numero molto alto di cluster ( $k = 1000$ ), per osservare con più dettaglio i comportamenti locali e le differenze tra zone specifiche della città. Con così tanti gruppi, emergono particolarità interessanti legate a quartieri, orari e mezzi utilizzati.

Come esempio, abbiamo scelto due cluster molto diversi tra loro:

- Il cluster 810, con spostamenti in scooter nella zona interna alla circonvallazione, usati in modo anomalo per periodi molto lunghi, spesso legati a uscite serali o eventi.
- Il cluster 328, con spostamenti in auto nei comuni periferici, effettuati di giorno e spesso prolungati, probabilmente per motivi pratici o lavorativi.

Questi casi aiutano a capire quanto possano cambiare gli usi della mobilità condivisa a seconda del luogo, dell'orario e del mezzo scelto.

**Analisi comparativa dei cluster 810 e 328** L'analisi dei cluster 810 e 328, emersi dal processo di clustering con  $k = 1000$ , consente di confrontare due dinamiche di mobilità urbana profondamente differenti: da un lato, gli spostamenti nella zona della circonvallazione milanese (cluster 810 vd. tabella 5.19), e dall'altro, la mobilità che ha origine o destinazione nei comuni periferici e paesi limitrofi (cluster 328 vd. tabella 5.20). Sebbene entrambi i cluster rappresentino una quota marginale del dataset complessivo

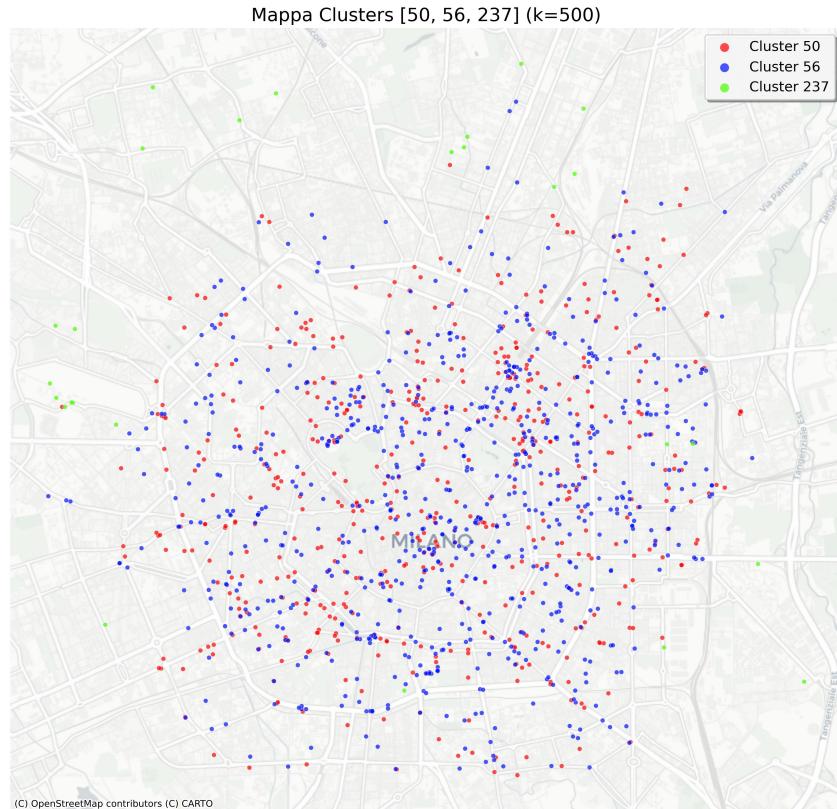


Figure 5.13: Mappa geografica dei cluster 50, 56 e 237 ( $k = 500$ ).

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 4830.50   | 2950.45  | 1695.0    | 2676.0    | 3956.50   | 5611.75   | 12254.0   |
| latitude_start       | 45.473860 | 0.013786 | 45.449190 | 45.465827 | 45.475210 | 45.483665 | 45.502670 |
| longitude_start      | 9.181929  | 0.028130 | 9.124450  | 9.162342  | 9.181670  | 9.198602  | 9.238480  |
| latitude_end         | 45.473223 | 0.016689 | 45.447990 | 45.461132 | 45.475430 | 45.482510 | 45.507490 |
| longitude_end        | 9.197398  | 0.024796 | 9.156180  | 9.179623  | 9.190530  | 9.210358  | 9.245550  |

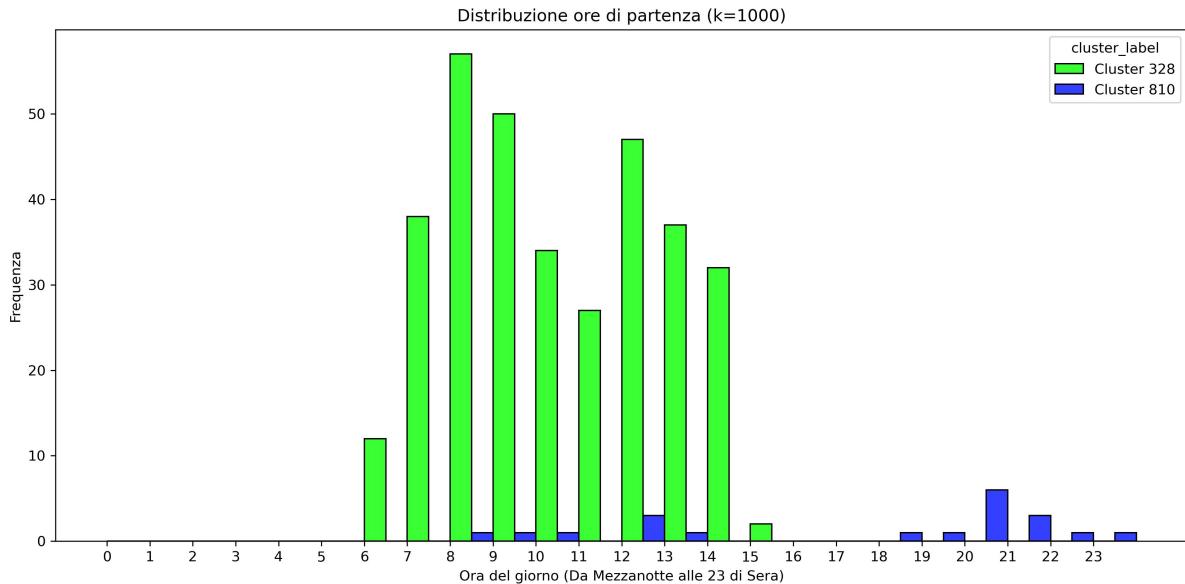
Table 5.19: Caratteristiche geospaziali del cluster 810

(rispettivamente 0.006% e 0.10% dei 334.353 spostamenti totali), la loro analisi rivela pattern significativi legati a profili d'uso, scelte modali e strutture urbane differenti.

### 5.12.1 Cluster 810

Il cluster 810 include 20 spostamenti tutti effettuati con scooter in sharing, e localizzati prevalentemente nella zona nord-orientale della città, nei quartieri interni alla circonvallazione come Lambrate, Ortica, Città Studi e Porta Venezia. Le caratteristiche temporali sono particolarmente anomale: tutti i viaggi iniziano il martedì pomeriggio/sera (media ore: 17 : 22, con una mediana intorno alle 20 : 00) e terminano il giovedì mattina successiva (media ore: 07 : 35, con una mediana intorno alle 4 : 00), con una durata effettiva media di circa 38 ore. A fronte di una durata stimata breve (12.75 minuti).

Questo gruppo usa soltanto lo scooter sharing: non compaiono altri mezzi. Però spesso il veicolo rimane occupato a lungo invece di servire per veri spostamenti. Le partenze e gli arrivi si concentrano vicino a ristoranti e locali, quindi sembrano legati a uscite serali o notturne, eventi o vita universitaria. In zona ci sono molti mezzi di superficie, ma poche fermate di metro o treni: forse per questo si preferisce lo scooter, garantendo quindi più indipendenza dagli orari lavorativi dei mezzi pubblici.

Figure 5.14: Distribuzione temporale per i cluster 328 e 810 ( $k = 1000$ ).

| Variabile            | Media     | STD      | Min       | 25%       | Mediana   | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 4915.32   | 2976.22  | 215.0     | 2631.50   | 4363.0    | 6743.50   | 14028.0   |
| latitude_start       | 45.472568 | 0.021043 | 45.418820 | 45.454495 | 45.470850 | 45.490713 | 45.528480 |
| longitude_start      | 9.180165  | 0.032029 | 9.084810  | 9.155788  | 9.181830  | 9.206315  | 9.256670  |
| latitude_end         | 45.474227 | 0.019571 | 45.412520 | 45.460652 | 45.475700 | 45.487200 | 45.519290 |
| longitude_end        | 9.184645  | 0.030213 | 9.083600  | 9.162983  | 9.188740  | 9.207693  | 9.251840  |

Table 5.20: Caratteristiche geospatiali del cluster 328

### 5.12.2 Cluster 328

Il cluster 328 raccoglie 336 spostamenti effettuati esclusivamente con auto in sharing, concentrati in modo sistematico il giovedì mattina. L'orario medio di partenza è attorno alle 10 : 28, con un arrivo medio verso le 11 : 28, ma la durata reale media del noleggio supera i 60 minuti di poco arrivando anche all'ora e 10 minuti, con una varianza molto elevata che indica la presenza di numerosi viaggi estesi anche ben oltre la soglia dell'ora singola arrivando anche alla decina di ore. La varianza dell'orario di inizio e di fine corrisponde all'effettivo alla durata attuale del noleggio, probabilmente dovuto a viaggi prolungati causa congestione del traffico o assenza di utilizzo prolungata del mezzo senza effettivamente chiudere lo *sharing* del veicolo.

Dal punto di vista spaziale, i percorsi si distribuiscono prevalentemente in aree periferiche e nei comuni limitrofi a Milano, con una scarsa presenza di stazioni della metropolitana. Tuttavia, si osserva una discreta densità di punti di interesse (POI) legati al commercio e ai servizi, come attività economiche, servizi alla persona e ristorazione. La distanza media percorsa, pari a circa 4,9 km, risulta contenuta, ma la durata complessiva degli spostamenti è relativamente elevata. Questo può suggerire un uso intermittente del veicolo, potenzialmente legato a commissioni multiple nell'arco della giornata, a esigenze lavorative flessibili o a spostamenti in zone scarsamente servite dal trasporto pubblico.

#### Conclusione comparativa

L'analisi comparativa dei cluster 810 e 328 evidenzia come la mobilità in sharing, pur all'interno di una stessa città, possa assumere forme radicalmente diverse a seconda del **contesto spaziale, del mezzo utilizzato e dell'orario** (vd. figura 5.14). Entrambi i cluster condividono un'anomalia temporale (uso prolungato del mezzo rispetto alla durata stimata), ma divergenze marcate emergono su più dimensioni:

- **Contesto urbano:** il cluster 810 si trova in zone centrali, all'interno della circonvallazione, dove la vita sociale serale è molto attiva; il cluster 328, invece, si concentra vicino ai confini della città,

in aree con funzioni più legate al lavoro o ad attività direzionali.

- **Mezzo scelto:** scooter per il primo, auto per il secondo. Ciò riflette non solo preferenze individuali ma anche disponibilità, topografia urbana e natura dell'attività svolta.
- **Fascia oraria:** serale/notturna per il cluster 810, mattutina e feriale per il cluster 328, suggerendo scopi d'uso differenti (tempo libero contro attività lavorative o personali).
- **Efficienza e criticità:** in entrambi i casi si osservano pratiche d'uso che riducono l'efficienza del servizio (veicoli occupati oltre il tempo utile), indicando la necessità di strategie di monitoraggio e regolazione dell'uso da parte dei provider.

In sintesi, questi due cluster rappresentano due polarità dell'ecosistema della mobilità condivisa a Milano: da un lato, una mobilità occasionale e concentrata nelle aree centrali con forte vocazione al tempo libero (cluster 810); dall'altro, una mobilità organizzata, orientata alla funzione, che privilegia comodità e autonomia nelle aree periferiche o direzionali (cluster 328). Questa diversità suggerisce l'importanza di approcci mirati nella pianificazione urbana e nei modelli di tariffazione e incentivazione dei servizi di sharing.

## 5.13 Analisi con $k = 2000$

Utilizzando un numero molto alto di cluster ( $k = 2000$ ), è stato possibile individuare tanti profili diversi di mobilità urbana. I dati mostrano come gli spostamenti cambino in base alla durata, all'orario, al mezzo di trasporto usato e alla zona della città coinvolta.

Per approfondire meglio questi risultati, sono stati scelti due cluster rappresentativi: il cluster 231 e il cluster 511. La scelta si basa principalmente sulle zone geografiche interessate e sulle differenze nei comportamenti osservati.

- **Cluster 231:** riguarda spostamenti nella zona Sud, Sud-Est ed Est periferica di Milano (come San Donato Milanese, Assago e Segrate). Gli spostamenti avvengono solo il sabato sera, tra le 18 : 00 e le 21 : 00, e vengono fatti quasi esclusivamente in scooter in sharing. Si tratta di viaggi brevi, con una durata effettiva tra i 5 e i 7 minuti, che partono da aree poco ricche di punti di interesse.
- **Cluster 511:** riguarda spostamenti tra le zone Nord, Ovest e Nord-Ovest della città (come Quinto Romano, Quarto Oggiaro e Gallaratese). Questi viaggi avvengono durante la settimana, soprattutto il mercoledì, e vengono fatti in automobile. Le durate sono molto più lunghe (anche oltre un'ora) e spesso collegano zone periferiche lontane tra loro.

In sintesi, questi due cluster rappresentano due modi diversi di muoversi a Milano: uno legato al tempo libero e al weekend, l'altro legato ad attività quotidiane o lavorative. L'analisi dettagliata di ciascun cluster viene presentata nelle prossime sezioni.

### 5.13.1 Cluster 231

Il cluster 231 raccoglie spostamenti che avvengono **esclusivamente durante la giornata di sabato**, concentrandosi principalmente nella fascia oraria compresa tra le 18 : 00 e le 21 : 00 (tra il primo e il terzo quartile).

Gli spostamenti sono medio-brevi, con una durata contenuta sia nella stima iniziale sia nella misurazione effettiva. Nello specifico:

- **Durata stimata:** da 2 a 5 minuti (minimo assoluto pari a 1 minuto, massimo pari a 23 minuti);
- **Durata effettiva:** da 5 a 7 minuti (minimo assoluto pari a 4 minuti, massimo pari a 30 minuti);
- **Distanza stimata:** principalmente inferiore ai 2 km, con un massimo registrato di circa 9 km.

Il mezzo di trasporto impiegato è esclusivamente il motorscooter in sharing, a conferma di una preferenza per modalità agili e veloci nel traffico urbano. L'assenza di altri mezzi come auto o biciclette suggerisce che lo scooter sia percepito come il mezzo più efficiente per questo tipo di spostamenti rapidi in contesti metropolitani densi.

Dal punto di vista spaziale, le aree attraversate presentano inizialmente una scarsa densità di punti di interesse (POI):

| Variabile            | Media     | STD      | Min       | 25%       | Median    | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 2182.80   | 2627.08  | 182.0     | 837.50    | 1182.0    | 2013.50   | 8891.0    |
| latitude_start       | 45.418807 | 0.033450 | 45.393450 | 45.396555 | 45.404200 | 45.416570 | 45.490630 |
| longitude_start      | 9.201318  | 0.058210 | 9.147010  | 9.164615  | 9.164980  | 9.264840  | 9.298530  |
| latitude_end         | 45.416441 | 0.034687 | 45.393450 | 45.395275 | 45.396500 | 45.429130 | 45.495500 |
| longitude_end        | 9.192137  | 0.052141 | 9.148870  | 9.162190  | 9.164820  | 9.202805  | 9.296110  |

Table 5.21: Caratteristiche geospatiali del cluster 231

- **Numero massimo di POI in partenza:** 12;
- **Terzo quartile:** 3;
- La maggioranza dei viaggi parte da aree prive di POI rilevanti (valore pari a 0).

I pochi POI presenti all'inizio appartengono a categorie come **turismo e intrattenimento, ristorazione e bar, servizi vari e business**, con una totale assenza di ferrovie e metropolitana, mentre si registrano alcune presenze limitate di linee di superficie.

In prossimità del punto di arrivo, invece, la situazione cambia significativamente:

- **Ferrovie e la metropolitana:** Essi restano assenti, ma aumenta lievemente la presenza di linee di superficie (numero massimo da 4 a 7);
- Il numero complessivo di POI cresce considerevolmente: da valori iniziali nulli fino a un massimo di 486;
- I POI maggiormente rappresentati nella destinazione appartengono alle categorie **turismo, business, ristorazione e bar, sport e cura personale, servizi vari e istituzioni ed educazione**. Tuttavia, solo nei casi di massima concentrazione i valori risultano attivi (1).

Alcune zone coinvolte dagli spostamenti di questo cluster sono: **San Donato, Corvetto, Assago, Rozzano e Segrate** (vd. tabella 5.21).

In sintesi, il cluster 231 rappresenta spostamenti brevi e rapidi effettuati il sabato sera in scooter in sharing, da zone periferiche poco servite e prive di attrazioni verso zone più ricche di POI, in particolare legati a svago, ristorazione e servizi. Questo suggerisce un utilizzo del servizio legato a motivazioni ludico-sociali in contesto serale-weekend.

### 5.13.2 Cluster 511

Il cluster 511 raccoglie spostamenti che si concentrano principalmente durante i giorni feriali, registrati il mercoledì, e una componente marginale in corrispondenza di giorni festivi (valori massimi pari a 1 per **holiday\_start** e **holiday\_end**).

La finestra temporale degli spostamenti è ampia ma contenuta all'interno della giornata, con partenze che avvengono prevalentemente tra le 8 : 00 e le 16 : 00, e conclusioni tra le 10 : 00 e le 17 : 00. Il periodo dell'anno in cui questi spostamenti si verificano va da ottobre a novembre, con casi isolati già da luglio e fino a dicembre, e una leggera concentrazione tra i giorni 7 e 15 del mese.

Gli spostamenti risultano marcatamente eterogenei in termini di durata e distanza, caratterizzati da una forte discrepanza tra stima iniziale e durata effettiva, che suggerisce una possibile sottostima sistematica o imprevisti nel percorso.

- **Durata stimata:** tra 3 e 8 minuti (minimo assoluto pari a 1 minuto, massimo pari a 23 minuti);
- **Durata effettiva:** tra 23 e 96 minuti (minimo assoluto pari a 9 minuti, massimo pari a 505 minuti);
- **Distanza stimata:** tra circa 1.8 e 5.1 km (minimo assoluto 342 metri, massimo 16.4 km).

Il mezzo di trasporto utilizzato è esclusivamente l'automobile, con assenza totale di altri veicoli. Questo, unito all'alta durata effettiva, lascia ipotizzare un contesto di congestione urbana o di spostamenti legati a esigenze lavorative o personali su lunga percorrenza.

| Variabile            | Media     | STD      | Min       | 25%       | Median    | 75%       | Max       |
|----------------------|-----------|----------|-----------|-----------|-----------|-----------|-----------|
| estimated_distance_m | 8398.20   | 4604.64  | 478.0     | 5177.50   | 8213.0    | 11831.0   | 17682.0   |
| latitude_start       | 45.477272 | 0.039844 | 45.358890 | 45.454750 | 45.476910 | 45.512205 | 45.534030 |
| longitude_start      | 9.188983  | 0.060990 | 9.086980  | 9.130935  | 9.193020  | 9.231780  | 9.290390  |
| latitude_end         | 45.473647 | 0.028640 | 45.417470 | 45.451590 | 45.475890 | 45.495430 | 45.529410 |
| longitude_end        | 9.188006  | 0.041527 | 9.084860  | 9.157075  | 9.191940  | 9.225700  | 9.264780  |

Table 5.22: Caratteristiche geospatiali del cluster 511

Dal punto di vista spaziale, le traiettorie coprono sia zone periferiche a est in maniera marginale (come **Calvairate**, **Chiaravalle** e quartieri limitrofi) sia zone ovest della città (in particolare **Baggio**, **Pero**, **Quarto Oggiaro**), che risultano le aree maggiormente coinvolte (vd. tabella 5.22).

Per quanto riguarda l'accessibilità ai mezzi di trasporto pubblico e ai punti di interesse, si riscontrano i seguenti elementi:

- **Alla partenza:**

- **Ferrovie e metropolitana:** generalmente assenti, salvo rari casi isolati (valore massimo pari a 1 per `metro_start`);
- **Superficie:** presenti in modo limitato (tra 0 e 4, massimo 7);
- **POI totali:** tra 30 e 66, con un massimo di 264;
- **Categorie POI:** sporadiche presenze di **turismo e intrattenimento**, **business**, **sport e salute**, **ristorazione e bar**, **servizi**, e **educazione**. I valori più rilevanti (quarto superiore e massimo) si attestano su 1, mentre il minimo e il primo quartile sono quasi ovunque pari a 0.

- **Alla destinazione:**

- **Ferrovie e metropolitana:** assenti nella quasi totalità dei casi (valore massimo per `metro_end` pari a 1);
- **Superficie:** leggermente più presente (tra 1 e 3, massimo 7);
- **POI totali:** tra 18 e 49, con un massimo pari a 126;
- **Categorie POI:** maggiore rappresentatività di **turismo**, **sport e salute**, **servizi**, **business**, e **educazione**. In particolare, **turismo** e **sport** mostrano valori costanti dal secondo quartile in su, suggerendo destinazioni con presenza regolare di attrattori urbani.

In sintesi, il cluster 511 rappresenta spostamenti in automobile durante i giorni feriali, con origine e destinazione distribuite tra zone periferiche e opposte della città, soprattutto a ovest. L'alta durata effettiva rispetto alla stima, unita alla bassa presenza di mezzi pubblici e alla dispersione geografica, suggerisce un utilizzo del veicolo privato per esigenze funzionali, come trasferimenti da/per il lavoro o attività individuali distribuite. La distribuzione dei POI, più marcata all'arrivo, potrebbe indicare destinazioni collegate a servizi, strutture sportive o istituzioni.

### 5.13.3 Confronto tra i cluster 231 e 511

I cluster 231 e 511 rappresentano due profili di mobilità urbana profondamente differenti, sia per caratteristiche temporali e modali, sia per la natura e la localizzazione degli spostamenti

**Periodo e orari di attività:** Il cluster 231 è fortemente caratterizzato da una concentrazione temporale precisa: gli spostamenti avvengono esclusivamente il sabato sera, tra le 18 : 00 e le 21 : 00. Si tratta dunque di un comportamento circoscritto al weekend e legato a fasce orarie serali. Al contrario, il cluster 511 è attivo durante i giorni feriali, in particolare il mercoledì, con un intervallo temporale molto più ampio: le partenze avvengono dalle 8 : 00 alle 16 : 00, e le destinazioni vengono raggiunte fino alle 17 : 00 (vd. figura 5.15). Questo suggerisce che i due cluster rispondano a bisogni diversi: uno ludico-ricreativo e l'altro funzionale o lavorativo.

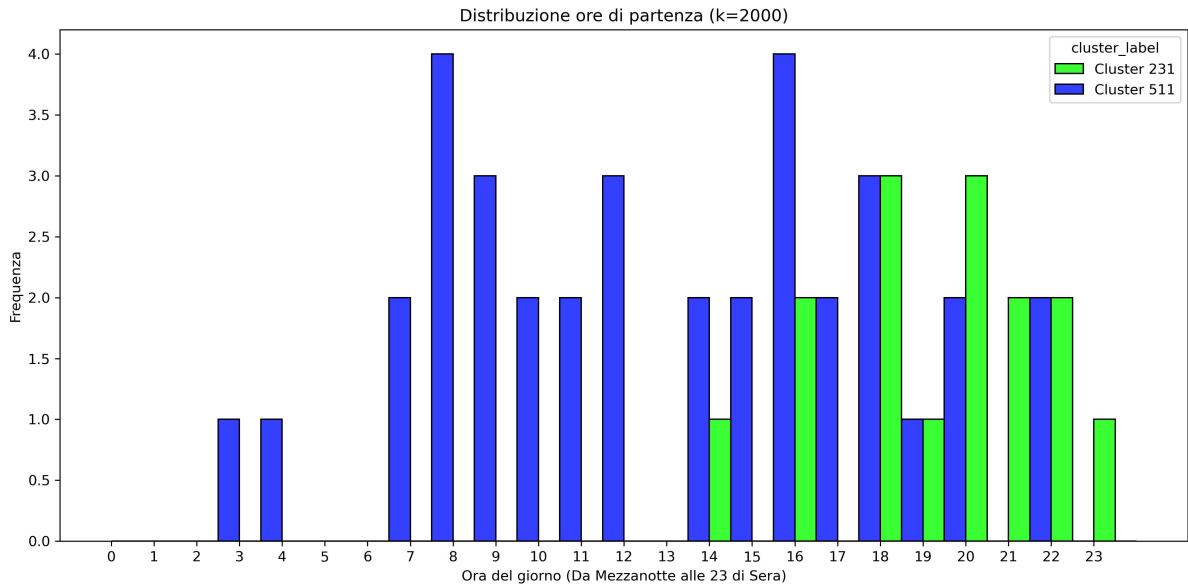


Figure 5.15: Distribuzione temporale per i cluster 231 e 511 ( $k = 2000$ ).

**Mezzi di trasporto:** Un'altra distinzione netta è data dal mezzo di trasporto utilizzato: nel cluster 231 compare esclusivamente il motorscooter in sharing, mentre nel cluster 511 viene utilizzata solo l'automobile. Questo elemento non solo evidenzia diverse preferenze modali, ma riflette anche il tipo di spostamento: breve e agile nel primo caso, più lungo e probabilmente meno servito dal trasporto pubblico nel secondo.

**Durata e distanza degli spostamenti:** I viaggi nel cluster 231 sono brevi e omogenei, con durate effettive contenute (tra 5 e 7 minuti) e distanze limitate, tipiche di tragitti intra-zona o spostamenti di quartiere. Al contrario, nel cluster 511 emerge una forte eterogeneità: le durate effettive sono molto superiori a quelle stimate (fino a oltre 8 volte superiori), con alcuni viaggi che superano abbondantemente i 60 minuti. Le distanze percorse sono più ampie e variabili, suggerendo spostamenti interquartiere o intercomunali.

Anche dal punto di vista spaziale i due cluster si distinguono nettamente. Il cluster 231 è localizzato nelle aree sud, sud-est ed est della città metropolitana (es. **San Donato, Assago, Segrate**), mentre il cluster 511 si sviluppa soprattutto tra la zona nord e ovest (es. **Quarto Oggiaro, Baggio, Pero**), con traiettorie più ampie e dispersive. Inoltre, il cluster 231 mostra una direzionalità dagli estremi verso zone più centrali e ricche di POI, mentre il 511 si muove tra zone periferiche relativamente povere di attrattori (vd. figura 5.16).

**Accessibilità ai mezzi pubblici e presenza di POI:** Nel cluster 231, le aree di partenza sono spesso prive di POI significativi e scarsamente servite, mentre le destinazioni presentano una maggiore densità di attrattori urbani (fino a 486 POI) e una leggera crescita nella presenza di trasporto pubblico di superficie. Il cluster 511, pur partendo e arrivando in zone leggermente più attrezzate (tra 30 e 66 POI in partenza; fino a 126 in arrivo), presenta comunque una scarsa accessibilità al trasporto ferroviario e metropolitano, con una presenza più marcata di POI legati a funzioni quotidiane (sport, educazione, servizi) piuttosto che al tempo libero.

**Finalità degli spostamenti:** Infine, la natura degli spostamenti riflette bisogni diversi:

- Il cluster 231 è indicativo di un comportamento legato allo svago serale del fine settimana, effettuato con mezzi agili per raggiungere zone ricche di attrattori sociali e ricreativi.
- Il cluster 511 appare invece legato a esigenze pratiche e funzionali, come il raggiungimento del luogo di lavoro, di servizi pubblici o privati, o di strutture sportive/educative.

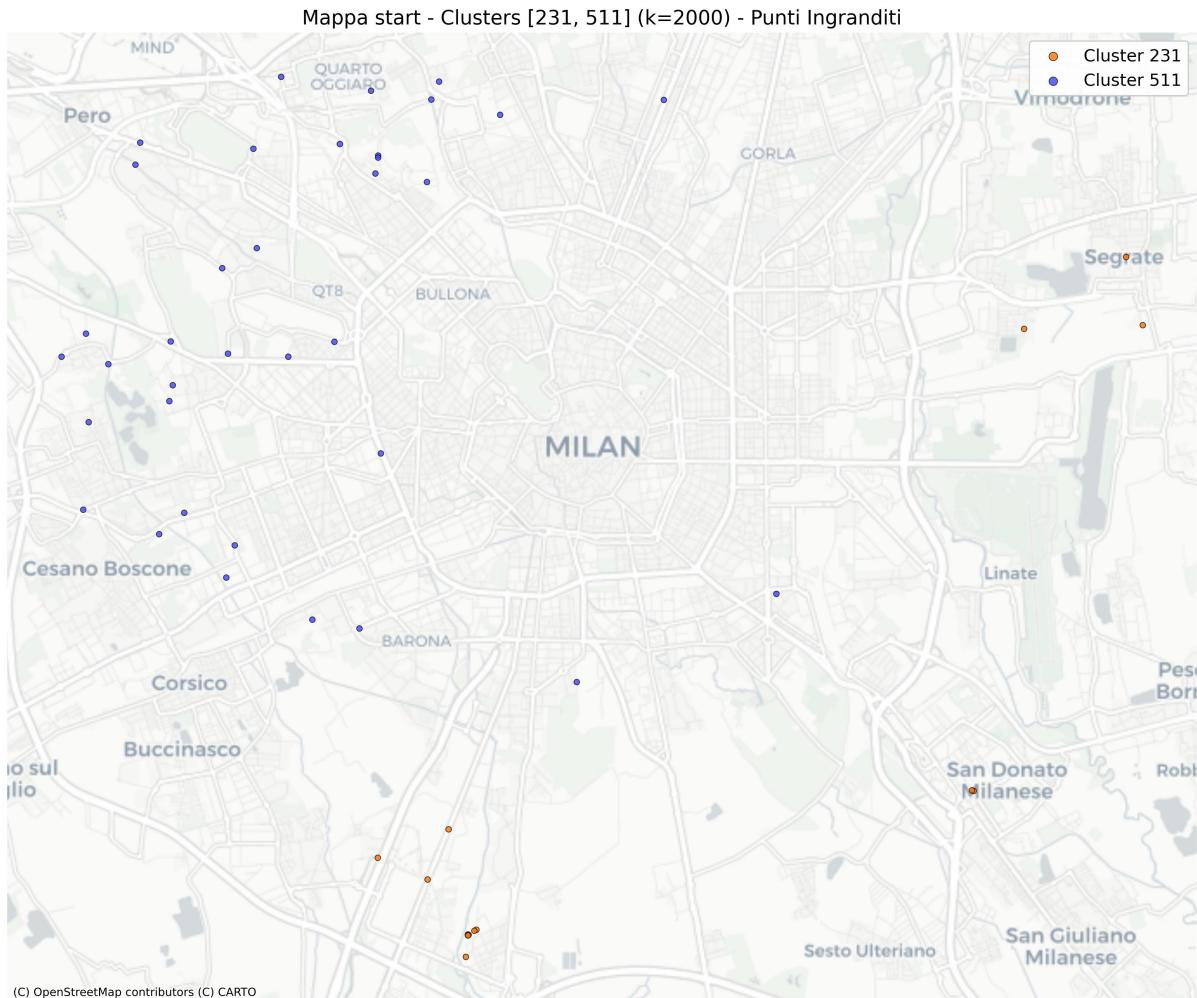


Figure 5.16: Mappa geografica dei cluster 231 e 511 ( $k = 2000$ ).

**Conclusione:** Il confronto tra i cluster 231 e 511 mette in evidenza la pluralità dei comportamenti di mobilità all'interno dell'area milanese: da un lato viaggi brevi e orientati al tempo libero in orari specifici, dall'altro spostamenti lunghi e funzionali distribuiti durante la settimana. Questa diversità sottolinea l'importanza di politiche di mobilità flessibili e multimodali, capaci di rispondere a esigenze sia occasionali sia quotidiane, su scala urbana e metropolitana.

## 5.14 Conclusioni finali

### 5.14.1 Aspettative

All'inizio dell'analisi, mi aspettavo che cambiando il numero di cluster  $k$  sarei riuscito a ottenere una divisione sempre più dettagliata dei comportamenti di mobilità presenti nel dataset. In particolare, pensavo che:

- $k = 12$  e  $k = 14$ : speravo di ottenere una suddivisione netta tra i diversi giorni della settimana, con la possibilità di individuare cluster distinti per i giorni feriali e quelli del fine settimana. Pensavo che il modello potesse cogliere routine giornaliere ricorrenti.
- $k = 50$ : mi aspettavo una distinzione più fine, legata ai diversi giorni del mese. Immaginavo che si potessero cogliere differenze tra inizio, metà e fine mese, oppure tra le varie settimane.
- $k = 100$ : credevo che con un numero maggiore di cluster il modello sarebbe riuscito a cogliere pattern temporali più dettagliati, legati per esempio alle fasce orarie, alla durata degli spostamenti o alla tipologia di mezzo di trasporto utilizzato.

- $k = 500$ : con un valore così alto pensavo di poter ottenere una segmentazione ancora più specifica, utile per individuare cluster caratterizzati da comportamenti particolari o ricorrenti, legati ad esempio a certe zone della città o a determinati momenti della giornata.
- $k = 1000$ : a questo livello, mi aspettavo che i cluster potessero riflettere la suddivisione spaziale della città, in particolare distinguendo tra zone omogenee come gli ACE (Ambiti di Censimento Elementare) o i NIL (Nuclei di Identità Locale).
- $k = 2000$ : infine, con il numero massimo di cluster utilizzato, pensavo di ottenere una distinzione molto dettagliata che potesse evidenziare non solo le aree ACE o NIL, ma anche la presenza e l'influenza di punti di interesse (POI), come scuole, stazioni, luoghi turistici, ecc.

| Valore di $k$ | Osservazioni principali  |
|---------------|--|
| 12, 14        | Cluster molto simili tra loro. Suddivisione quasi netta per mezzo di trasporto (auto, scooter, bici). Alcune informazioni generali anche sui giorni della settimana.           |
| 50            | Inizio di una segmentazione temporale: emergono pattern legati ai giorni della settimana (feriali vs festivi).   |
| 100           | Maggior dettaglio temporale: i cluster mostrano differenze legate alle fasce orarie (mattina, pomeriggio, sera).   |
| 500           | Informazioni combinate: orario, giorni del mese e prime evidenze sui comportamenti di spostamento (durata, distanza). Cluster più frammentati ma informativi.                  |
| 1000, 2000    | Segmentazione spaziale più marcata. Inizio di differenziazione tra zone urbane, con presenza di POI. Tuttavia, la distinzione tra ACE e NIL non è ancora chiaramente definita. |

Table 5.23: Sintesi dei risultati ottenuti in base al numero di cluster  $k$

## 5.15 Discussione e comparazione dei risultati

I risultati ottenuti mostrano che l'aumento progressivo del numero di cluster  $k$  ha effettivamente permesso di cogliere informazioni sempre più dettagliate, anche se con alcune limitazioni.

- $k = 12$  e  $k = 14$ : i risultati ottenuti per questi due valori sono molto simili. Si osserva una suddivisione quasi netta in base ai mezzi di trasporto utilizzati (scooter, auto, bici, ecc.), e si colgono anche informazioni generali legate ai giorni della settimana.
- $k = 50$ : in questo caso emergono chiaramente informazioni sui giorni della settimana, con la possibilità di distinguere tra giorni feriali e festivi. Il modello inizia a cogliere pattern temporali più definiti.
- $k = 100$ : la suddivisione diventa ancora più precisa e focalizzata soprattutto sull'orario degli spostamenti. Emergono cluster legati a fasce orarie specifiche, come mattina, pomeriggio o sera.
- $k = 500$ : con questo valore si notano informazioni combinate su orario, giorni del mese e alcuni indizi anche sugli spostamenti stessi, come durata o distanza. I cluster diventano più ricchi, ma anche leggermente più complessi da interpretare.
- $k = 1000$  e  $k = 2000$ : a questi livelli si ottiene una suddivisione geografica più marcata. I cluster iniziano a riflettere differenze tra zone della città, e in alcuni casi emergono anche informazioni relative ai POI (Point of Interest). Tuttavia, la distinzione tra zone urbanistiche come ACE o NIL non è ancora del tutto chiara.

In generale, le *metriche geometriche* non indicano un valore ottimale univoco per  $k$ , ma confermano che un valore compreso tra  $k = 50$  e  $k = 100$  rappresenta un buon compromesso tra compattezza dei cluster e facilità di interpretazione.

Infine, si potrebbe pensare che aumentando ancora il valore di  $k$  oltre i 2000 sia possibile ottenere una suddivisione ancora più precisa dei POI e magari una distinzione più chiara tra le zone ACE e NIL. Tuttavia, non è detto che questo porti davvero a risultati migliori. Aumentare troppo il numero di cluster non garantisce che i gruppi siano più coerenti o più interessanti dal punto di vista spaziale.

Inoltre, più  $k$  è alto, più diventa difficile analizzare e interpretare i risultati, perché il numero di cluster da considerare cresce molto e richiede tempo e attenzione. Per questo motivo, conviene trovare un miglior compromesso tra dettaglio e semplicità.

In futuro, potrebbe essere utile provare anche altri algoritmi di clustering, che forse riescono a sfruttare meglio la struttura dello spazio latente e rendono l'analisi più chiara e gestibile (vd. tabella 5.23 per un riassunto delle conclusioni finali).

# Chapter 6

## Conclusioni e sviluppi futuri

### 6.1 Valutazione dei metodi adottati

Nel corso dello studio è emersa l'efficacia dell'utilizzo degli **Autoencoder** come strumenti per la riduzione non lineare della dimensionalità del dataset. L'architettura progettata ha permesso di ottenere una rappresentazione compatta e significativa degli spostamenti urbani, mantenendo una buona qualità nella ricostruzione dei dati originari.

Le scelte architettoniche, come la profondità della rete o l'uso della regolarizzazione L1/L2, si sono rivelate adeguate per il tipo di dati trattati. Inoltre, il comportamento della **loss function** durante il training ha mostrato una buona capacità di apprendimento e generalizzazione, con una riduzione progressiva dell'errore anche in assenza di overfitting significativo.

### 6.2 Riflessioni conclusive

L'analisi condotta ha mostrato come le tecniche di apprendimento non supervisionato possano offrire un valido supporto nello studio dei fenomeni complessi legati alla mobilità urbana in sharing. In particolare, l'approccio basato su una pipeline integrata composta da **Autoencoder**, **UMAP** e **K-Means** ha permesso di esplorare i dati in profondità e di individuare strutture latenti altrimenti difficili da osservare.

Attraverso l'addestramento dell'Autoencoder, è stato possibile ottenere una rappresentazione compressa dei dati originali, in uno spazio latente che ha evidenziato una certa capacità di organizzare gli spostamenti secondo criteri temporali, modali e, in parte, anche spaziali. L'utilizzo successivo di **UMAP** ha ulteriormente favorito l'analisi, permettendo una riduzione efficace della dimensionalità per la visualizzazione e la successiva applicazione del clustering.

L'algoritmo **K-Means**, applicato su questo spazio ridotto, ha prodotto cluster interpretabili in diversi casi, soprattutto per valori medio-bassi di  $k$ . Sono stati identificati gruppi di spostamenti accomunati da caratteristiche specifiche, ad esempio legate alla fascia oraria (pendolarismo, notturno, festivo), al mezzo utilizzato (biciclette, scooter, automobili) o al tipo di zona attraversata (centro, periferia, aree con funzioni specifiche).

Tuttavia, non tutte le configurazioni hanno portato a risultati immediatamente leggibili. All'aumentare del numero di cluster, l'interpretazione è diventata progressivamente più complessa. Questo perché i gruppi ottenuti tendevano a differenziarsi tra loro per dettagli sempre più fini, spesso difficili da descrivere in modo chiaro. In altre parole, se da un lato aumentare  $k$  consente di catturare maggiore variabilità nel dataset, dall'altro rende più difficile sintetizzare e comunicare i risultati. Inoltre, non è garantito che suddivisioni più numerose portino a una segmentazione spaziale più significativa o utile.

Anche dal punto di vista sperimentale, non tutte le scelte si sono rivelate efficaci. Architetture di Autoencoder più complesse non hanno portato benefici evidenti rispetto a quelle più semplici, mentre l'utilizzo di tecniche alternative di riduzione dimensionale come **PCA** ha restituito proiezioni meno stabili e meno interpretabili rispetto a quelle ottenute con **UMAP**.

In generale, la pipeline proposta si è dimostrata solida e promettente. Ha permesso di esplorare e visualizzare la struttura nascosta nei dati, di individuare cluster significativi e di verificare la possibilità di descrivere comportamenti ricorrenti negli spostamenti urbani. Allo stesso tempo, il lavoro ha evidenziato alcuni limiti importanti: l'interpretabilità dei risultati dipende fortemente dalla qualità e dalla natura dei dati disponibili, e l'utilizzo di un solo algoritmo di clustering può non essere sufficiente per cogliere tutte le strutture rilevanti nei dati.

Per questo motivo, un possibile sviluppo futuro consiste nell'esplorazione di altri algoritmi di clustering, come DBSCAN, HDBSCAN o approcci gerarchici, che potrebbero essere più adatti a gestire la complessità dello spazio latente e ad adattarsi meglio alla forma non sferica dei cluster. Inoltre, l'arricchimento del dataset con variabili semantiche o contesto urbano (come la presenza di POI, la funzione dei quartieri, dati socio-demografici, ecc.) potrebbe migliorare la qualità e la leggibilità dei risultati.

In conclusione, il lavoro svolto ha dimostrato la fattibilità e l'utilità di un approccio non supervisionato per l'analisi della mobilità urbana in sharing. L'integrazione di tecniche di deep learning, riduzione dimensionale e clustering ha consentito di ottenere risultati interessanti e, in diversi casi, significativi. Questo tipo di analisi può offrire uno strumento utile per il supporto alle decisioni in ambito urbano, ad esempio nella progettazione di servizi di mobilità più efficienti e adattati alle reali esigenze degli utenti.

### 6.3 Limiti e criticità dell'approccio

Nonostante i risultati incoraggianti, il lavoro presenta alcuni limiti strutturali che meritano di essere discussi. In primo luogo, il dataset utilizzato, pur essendo ricco e ben strutturato, poteva essere esteso ulteriormente sia in termini di copertura temporale sia nella varietà delle variabili considerate. Tuttavia, per motivi legati ai tempi di esecuzione degli algoritmi e alla disponibilità limitata di risorse computazionali, si è scelto di limitare la dimensione del campione analizzato al solo 10% del dataset totale.

Infine, l'approccio non supervisionato, sebbene potente e flessibile, è intrinsecamente sensibile alla presenza di bias latenti nei dati, i quali risultano difficili da individuare e correggere in assenza di etichette o annotazioni esplicite.

### 6.4 Implicazioni per l'analisi della mobilità urbana

I risultati ottenuti possono offrire un contributo concreto alla comprensione dei fenomeni di mobilità urbana, evidenziando come la mobilità in sharing venga utilizzata in modo eterogeneo nei diversi contesti urbani.

L'identificazione di cluster distinti permette di ottenere informazioni utili per la pianificazione urbana e per la gestione operativa dei servizi di sharing. Ad esempio, è possibile individuare zone a elevata domanda in determinati orari, prevedere comportamenti pendolari o ricreativi, e ottimizzare la distribuzione dei mezzi.

Questa conoscenza può essere utile per supportare decisioni data-driven in ambito pubblico (mobilità sostenibile, policy urbane) e privato (allocazione mezzi, modelli di pricing, campagne marketing mirate).

### 6.5 Sviluppi futuri

Il lavoro apre a numerose possibilità di estensione:

- **Uso del dataset completo:** estendere l'analisi all'intero dataset, senza campionamento, potrebbe permettere una rappresentazione più fedele e dettagliata dei pattern di mobilità.
- **Esplorazione di algoritmi alternativi:** valutare l'impiego di algoritmi di clustering come DBSCAN, più adatti a rilevare strutture a densità variabile, potrebbe migliorare l'individuazione di cluster non lineari o di dimensioni irregolari.
- **Integrazione di nuovi dataset:** includere informazioni meteo, eventi cittadini o dati socio-demografici potrebbe arricchire le variabili e offrire spiegazioni più solide sui pattern rilevati.

- **Esplorazione di modelli alternativi:** l'uso di Variational Autoencoder (VAE), *Contrastive Learning*, o tecniche di clustering gerarchico potrebbe portare a rappresentazioni latenti più informative o più stabili.

Questa progressione mostra come la scelta del parametro  $k$  influenzi profondamente il tipo di informazioni che si possono estrarre dai dati: si passa da una visione complessiva dei macro-pattern a una caratterizzazione dettagliata e locale del comportamento urbano.

# Bibliography

- [1] Alessandro Zoia. CNN: Convolutional Neural Networks | Development, Java | HTML.it. <https://www.html.it/pag/406477/cnn-convolutional-neural-networks/>.
- [2] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, seconda edition, 2019.
- [3] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, January 2019.
- [4] Charles Elkan. Using the Triangle Inequality to Accelerate k-Means. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 147–153, 2003.
- [5] David Arthur and Sergei Vassilvitskii. K-Means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA, January 2007. Society for Industrial and Applied Mathematics.
- [6] David Sculley. Web-Scale K-Means Clustering. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 1177–1178, 2010.
- [7] Matthew Watson Francois Chollet. *Deep Learning with Python*. Manning Publications Company, terza edition, 2021.
- [8] GeeksforGeeks. Ball Tree and KD Tree Algorithms - GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/ball-tree-and-kd-tree-algorithms/>.
- [9] Lin, Jie and Long, Liang. What neighborhood are you in? Empirical findings of relationships between household travel and neighborhood characteristics. *Transportation*, 35:739–756, 2008.
- [10] Martí, Pablo and Serrano-Estrada, Leticia and Nolasco-Cirugeda, Almudena and López Baeza, Jesùs. Revisiting the Spatial Definition of Neighborhood Boundaries: Functional Clusters versus Administrative Neighborhoods. *Journal of Urban Technology*, 29:73–94, 2022.
- [11] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, September 2020.
- [12] Park, Souneil and Serr{\'a}, Joan and Frias-Martinez, Enrique and Oliver, Nuria. MobInsight: A Framework Using Semantic Neighborhood Features for Localized Interpretations of Urban Mobility. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8:1–25, 2018.
- [13] Pedregosa, Fabian and Varoquaux, Gael and Gramfort, Alexandre and Michel, Vincent and Thirion, Bertrand and Grisel, Olivier and Blondel, Mathieu and Prettenhofer, Peter and Weiss, Ron and Dubourg, Vincent and Vanderplas, Jake and Passos, Alexandre and Cournapeau, David and Brucher, Matthieu and Perrot, Matthieu and Duchesnay, Edouard. BallTree. <https://scikit-learn/stable/modules/generated/sklearn.neighbors.BallTree.html>.
- [14] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, 33(11):2881–2907, October 2021.
- [15] V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, March 2019.

# Appendice

## Link al notebook o repository

Link al repository GitHub contenente il codice usato per la sperimentazione:  
<https://github.com/Scoal2053/Relazione-Tesi-prova-finale.git>.

Il codice sviluppato è rilasciato sotto licenza open source **MIT License**<sup>1</sup>. Per maggiori dettagli, si veda il file **LICENSE** incluso nel repository.

## Configurazioni sperimentali

Le configurazioni hardware e software adottate per le diverse fasi del progetto sono:

### Ambienti di calcolo

Le sperimentazioni sono state condotte principalmente su Google Colab, sfruttando differenti configurazioni hardware in base alla fase del processo:

- **Generazione e analisi dei cluster (T4 GPU)**
  - RAM di sistema: 12.7 GB
  - RAM GPU: 15.0 GB
  - Spazio su disco: 112.6 GB
- **Addestramento dell'autoencoder e rilevamento delle anomalie (CPU)**
  - RAM: 12.7 GB
  - Spazio su disco: 107.7 GB
- **Creazione del tensore dati (v2-8 TPU)**
  - RAM: 334.6 GB
  - Spazio su disco: 225.3 GB

### Ambiente locale

La generazione dei grafici e la visualizzazione finale dei risultati sono state realizzate in ambiente locale, su una macchina con le seguenti specifiche hardware:

- RAM: 16.0 GB
- RAM GPU: 128 MB
- Spazio su disco: 477 GB

---

<sup>1</sup><https://opensource.org/licenses/MIT>

<sup>1</sup>Disclaimer: Nella preparazione di questa tesi sono stati utilizzati strumenti di Intelligenza Artificiale come ausilio per la revisione linguistica e stilistica minore.