

UNIVERSITÀ DEGLI STUDI DI MODENA E  
REGGIO EMILIA

DIPARTIMENTO DI INGEGNERIA "ENZO FERRARI"

Laurea Triennale in Ingegneria Informatica

**Mobilità urbana data-driven: event  
detection e predizione dei flussi con  
Gradient Boosting**

Relatore:

**Prof: Nicola Bicocchi**

Candidato:

**Michele Arioli**

---

Anno Accademico 2024/2025

# Indice

<b>Introduzione</b>	<b>5</b>
<b>1 Archiviazione e pulizia dei dati</b>	<b>6</b>
1.1 Struttura dei dati . . . . .	6
1.2 Archiviazione con PostgreSQL . . . . .	8
1.3 Pulizia e validazione dei dati . . . . .	9
<b>2 Analisi preliminari dei dati</b>	<b>11</b>
2.1 Librerie per analisi e rappresentazione dei dati . . . . .	11
2.2 Analisi Datavalue . . . . .	12
2.3 Analisi spostamenti per categoria . . . . .	14
2.4 Rappresentazioni geografiche . . . . .	15
<b>3 Analisi dei Flussi</b>	<b>17</b>
3.1 Analisi dei flussi di mobilità nell'area di Bologna . . . . .	18
<b>4 Event Detection</b>	<b>21</b>
4.1 Comune di Bologna . . . . .	21
4.2 Stadio Bologna . . . . .	23
4.3 Autodromo di Imola . . . . .	24
4.4 Comune di Ferrara . . . . .	25
4.5 Comune di Modena . . . . .	25
<b>5 Previsione flussi di mobilità con Gradient Boosting</b>	<b>27</b>
5.1 Librerie e strumenti utilizzati . . . . .	28
5.2 Predizione dei flussi con dati locali . . . . .	28
5.3 Utilizzo del dataset completo per la stima dei flussi . . . . .	32
5.4 Generalizzazione del modello . . . . .	33
<b>6 Mobility api</b>	<b>37</b>
6.1 Implementazione con FastAPI e Uvicorn . . . . .	37
6.2 Modello predittivo . . . . .	38
6.3 Funzionamento engine . . . . .	38
<b>Conclusioni</b>	<b>40</b>

# Elenco delle figure

1.1	Query per la creazione della tabella <code>geom_data</code>	8
1.2	Query per la creazione della tabella <code>movements</code>	9
2.1	Distribuzione della variabile Datavalue	12
2.2	Numero totale spostamenti e distribuzione spostamenti per giorno della settimana	13
2.3	Distribuzione degli spostamenti per fascia oraria	13
2.4	Distribuzione degli spostamenti per fascia d'età	14
2.5	Confronto spostamenti tra stranieri e lavoratori	15
2.6	Rappresentazione partenze degli over 60	16
3.1	Rappresentazione matrice Origin-Destination	17
3.2	Confronto spostamenti agosto-settembre tra provincie di Bologna e Rimini	18
3.3	Top 10 flussi nel comune di Bologna	18
3.4	Top 15 flussi tra comune e provincia di Bologna	19
3.5	Confronto Flussi giorno feriale e festivo	20
4.1	Flusso orario comune di Bologna con z-score	22
4.2	Flusso orario ACE Bologna fiere con z-score	22
4.3	Flusso orario ACE Bologna stazione con z-score	23
4.4	Flusso orario ACE Comune di Dozza con z-score	23
4.5	Flusso orario ACE Bologna stadio con z-score	24
4.6	Flusso orario ACE autodromo Imola con z-score	24
4.7	Flusso orario ACE comune di Ferrara con z-score	25
4.8	Flusso orario ACE comune di Modena con z-score	26
5.1	Dataset con feature per analisi su Dozza	30
5.2	Risultati confronto feature	31
5.3	Dataset che ricopre tutti gli ACE	32
5.4	Risultati Gradient Boosting su Dozza secondo approccio	33
5.5	Distribuzione Mape nei tre set	34
5.6	ACE più prevedibili	35
5.7	ACE meno prevedibili	36

6.1	Risultati modello api . . . . .	38
6.2	Esempio di richiesta . . . . .	39
6.3	Esempio di risposta . . . . .	39

# Introduzione

Questa tesi affronta il problema dell’analisi e della previsione dei flussi di mobilità urbana nella regione Emilia-Romagna, utilizzando un dataset fornito da TIM Enterprise. Il dataset registra i movimenti tra aree censuarie elementari (ACE) attraverso segnali di rete mobile nel periodo agosto-settembre 2019, includendo informazioni spaziali, temporali e demografiche (età, genere, nazionalità e motivazione dello spostamento).

L’obiettivo è duplice: da un lato, esplorare in profondità le dinamiche della mobilità, evidenziandone caratteristiche salienti e variazioni temporali; dall’altro, sviluppare un modello predittivo in grado di stimare il numero di ingressi in ciascuna ACE, supportando così attività di pianificazione territoriale e gestione urbana.

Il lavoro si articola in più fasi: pulizia e strutturazione dei dati tramite PostgreSQL, analisi esplorativa dei flussi di spostamento e rappresentazione geografica tramite strumenti Python (Folium, GeoPandas), modellazione predittiva tramite tecniche di Machine Learning, e infine sviluppo di un’API in FastAPI per rendere il modello accessibile.

Per la previsione dei flussi, è stato adottato un modello di regressione basato su Gradient Boosting, ottimizzato con GridSearchCV e validato tramite MAE e MAPE. Il modello ha mostrato buona accuratezza sia su casi locali (es. Dozza), sia su scala regionale, con generalizzazione su tutte le ACE.

Il lavoro offre infine strumenti pratici per analizzare, anticipare e integrare le dinamiche di mobilità in sistemi di supporto decisionale, con potenziali applicazioni in ambito urbano e di pianificazione territoriale.

# Capitolo 1

# Archiviazione e pulizia dei dati

La gestione efficace dei dati grezzi è stata una fase fondamentale del progetto. Considerato l'elevato numero di record e la varietà di file forniti, si è resa necessaria un'infrastruttura in grado di supportare interrogazioni efficienti, verifiche di integrità e operazioni di pulizia su larga scala. In questo capitolo viene descritta la struttura dei dati e le scelte adottate per l'archiviazione e la successiva validazione del dataset, con particolare attenzione all'utilizzo del database PostgreSQL e agli script di pulizia sviluppati in Python.

## 1.1 Struttura dei dati

Il dataset utilizzato per questo progetto è stato fornito da TIM Enterprise e si basa su dati aggregati ottenuti tramite segnali di telefonia mobile, anonimizzati e geocalizzati. Questi dati descrivono i flussi di spostamento di persone tra diverse aree censuarie (ACE) della regione Emilia-Romagna, nel periodo compreso tra agosto e settembre 2019.

### File fdestinationdata

Il cuore del dataset è costituito da **671 file CSV**, ognuno dei quali rappresenta i movimenti registrati in un intervallo temporale di 24 ore. Ciascun file contiene una serie di record (nello specifico, il file con il maggior numero di record ha 89.607 righe, mentre quello con il minor numero ha 4.157 righe), ciascuno dei quali rappresenta un flusso di persone da un'area geografica a un'altra in un dato giorno, corredata da informazioni temporali, demografiche e comportamentali.

Le colonne principali dei file sono:

- **layerid** e **toid**: identificatori dell'area geografica di partenza e di arrivo. Ogni codice rappresenta una zona ACE ed è composto da una struttura gerarchica: Regione | Provincia | Comune | Area | Altro.
- **datefrom** e **dateto**: indicano rispettivamente la data e l'ora di inizio e fine dello spostamento.

- **toname**: nome dell'area di destinazione.
- **datavalue**: indica il numero di persone che si spostano dall'area layerid all'area toid.

Inoltre datavalue si compone delle seguenti colonne:

- **F1 – F6**: numero di individui suddivisi in sei fasce d'età.
- **Gm, Gf**: numero di uomini e donne coinvolti nello spostamento.
- **Ni, Ns**: numero di italiani e stranieri.
- **Tb, Tc**: motivazione dello spostamento, rispettivamente per lavoro (business) o consumo (consumer).

Complessivamente, ogni record offre una fotografia dettagliata di uno specifico flusso giornaliero, con la possibilità di segmentare gli spostamenti per molteplici dimensioni: spazio, tempo, età, genere, cittadinanza e scopo del viaggio.

## File layers

Parallelamente ai dati CSV, è stato fornito un file aggiuntivo `layers.csv`, contenente la rappresentazione geografica delle ACE in formato **GeoJSON**. Questo file, composto da 506 righe, associa a ciascun ID geografico (corrispondente ai valori di `layerid` e `toid`) una struttura GeoJSON strutturata come segue:

```
{
  "type": "Feature",
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [[[lon1, lat1], [lon2, lat2], ..., [lonN, latN]]]]
  },
  "properties": {},
  "id": "ID_ACE"
}
```

In questa struttura:

- **type**: **"Feature"** definisce l'oggetto come un'entità geografica.
- **geometry.type**: **"MultiPolygon"** indica che l'area può essere composta da più poligoni distinti (utile in presenza di frazioni o aree non contigue).
- **coordinates** contiene le coordinate geografiche in formato longitudine e latitudine.
- **id** rappresenta l'identificativo dell'area ACE, univoco e corrispondente agli altri file.

La presenza del campo GeoJSON ha permesso l'integrazione spaziale tra i dati dei flussi e la geometria delle aree ACE, rendendo possibile lo sviluppo di visualizzazioni geografiche interattive mediante strumenti come Folium e GeoPandas.

Infine, è opportuno sottolineare che la granularità dei dati è giornaliera per ogni coppia di aree e che il livello di dettaglio, sia temporale che demografico, rende questo dataset particolarmente adatto allo studio di fenomeni dinamici e complessi come gli spostamenti di popolazione.

## 1.2 Archiviazione con PostgreSQL

Per la memorizzazione dei dati è stato utilizzato **PostgreSQL** [6], un sistema di gestione di basi di dati relazionali open-source, noto per la sua robustezza, flessibilità e capacità di gestire grandi moli di dati. L'intero dataset, composto da 671 file CSV relativi ai movimenti tra ACE e un file contenente le geometrie spaziali delle aree, è stato importato in due tabelle principali:

- **geom\_data**: contenente i dati spaziali relativi agli ACE, importati direttamente dal file `layers.csv`.

```
CREATE TABLE geom_data (
    layerid VARCHAR(18) PRIMARY KEY,
    geojson JSONB
);
```

Figura 1.1: Query per la creazione della tabella `geom_data`

- **movements**: contenente i dati sui flussi di mobilità giornalieri.

```

CREATE TABLE movements(
    layerid VARCHAR(18),
    datefrom TIMESTAMP,
    dateto TIMESTAMP,
    datavalue INT,
    toid VARCHAR(18),
    toname VARCHAR(100),
    Ni INT,
    Ns INT,
    Tb INT,
    Tc INT,
    Gm INT,
    Gf INT,
    F1 INT,
    F2 INT,
    F3 INT,
    F4 INT,
    F5 INT,
    F6 INT,
    PRIMARY KEY (layerid, toid, datefrom, dateto),
    FOREIGN KEY (layerid) REFERENCES geom_data(layerid)
);

```

Figura 1.2: Query per la creazione della tabella movements

Le tabelle sono state progettate per supportare una struttura coerente con il contenuto informativo dei file sorgente, con vincoli di integrità referenziale e controlli automatici sulle chiavi primarie. Il caricamento dei dati è stato gestito tramite script Python dedicati, che, oltre a eseguire l'inserimento, si occupano anche della validazione e della trasformazione iniziale.

Nel paragrafo successivo verranno descritte le operazioni di pulizia effettuate prima dell'importazione, ovvero i criteri di validazione adottati per la rimozione di record non coerenti.

### 1.3 Pulizia e validazione dei dati

Per garantire la qualità e la coerenza del dataset, sono stati sviluppati due script principali in Python: `data_cleansing1.py` e `data_cleansing2.py`. Questi script hanno il compito di analizzare i file CSV in ingresso, filtrare i record non validi e caricare i dati corretti nel database chiamato `project_mobility`.

Le operazioni di pulizia includono:

- **Controllo delle date:** per ogni record viene verificato che `datefrom` sia antecedente a `dateto`. In caso contrario, il record viene scartato.
- **Verifica di coerenza su datavalue:** la somma delle variabili demografiche (età, genere, cittadinanza, motivazione) deve essere coerente con il valore di `datavalue`. Solo i record in cui le somme risultano corrette vengono inseriti nel database.
- **Controllo geografico:** vengono esclusi i record che non riguardano spostamenti interni alla regione Emilia-Romagna.

- **Gestione dei valori mancanti:** eventuali campi vuoti vengono gestiti tramite conversione sicura in numeri decimali o scartati automaticamente dallo script.
- **Validazione chiavi primarie:** lo script controlla che non vi siano duplicati rispetto alle chiavi primarie definite nel database.

Il file `layers.csv` è stato importato nel database direttamente in `geom_data` attraverso la piattaforma pgAdmin.

Questa fase ha permesso di costruire un dataset pulito, consistente e pronto per l'analisi statistica e la modellazione predittiva nei capitoli successivi.

# Capitolo 2

## Analisi preliminari dei dati

In questo capitolo vengono presentate le analisi esplorative condotte sui dati di mobilità al fine di comprendere le dinamiche principali degli spostamenti nella regione Emilia-Romagna. Le elaborazioni sono state realizzate utilizzando Python, con il supporto delle librerie NumPy, Pandas e Matplotlib, strumenti fondamentali per il trattamento efficiente dei dati e la loro rappresentazione grafica.

### 2.1 Librerie per analisi e rappresentazione dei dati

Nel corso delle analisi preliminari, sono state utilizzate tre librerie fondamentali [1] per la manipolazione e visualizzazione dei dati: NumPy, Pandas e Matplotlib.

**NumPy** fornisce strutture dati efficienti per l'elaborazione numerica, in particolare array multidimensionali, e una vasta gamma di funzioni matematiche ad alte prestazioni. È stata utilizzata per operazioni vettoriali e trasformazioni numeriche sui dataset, facilitando il calcolo di statistiche descrittive e la gestione di grandi volumi di dati.

**Pandas** rappresenta la colonna portante per la gestione dei dati tabellari. I DataFrame offrono un'interfaccia intuitiva e potente per la pulizia, la trasformazione e l'analisi dei dati. In questo progetto, è stata utilizzata per filtrare i record rilevanti, aggregare i dati secondo variabili temporali e demografiche, e costruire dataset pronti per l'analisi statistica e la modellazione predittiva.

**Matplotlib** è stata impiegata per la realizzazione di grafici statici, tra cui istogrammi, scatter plot e violin plot, con lo scopo di visualizzare la distribuzione dei flussi di mobilità e individuare pattern temporali. Grazie alla sua flessibilità, ha consentito di personalizzare le visualizzazioni per adattarle al contesto analitico, migliorando l'interpretabilità dei risultati.

Queste tre librerie, utilizzate congiuntamente, hanno costituito la base operativa per l'analisi esplorativa, permettendo di passare in modo fluido dalla manipolazione numerica alla rappresentazione visiva dei dati.

## 2.2 Analisi Datavalue

La variabile `datavalue` rappresenta il numero di persone che si spostano da un'area censuaria di partenza (ACE) a un'area di arrivo, in un determinato intervallo orario. Essa costituisce il cuore informativo del dataset, in quanto consente di quantificare l'intensità dei flussi di mobilità tra zone geografiche.

L'analisi di questa variabile è fondamentale per comprendere la distribuzione e la natura dei movimenti registrati, identificare eventuali anomalie, picchi o concentrazioni, e individuare pattern temporali ricorrenti.

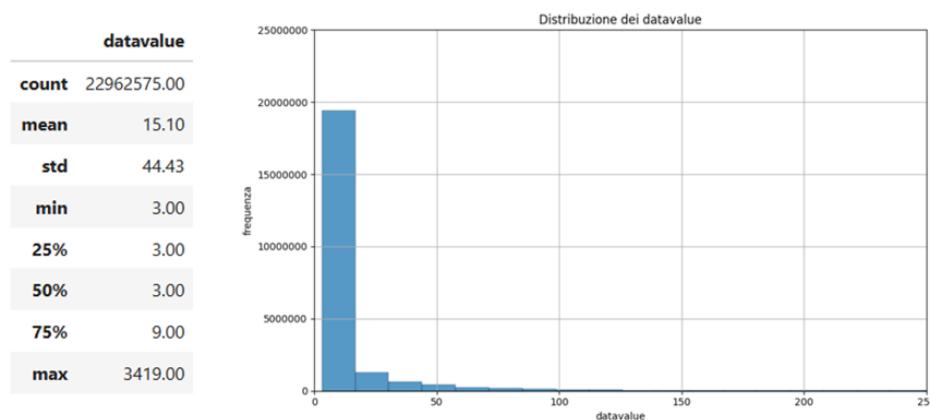


Figura 2.1: Distribuzione della variabile Datavalue

Dalla distribuzione statistica e dal grafico si osserva che la maggior parte dei flussi di mobilità coinvolge un numero molto ridotto di persone. Il valore minimo e mediano di `datavalue` è pari a 3, indicando una prevalenza di micro-spostamenti tra le aree censuarie. La media è di 15.1, ma la deviazione standard elevata (44.43) rivela una forte dispersione e asimmetria nella distribuzione.

Il valore massimo raggiunge 3419, evidenziando la presenza di casi eccezionali con volumi di traffico molto elevati, anche se rari. Il grafico mostra una distribuzione fortemente sbilanciata a destra (right-skewed), tipica dei fenomeni in cui pochi eventi concentrano grandi quantità mentre la maggior parte rimane su livelli minimi. Questa struttura suggerisce che il sistema di mobilità analizzato è caratterizzato da un'alta frequenza di spostamenti locali o occasionali, con pochi nodi ad alta intensità di traffico.

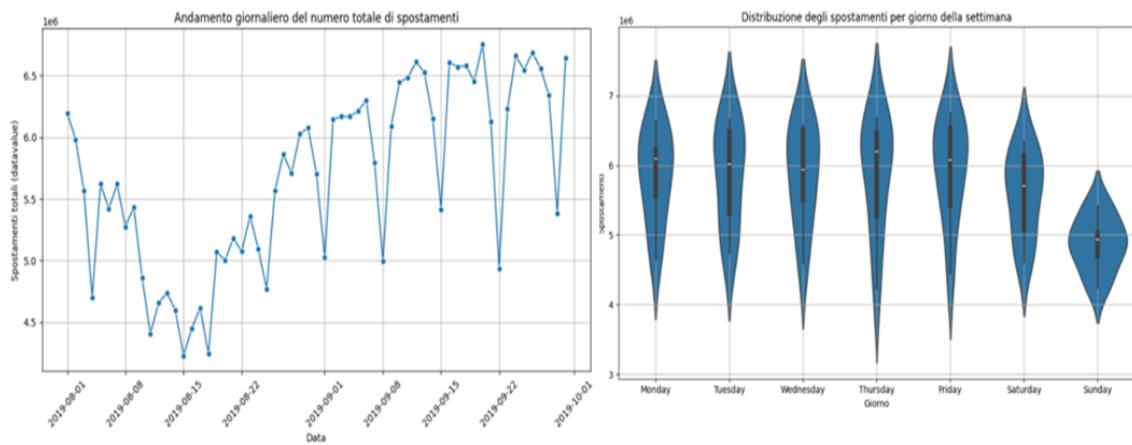


Figura 2.2: Numero totale spostamenti e distribuzione spostamenti per giorno della settimana

I grafici mostrano l'**andamento temporale degli spostamenti** nella regione Emilia-Romagna durante i mesi di agosto e settembre 2019.

- Il primo evidenzia l'andamento giornaliero del numero totale di spostamenti. Si notano oscillazioni regolari settimanali, con un calo ricorrente nei fine settimana, in particolare la domenica, e una tendenza crescente da metà agosto a fine settembre.
- Il violin plot mostra la distribuzione degli spostamenti per giorno della settimana. I giorni feriali presentano volumi più alti e stabili, mentre la domenica registra i valori più bassi, confermando una riduzione della mobilità legata a esigenze lavorative.

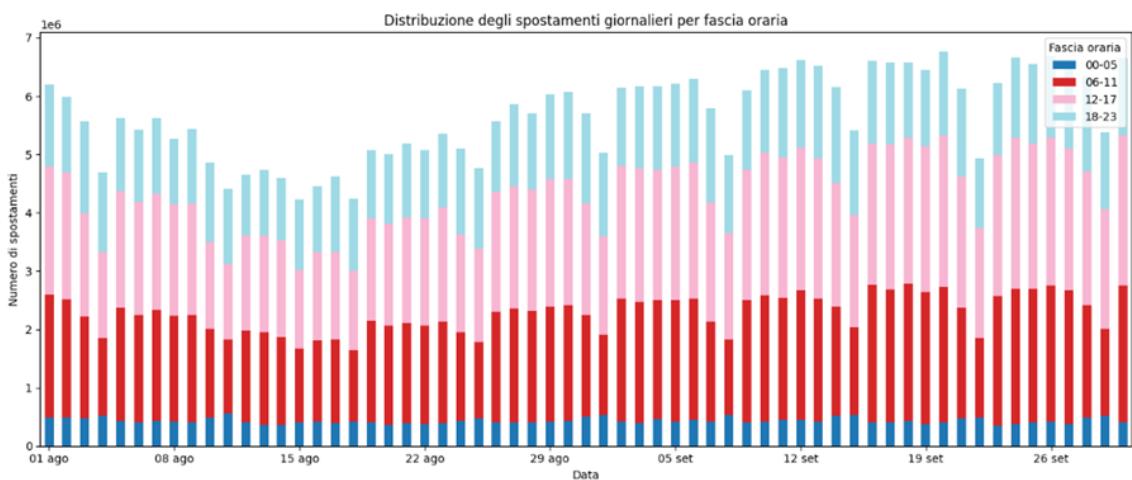


Figura 2.3: Distribuzione degli spostamenti per fascia oraria

Il grafico mostra la **distribuzione giornaliera degli spostamenti suddivisi per fascia oraria**. Dall'analisi emerge che:

- La maggior parte degli spostamenti avviene tra le 12 e le 23, con un picco nel tardo pomeriggio-sera.
- Gli spostamenti nella fascia mattutina (06–11) sono stabili e rappresentano una parte significativa del totale, probabilmente legati ad attività lavorative e scolastiche.
- Le fasce notturne (00–05) presentano volumi molto ridotti, come atteso.

## 2.3 Analisi spostamenti per categoria

In questa sezione vengono analizzati i flussi di mobilità suddivisi per categoria, al fine di evidenziare differenze nei comportamenti di spostamento tra fasce d'età, italiani e stranieri e motivazioni legate allo spostamento.

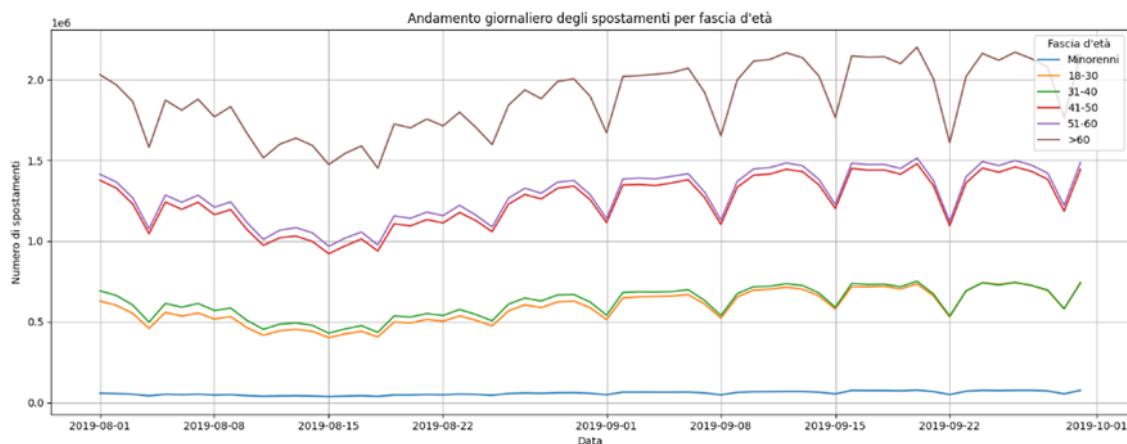


Figura 2.4: Distribuzione degli spostamenti per fascia d'età

Il grafico rappresenta l'**andamento giornaliero degli spostamenti per fascia d'età**.

La fascia **over 60** risulta quella con il numero più elevato di spostamenti, superando costantemente le altre categorie. Questo comportamento potrebbe essere legato a fattori stagionali (es. ferie, turismo, tempo libero).

I **minorenni** sono il gruppo meno mobile, coerentemente con il periodo estivo in cui le scuole sono chiuse.

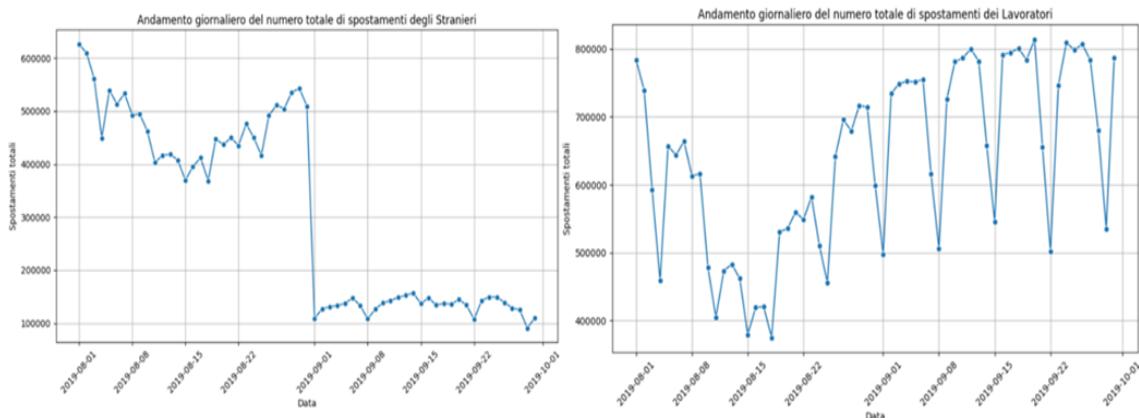


Figura 2.5: Confronto spostamenti tra stranieri e lavoratori

I due grafici mostrano l'**andamento giornaliero del numero totale di spostamenti** per due categorie demografiche: stranieri (a sinistra) e lavoratori (a destra).

Possiamo vedere come gli spostamenti degli **stranieri** siano relativamente stabili per gran parte di agosto, ma subiscano un calo improvviso e marcato a partire da inizio settembre, mostrando il rientro post-estate.

Situazione opposta per quanto riguarda i **lavoratori**, dove possiamo vedere un calo netto durante le due settimane centrali di agosto e una forte crescita a settembre con una forte ciclicità settimanale.

## 2.4 Rappresentazioni geografiche

Per comprendere meglio la distribuzione spaziale dei flussi di mobilità, è stata realizzata una **rappresentazione geografica interattiva dei dati**. Questa visualizzazione consente di esplorare i volumi di spostamenti tra le diverse aree censuarie della regione Emilia-Romagna, evidenziando in modo intuitivo le zone caratterizzate da maggiore o minore intensità di traffico.

L'elaborazione è stata realizzata in Python utilizzando la libreria **Folium**[5], un framework open-source basato su **Leaflet.js**, progettato per costruire mappe interattive a partire da dati geografici. Folium si integra perfettamente con **GeoPandas** [4], che estende le funzionalità di pandas per la gestione di dati geospaziali, e con **shapely**, una libreria per la manipolazione di oggetti geometrici.

A partire da una query SQL, i dati aggregati dei flussi di mobilità (suddivisi per età, genere, nazionalità e motivazione) vengono uniti alle geometrie delle ACE e convertiti in un oggetto **GeoDataFrame**. Le geometrie vengono ricostruite a partire dai dati in formato **GeoJSON** tramite la funzione **shape** di **shapely**. Per ciascuna variabile (ad esempio la fascia d'età over 60), viene generato un layer separato della mappa, in cui ogni area è colorata in modo proporzionale all'intensità del flusso.

Il risultato finale è una **mappa interattiva multilivello**, in cui l'utente può attivare o disattivare i singoli layer corrispondenti alle varie categorie (es. over 60,

donne, lavoratori, stranieri), esplorando visivamente i pattern di mobilità. Nella Figura 2.6, ad esempio, si evidenziano le partenze degli individui nella fascia d'età over 60: le zone con maggiore intensità si concentrano nei poli urbani e lungo la costa romagnola, suggerendo una mobilità legata principalmente al turismo e al tempo libero.

Questa modalità di rappresentazione risulta particolarmente utile per identificare aree con flussi anomali e mettere in luce disomogeneità territoriali.

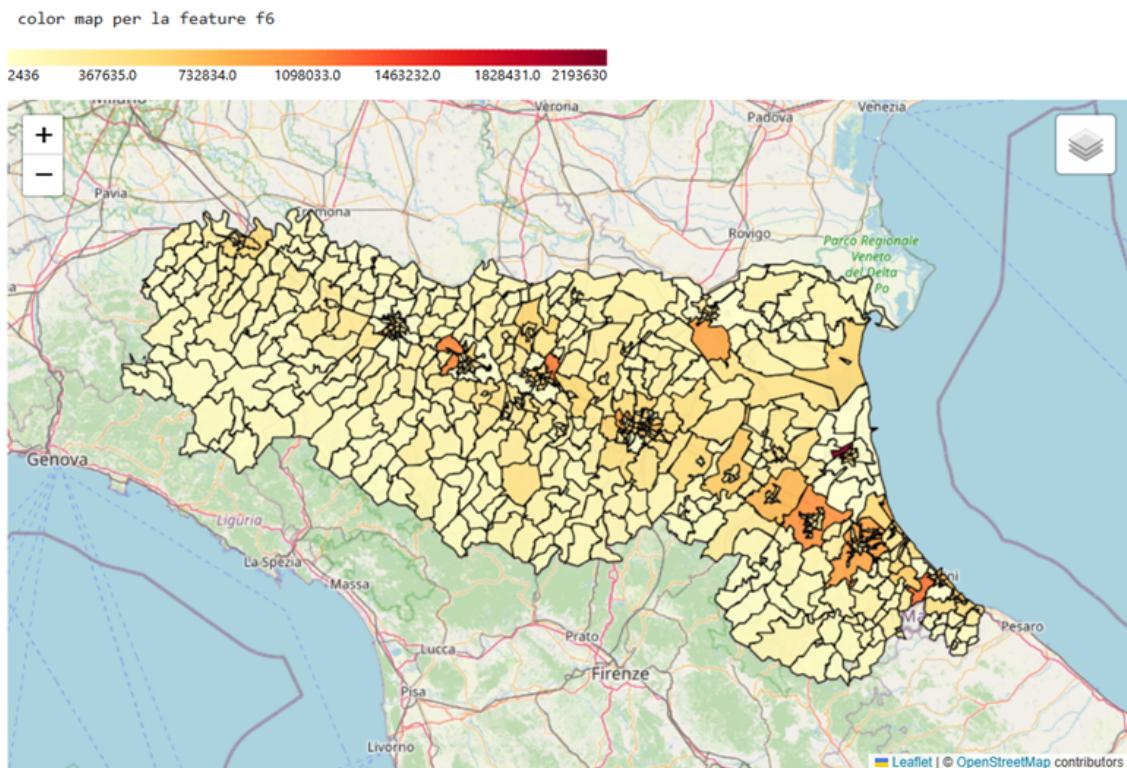


Figura 2.6: Rappresentazione partenze degli over 60

# Capitolo 3

## Analisi dei Flussi

In questo capitolo si analizzano i flussi di mobilità tra aree censuarie elementari (ACE) della regione Emilia-Romagna, attraverso la **matrice OD** (Origin/Destination), una rappresentazione che quantifica gli spostamenti da ogni area di origine verso le possibili destinazioni. L'obiettivo principale è comprendere l'intensità, la direzione degli spostamenti e l'individuazione dei principali nodi di partenza e arrivo.

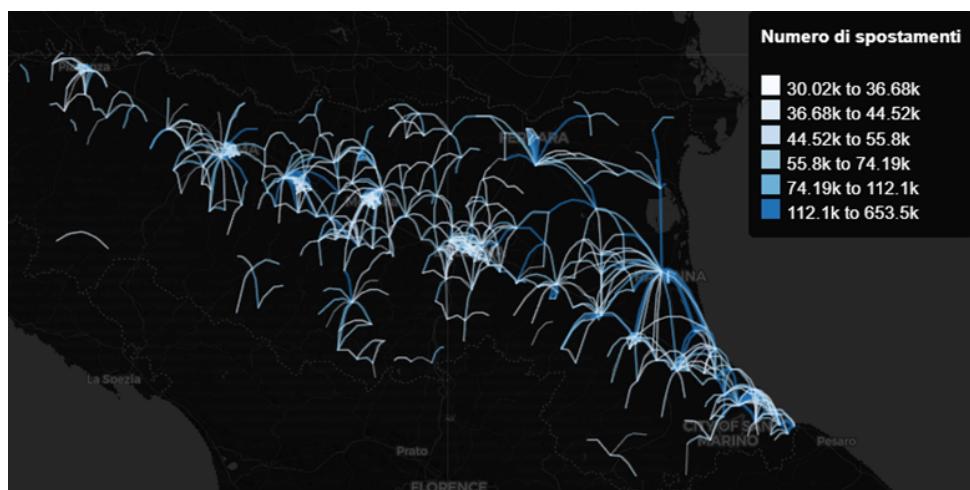


Figura 3.1: Rappresentazione matrice Origin-Destination

La visualizzazione dei flussi è stata realizzata mediante una **flow map** interattiva sviluppata con Folium. Le curve congiungono i centroidi delle aree di origine e destinazione, utilizzando colore e spessore per rappresentare proporzionalmente il numero di spostamenti. Per garantire una migliore leggibilità, sono stati inclusi solo i flussi con almeno 30.000 spostamenti, calcolati a partire da join geografici e geometrie in formato GeoJSON.

Il fine dell'analisi è individuare le **tratte più percorse**, al fine di supportare **valutazioni sulla sostenibilità** dei trasporti e pianificare interventi infrastrutturali mirati nelle aree a maggiore mobilità. In particolare, nelle sezioni successive vie-

ne presentato un **approfondimento sull'area di Bologna**, utilizzata come caso studio per valutare potenziali strategie di ottimizzazione della mobilità urbana.

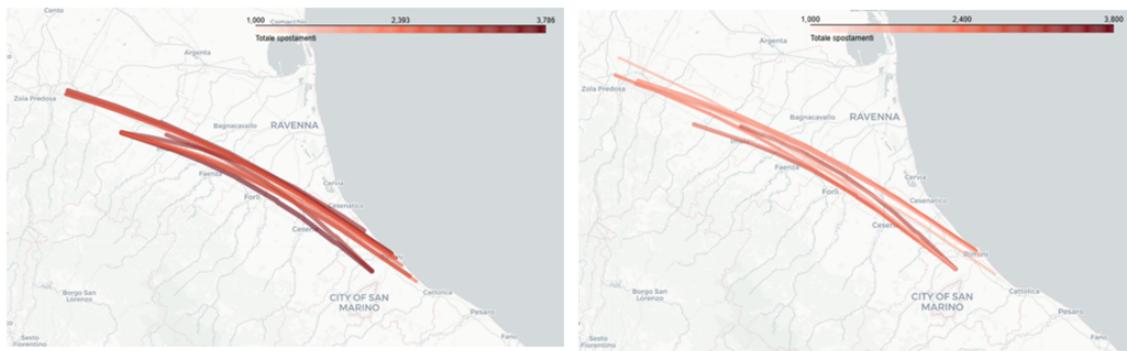


Figura 3.2: Confronto spostamenti agosto-settembre tra provincie di Bologna e Rimini

In questo esempio possiamo osservare la differenza nel numero di spostamenti tra le provincie di Bologna e Rimini nei mesi di agosto e settembre rispettivamente.

### 3.1 Analisi dei flussi di mobilità nell'area di Bologna

Per approfondire l'analisi dei flussi e verificarne le potenzialità applicative in ambito urbano, si propone un caso studio focalizzato sull'area del comune di Bologna. Questa sezione si concentra sui collegamenti interni alla città e tra il comune e il resto della provincia, con l'obiettivo di identificare le tratte più trafficate. Tali informazioni risultano fondamentali per valutare interventi di ottimizzazione della mobilità, supportare decisioni infrastrutturali e migliorare la sostenibilità del sistema di trasporto locale.

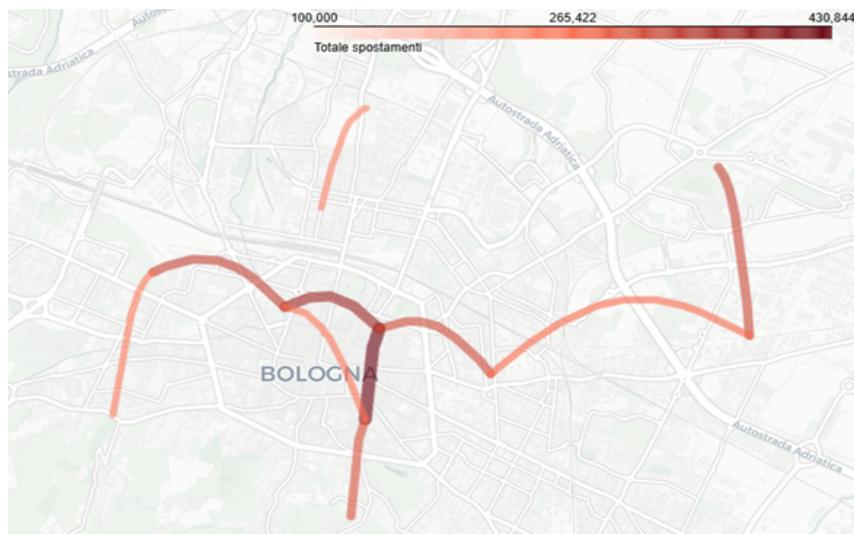


Figura 3.3: Top 10 flussi nel comune di Bologna

La visualizzazione 3.3 mostra i **10 flussi interni più intensi all'interno del comune di Bologna**, ovvero le tratte che registrano il maggior numero di spostamenti totali nel periodo considerato tra coppie di aree censuarie (ACE).

Le tratte maggiormente trafficate si concentrano attorno al centro storico, in particolare nei pressi della Stazione Centrale e di Piazza Maggiore, confermando il ruolo centrale di queste aree nella rete di mobilità urbana. I flussi si dirigono principalmente lungo le direttrici che collegano il centro con le principali vie di accesso alla città, come via San Donato a nord-est, via Andrea Costa a ovest e via Emilia Levante verso sud-est.



Figura 3.4: Top 15 flussi tra comune e provincia di Bologna

L'immagine 3.4 mostra i **15 flussi più intensi tra il comune di Bologna e le aree censuarie (ACE) appartenenti all'intera provincia**. I collegamenti più marcati si distribuiscono in modo radiale, evidenziando come Bologna rappresenti un polo di attrazione centrale per un'ampia fascia del territorio provinciale. Tra i flussi più rilevanti si osservano quelli diretti verso Calderara di Reno, Castel Maggiore, San Lazzaro di Savena, Granarolo dell'Emilia, ma anche tratte più lunghe verso comuni come Pianoro e Sasso Marconi.

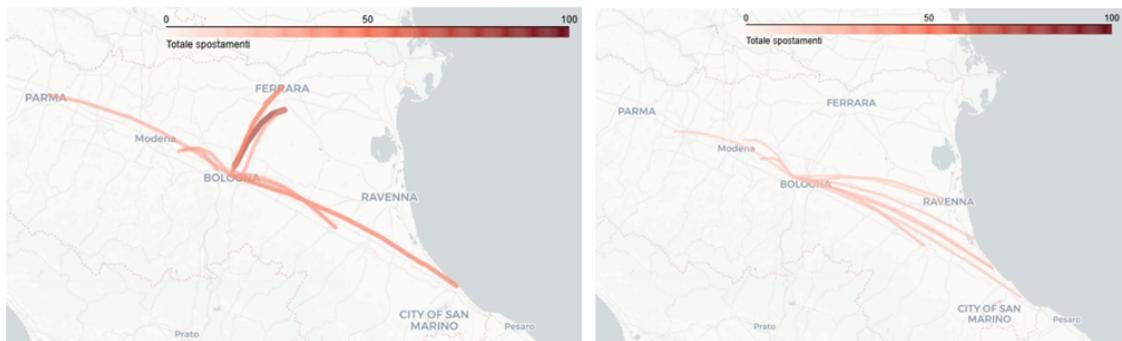


Figura 3.5: Confronto Flussi giorno feriale e festivo

La figura 3.5 mostra un **confronto tra i flussi di mobilità in un giorno feriale (a sinistra) e in uno festivo (a destra)** con riferimento al comune di Bologna e gli ACE fuori dalla provincia di Bologna.

Dal confronto tra i due scenari emerge una chiara differenza nei pattern di mobilità: nei giorni feriali (immagine a sinistra), i flussi sono più concentrati lungo direttrici funzionali che collegano Bologna con centri urbani rilevanti come Modena, Ferrara e la costa romagnola, riflettendo dinamiche tipiche del pendolarismo lavorativo e scolastico. Nei giorni festivi (immagine a destra), invece, i flussi si distribuiscono in modo più omogeneo verso le aree costiere e turistiche, suggerendo una mobilità legata al tempo libero.

## Considerazioni finali

Le analisi presentate in questo caso studio, basate sulla rappresentazione dei flussi più intensi all'interno del comune di Bologna, tra il comune e la provincia, e con l'esterno regionale, offrono uno strumento prezioso per comprendere in modo dettagliato le dinamiche di mobilità urbana e metropolitana.

L'identificazione delle tratte più percorse consente di individuare **punti critici e potenziali aree di intervento** per migliorare la sostenibilità complessiva del sistema di trasporto. Le informazioni ricavate dai flussi rappresentano una base oggettiva per:

- progettare **soluzioni infrastrutturali mirate**, ad esempio lungo i corridoi di maggiore congestione;
- valutare l'**adeguatezza dell'offerta di trasporto pubblico** in rapporto alla domanda effettiva;
- promuovere forme di **mobilità alternativa**, come ciclabilità e intermodalità;
- supportare **strategie di mobilità intelligente**, in grado di adattarsi alle variazioni settimanali e stagionali degli spostamenti.

Queste analisi possono guidare azioni concrete per **ridurre la dipendenza dall'auto privata**, contenere le emissioni e migliorare l'accessibilità urbana.

# Capitolo 4

## Event Detection

In questo capitolo viene presentato un algoritmo sviluppato per individuare **anomalie nei flussi di mobilità** tra aree censuarie (ACE), con l'obiettivo di rilevare possibili eventi straordinari o variazioni inattese nel comportamento degli spostamenti.

L'approccio si basa sull'uso dello *Z-score* [3], una misura statistica che consente di valutare quanto un valore si discosti dalla media storica dei flussi per una determinata ACE. La formula utilizzata è:

$$Z_{i,t} = \frac{FlowNet_{i,t} - \mu_i}{\sigma_i} \quad (4.1)$$

Dove:

- $FlowNet_{i,t}$  è il numero di spostamenti netti osservati nell'ACE  $i$  al tempo  $t$
- $\mu_i$  e  $\sigma_i$  sono rispettivamente la media e la deviazione standard dei flussi storici dell'ACE  $i$

Un valore assoluto di Z-score superiore a **2** è considerato potenzialmente anomalo e può indicare la presenza di un **evento rilevante**, come manifestazioni, festività, blocchi del traffico o altri fattori che alterano i normali pattern di mobilità.

Nelle sezioni successive vengono mostrati alcuni *casi studio* in cui l'algoritmo ha evidenziato anomalie significative nei dati, fornendo un utile strumento per l'analisi dinamica e reattiva dei flussi urbani.

### 4.1 Comune di Bologna

In questa sezione viene analizzato il comportamento dei flussi di mobilità nel Comune di Bologna, con particolare attenzione all'identificazione di anomalie significative attraverso l'algoritmo di *event detection*. L'obiettivo è evidenziare come eventi specifici – quali festività, manifestazioni o fiere – possano influenzare in modo rilevante la dinamica degli spostamenti all'interno del tessuto urbano.

Attraverso l'analisi del *flow net* orario e del relativo *Z-score*, vengono mostrati i principali picchi anomali registrati nel periodo osservato, mettendo in luce le correlazioni tra i dati di mobilità e gli eventi reali avvenuti in città.

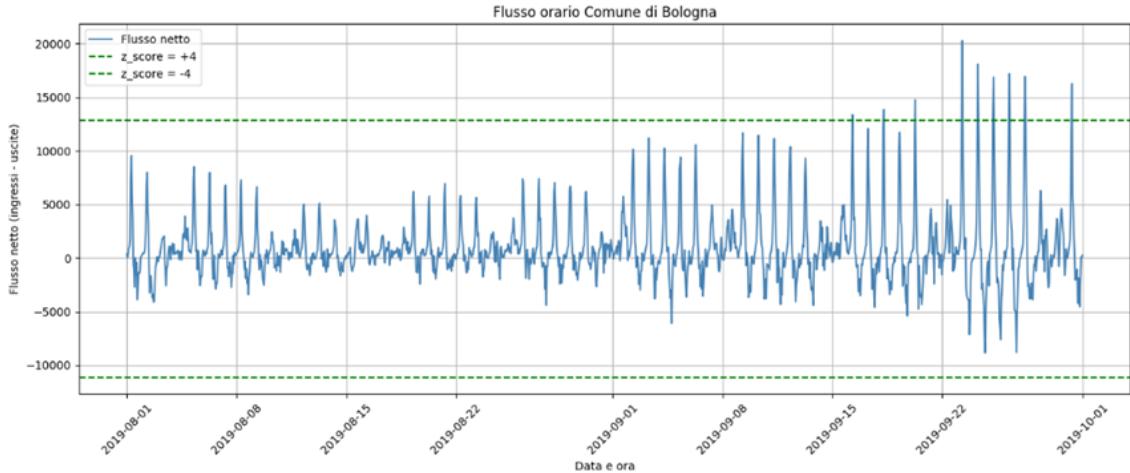


Figura 4.1: Flusso orario comune di Bologna con z-score

Applicando l'algoritmo al *flusso orario* nel comune di Bologna (Figura 4.1) e considerando solo anomalie molto rilevanti ( $|z\text{-score}| > 4$ ), è possibile osservare, a partire dal 16 settembre, un primo aumento del *flusso netto in ingresso*, corrispondente al periodo di rientro dalle vacanze estive e alla ripresa delle attività lavorative e scolastiche.

Una seconda, e più marcata, serie di anomalie è stata rilevata a partire dal 23 settembre, in concomitanza con l'avvio del **Cersaie**, la fiera internazionale della ceramica per l'architettura e l'arredobagno, che richiama ogni anno decine di migliaia di visitatori da tutta Italia e dall'estero.

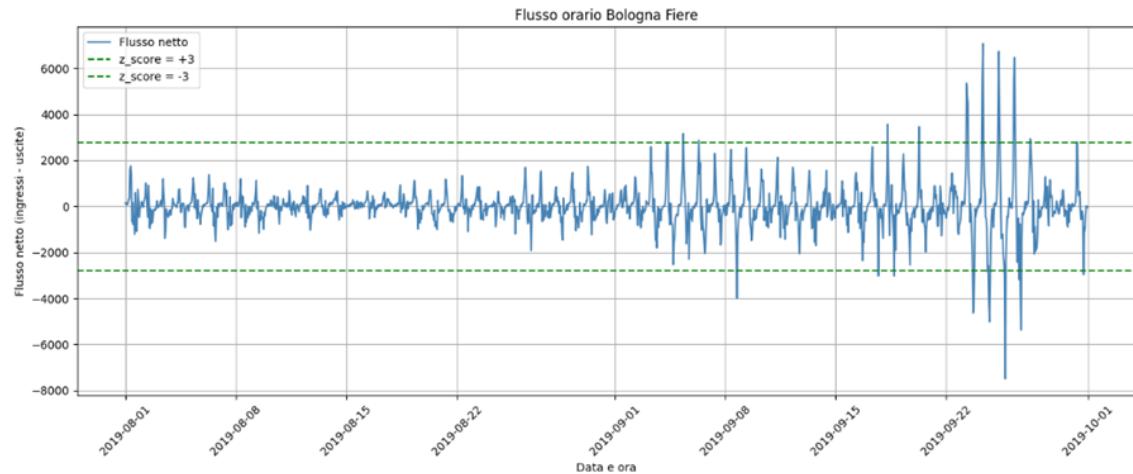


Figura 4.2: Flusso orario ACE Bologna fiere con z-score

Difatti, possiamo osservare chiaramente l'effetto del **Cersaie** anche analizzando

singolarmente l'*area censuaria* (ACE) corrispondente alla *Fiera di Bologna* (Figura 4.2), e rilevare le conseguenze anche sull'area della *stazione di Bologna* (Figura 4.3), che rappresenta uno dei principali snodi di accesso alla città.

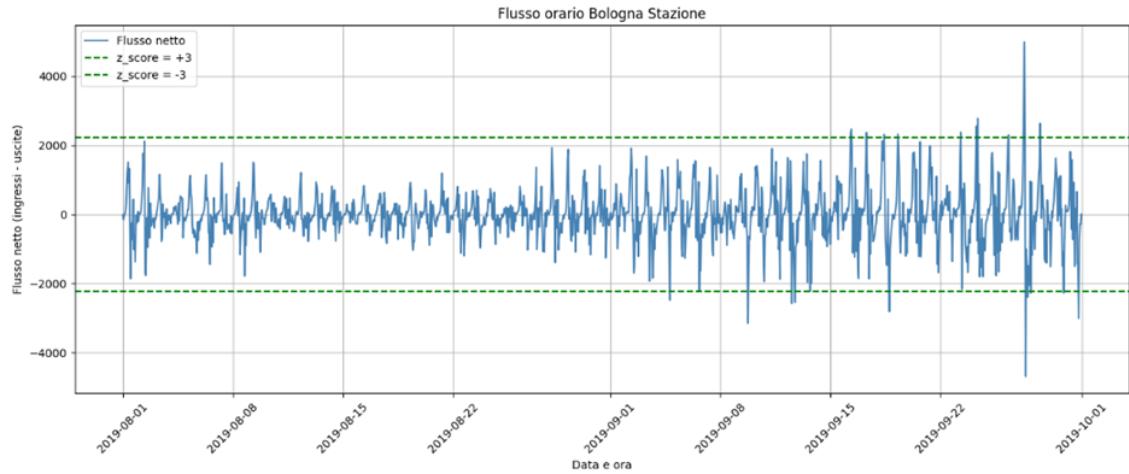


Figura 4.3: Flusso orario ACE Bologna stazione con z-score

Possiamo osservare gli effetti del **Cersaie** anche nei comuni della provincia di Bologna (Figura 4.4). Infatti, come evidenziato nel grafico relativo al flusso netto orario del comune di **Dozza**, si registrano picchi anomali corrispondenti ai giorni della fiera. Questo suggerisce che l'evento abbia generato un impatto significativo non solo sul capoluogo, ma anche su centri limitrofi.

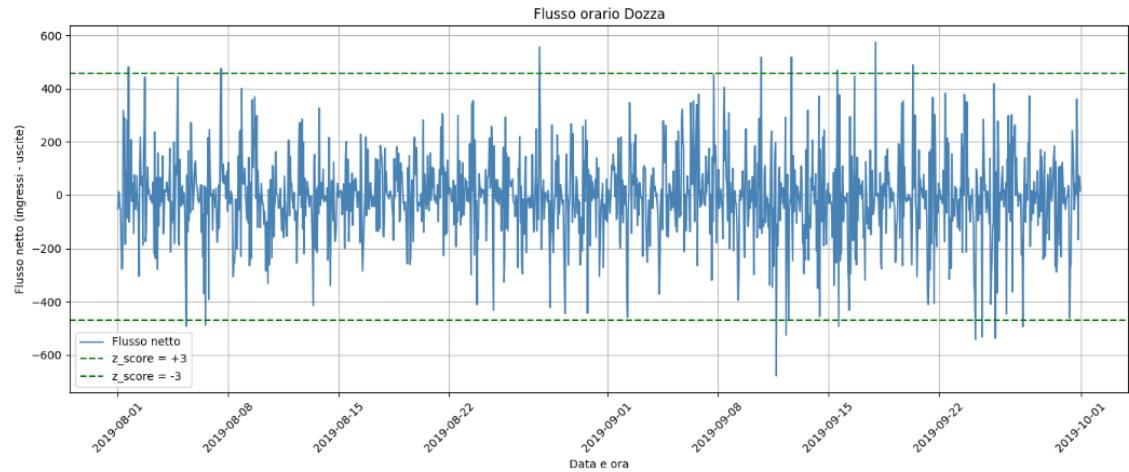


Figura 4.4: Flusso orario ACE Comune di Dozza con z-score

## 4.2 Stadio Bologna

Un altro esempio evidente di *rilevamento anomalo* si osserva nell'area dello **stadio di Bologna**, dove l'algoritmo ha individuato *picchi netti di flusso* in corrispondenza

delle partite casalinghe del **Bologna FC**.

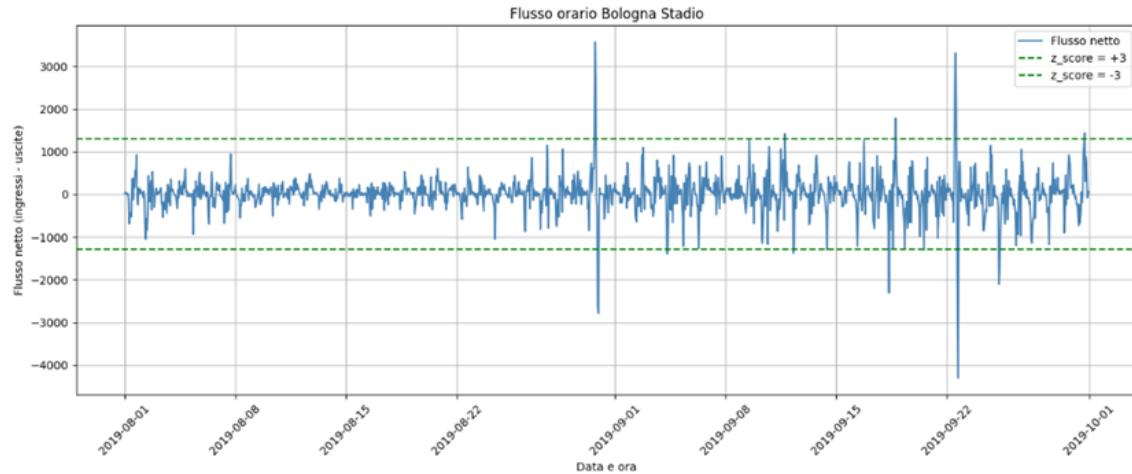


Figura 4.5: Flusso orario ACE Bologna stadio con z-score

### 4.3 Autodromo di Imola

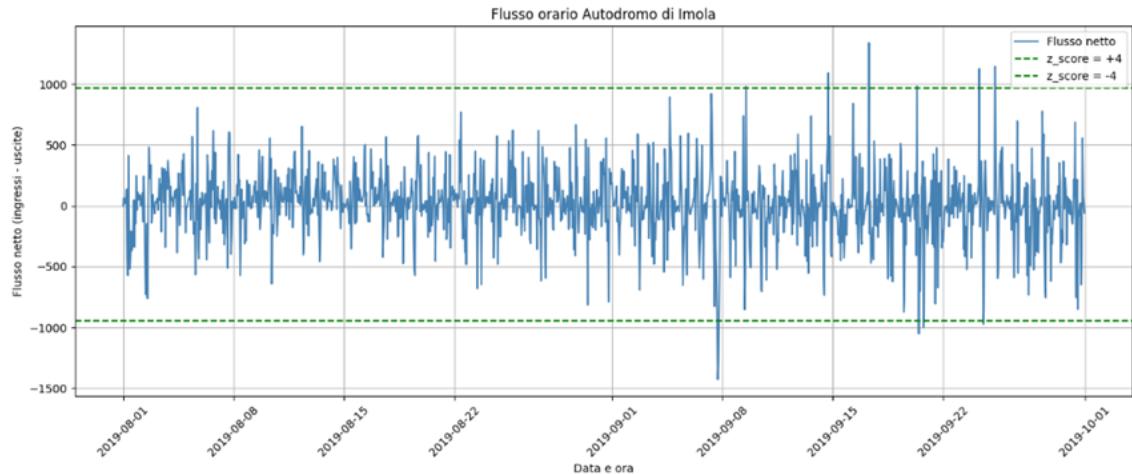


Figura 4.6: Flusso orario ACE autodromo Imola con z-score

L'analisi del *flusso netto orario* nell'area dell'**Autodromo di Imola** ha evidenziato diversi *picchi anomali* nel mese di settembre 2019, che coincidono con alcuni eventi ad alta affluenza svolti nella struttura.

In particolare, il 7 settembre si registra un *flusso negativo marcato*, compatibile con la **Mostra Scambio CRAME** (6–8 settembre), una delle più grandi fiere europee dedicate a veicoli d'epoca e pezzi di ricambio, che ha richiamato oltre 2.000 espositori e decine di migliaia di visitatori. L'elevato numero di uscite registrate quel giorno (alle ore 17:00) suggerisce l'effetto del deflusso al termine della manifestazione.

Un altro picco rilevante si osserva attorno al 20–22 settembre, periodo in cui si è tenuto il **Summer Food Experience**, evento enogastronomico con intrattenimento. Questo potrebbe aver contribuito a un *aumento degli ingressi* nella zona durante quel weekend.

## 4.4 Comune di Ferrara

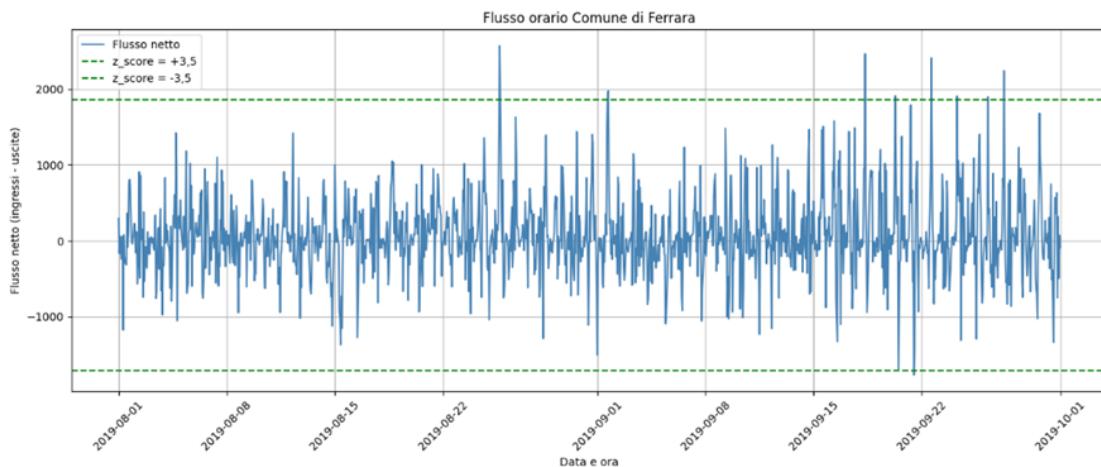


Figura 4.7: Flusso orario ACE comune di Ferrara con z-score

L’analisi dei *flussi netti orari* per il **Comune di Ferrara** evidenzia diversi *picchi anomali* in corrispondenza di eventi di rilievo che si sono svolti tra agosto e settembre 2019.

In particolare, il **Ferrara Buskers Festival** (22 agosto – 1 settembre), uno dei festival internazionali più noti dedicati alla musica di strada, ha generato *flussi positivi significativi*, soprattutto nei fine settimana del 25 agosto e 1 settembre. In queste date, l’afflusso verso il centro città ha raggiunto valori fuori soglia, con migliaia di ingressi in più rispetto alle uscite.

Un ulteriore aumento dei flussi è stato osservato tra il 18 e il 20 settembre, in corrispondenza del **RemTech Expo**, evento fieristico nazionale dedicato alla bonifica ambientale e alla riqualificazione del territorio.

Infine, tra il 15 e il 26 settembre, si registrano ulteriori anomalie positive riconducibili alla **Ferrara Chamber Academy**. Le attività musicali diffuse in tutta la città, tra cui concerti e laboratori, hanno prodotto un *afflusso costante* di pubblico.

## 4.5 Comune di Modena

Nel Comune di Modena, l’algoritmo di *event detection* ha rilevato numerosi picchi anomali nei flussi netti orari durante il mese di settembre 2019, con valori di Z-score superiori a  $\pm 3.5$ . Queste anomalie risultano in forte correlazione con eventi locali

rilevanti, come evidenziato nel confronto tra i dati di mobilità e il calendario delle manifestazioni cittadine.

In particolare, un flusso negativo significativo è stato registrato il **4 settembre**, probabilmente legato alla conclusione delle vacanze estive e al ritorno alla routine quotidiana. Nei giorni **10–11 settembre**, si osserva un incremento degli ingressi compatibile con l'inizio dell'anno scolastico.

Tra il **16** e il **18 settembre**, si registrano picchi coerenti con il *Festivalfilosofia*, che richiama ogni anno un vasto pubblico. Un ulteriore aumento notevole è rilevato nei giorni **21–22 settembre**, in concomitanza con il *Modena Motor Gallery*.

Infine, il **30 settembre** rappresenta il valore massimo di anomalia osservata nel periodo, con un flusso netto pari a 8.614 persone e uno Z-score di **5.91**, che potrebbe essere legato al rientro post-weekend o ad attività legate alla fine del mese.

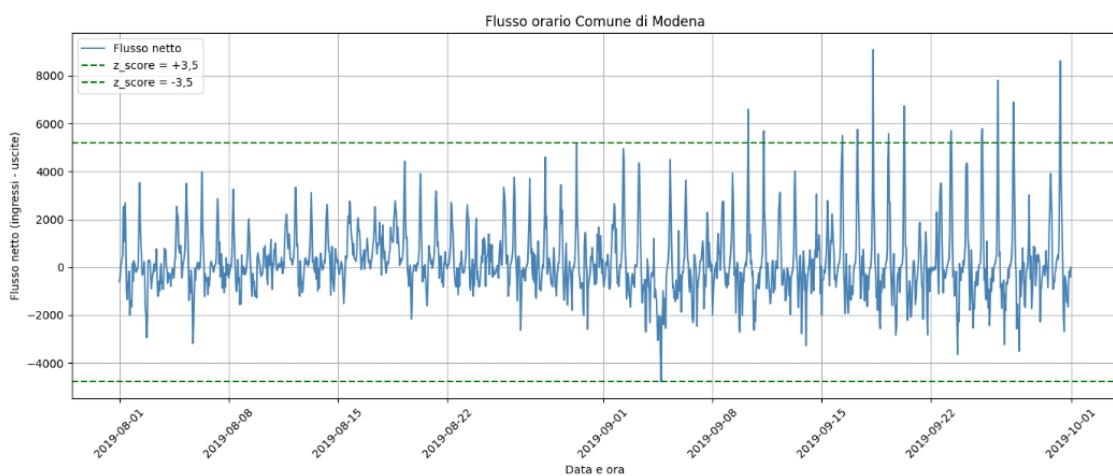


Figura 4.8: Flusso orario ACE comune di Modena con z-score

## Considerazioni finali

I risultati ottenuti attraverso l'applicazione dell'algoritmo di *event detection* mostrano come lo Z-score sia uno strumento efficace per individuare anomalie nei flussi di mobilità urbana. L'analisi ha evidenziato picchi significativi in corrispondenza di eventi locali rilevanti, permettendo di correlare direttamente i dati di mobilità con fenomeni reali sul territorio.

È importante sottolineare che le valutazioni sono state condotte su un periodo temporale limitato (agosto–settembre 2019), a causa della disponibilità parziale del dataset. Inoltre, le aree censuarie (ACE) analizzate sono state selezionate come esempi rappresentativi, utili a illustrare la validità e l'applicabilità del metodo, ma non esaustivi dell'intero contesto regionale.

## Capitolo 5

# Previsione flussi di mobilità con Gradient Boosting

In questo capitolo vengono presentati i risultati delle analisi predittive volte a stimare il **numero di persone in ingresso** nelle diverse aree censuarie elementari (ACE), utilizzando modelli di apprendimento automatico implementati tramite la libreria `scikit-learn`. L'obiettivo principale è valutare l'efficacia di tali modelli nel prevedere i flussi futuri sulla base dei dati storici disponibili, prendendo in considerazione diverse configurazioni di variabili.

In particolare è stato applicato il modello *Gradient Boosting Regressor* [2], scelto per la sua capacità di cogliere relazioni non lineari complesse e gestire in modo efficiente variabili eterogenee. I risultati ottenuti permettono di esplorare le potenzialità predittive dello strumento sviluppato e di identificare le configurazioni più performanti.

Inizialmente si è lavorato con l'obiettivo di stimare il numero di persone in ingresso in una singola area censuaria (ACE), definita come *ACE target*, adottando due diversi approcci modellistici:

- Il primo approccio considera esclusivamente i dati relativi all'ACE target, utilizzando solo le informazioni locali per la previsione dei flussi giornalieri in ingresso;
- Il secondo approccio utilizza l'intero dataset regionale, includendo i dati provenienti da tutte le ACE, al fine di valutare se una visione più ampia e globale della rete di mobilità possa migliorare le prestazioni predittive.

Successivamente, si è passati a una generalizzazione del modello, estendendo l'analisi a tutte le ACE della regione Emilia-Romagna. Tale estensione ha permesso di valutare l'efficacia del metodo su scala regionale e di confrontare le prestazioni in contesti territoriali eterogenei.

Verranno inoltre confrontati i risultati ottenuti in relazione a diverse configurazioni di feature, al fine di individuare le variabili più rilevanti nella determinazione

degli spostamenti in ingresso. L’analisi fornisce così una base quantitativa utile per applicazioni future legate alla pianificazione urbana e alla gestione dei flussi di mobilità.

## 5.1 Librerie e strumenti utilizzati

Per lo sviluppo del modello predittivo sono stati utilizzati diversi strumenti della libreria `scikit-learn` [1], uno dei principali framework open-source per il machine learning in Python. Questa libreria offre una vasta gamma di algoritmi per la classificazione, regressione, clustering e riduzione della dimensionalità, oltre a strumenti per la selezione delle feature, la validazione incrociata e la gestione dei dati.

Il modello adottato è il *Gradient Boosting Regressor*, un algoritmo di regressione basato sull’approccio dell’*ensemble learning*. In particolare, il metodo si fonda sull’idea di costruire una sequenza di modelli deboli (solitamente alberi di decisione a profondità limitata), ciascuno dei quali corregge gli errori commessi dai modelli precedenti. La procedura si basa sulla tecnica della discesa del gradiente: ad ogni iterazione viene calcolato il residuo, cioè la differenza tra la previsione corrente e il valore reale, e un nuovo albero viene addestrato per minimizzare tale residuo. Questo processo iterativo consente di migliorare progressivamente la precisione del modello finale, rendendolo particolarmente adatto a catturare relazioni complesse e non lineari tra le variabili.

Poiché il Gradient Boosting richiede la definizione di diversi parametri di controllo, è stata utilizzata `GridSearchCV` per l’ottimizzazione automatica degli *iperparametri*, cioè quei parametri che non vengono appresi dal modello ma devono essere impostati manualmente prima dell’addestramento. Tra gli iperparametri principali figurano il numero di stimatori (`n_estimators`), la profondità massima degli alberi (`max_depth`) e il tasso di apprendimento (`learning_rate`). `GridSearchCV` consente di esplorare in modo esaustivo tutte le combinazioni possibili di tali valori, valutando le prestazioni del modello su un set di validazione per selezionare la configurazione ottimale. La corretta scelta degli iperparametri è cruciale per ottenere un buon bilanciamento tra accuratezza e capacità di generalizzazione del modello.

Infine, per garantire la corretta scalabilità delle feature in ingresso, è stata utilizzata `MinMaxScaler`, una tecnica di normalizzazione che trasforma ogni variabile in un intervallo predefinito (tipicamente  $[0,1]$ ). Questa operazione risulta particolarmente utile quando si impiegano modelli sensibili alla scala dei dati, contribuendo a migliorare la stabilità e l’efficienza dell’addestramento.

## 5.2 Predizione dei flussi con dati locali

Come primo approccio per stimare la variabile `datavalue`, ovvero il numero di persone che si spostano tra le diverse aree, è stato scelto di utilizzare solamente i dati che hanno come `toid` (ovvero come ACE di arrivo) *l’ACE target*, raggruppandoli

per giorno. In questo modo è stato costruito un dataset di 61 righe, ciascuna rappresentante il numero totale di ingressi nell'ACE considerato per un determinato giorno.

Come ACE target è stato selezionato il comune di **Dozza (BO)**. La scelta è motivata da diverse considerazioni:

- Dozza è un piccolo borgo medievale di interesse turistico, noto per il suo centro storico, i murales e la rocca sforzesca, che attira flussi di visitatori variabili in occasione di eventi, nei fine settimana e durante le festività;
- la dimensione contenuta dell'area e la semplicità della rete di accessi permettono di isolare più facilmente le dinamiche di ingresso, rendendo il contesto ideale per un'analisi predittiva;
- questa configurazione consente, in prospettiva, l'integrazione di dati esterni (come condizioni meteorologiche o calendari di eventi locali), permettendo di approfondire l'influenza di fattori più specifici sulla mobilità.

## Feature

Di seguito si riportano le feature testate per la costruzione del modello predittivo:

- **date**: la data in cui si è verificato lo spostamento;
- **weekday**: il giorno della settimana [0 = lunedì, ..., 6 = domenica];
- **week**: la settimana dell'anno (intervallo [0,9] per il periodo analizzato);
- **weekend**: variabile booleana che indica se il giorno è un sabato o una domenica;
- **month**: mese di riferimento (0 = agosto, 1 = settembre);
- **festivo**: flag che assume valore attivo nei weekend e nelle due settimane centrali di agosto;
- **lag\_j**: numero di persone in ingresso  $j$  giorni prima;
- **fenomeni**: flag che indica la presenza di fenomeni atmosferici rilevanti (es. pioggia);
- **evento**: flag che rileva la presenza di eventi nel comune di Dozza;
- **bologna**: numero di ingressi registrati il giorno precedente nel comune di Bologna;
- **imola**: numero di ingressi registrati il giorno precedente nel comune di Imola.

	date	datavalue	weekday	week	weekend	festivo	month	lag_1	lag_2	lag_3	lag_7	fenomeni	evento	bologna	imola
0	2019-08-01	10241	3	0	0	0	0	0.0	0.0	0.0	0.0	0	0	0.0	0.0
1	2019-08-02	9752	4	0	0	0	0	10241.0	0.0	0.0	0.0	1	0	562915.0	85571.0
2	2019-08-03	9188	5	0	1	1	0	9752.0	10241.0	0.0	0.0	0	0	524811.0	81407.0
3	2019-08-04	7757	6	0	1	1	0	9188.0	9752.0	10241.0	0.0	0	0	405302.0	70136.0
4	2019-08-05	8431	0	1	0	1	0	7757.0	9188.0	9752.0	0.0	0	0	313369.0	55002.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56	2019-09-26	10703	3	8	0	0	1	9204.0	10194.0	9868.0	9480.0	1	0	807756.0	96977.0
57	2019-09-27	9880	4	8	0	0	1	10703.0	9204.0	10194.0	10697.0	0	0	801625.0	97325.0
58	2019-09-28	10000	5	8	1	1	1	9880.0	10703.0	9204.0	9616.0	0	0	798166.0	92172.0
59	2019-09-29	9900	6	8	1	1	1	10000.0	9880.0	10703.0	9232.0	0	0	679888.0	85540.0
60	2019-09-30	10096	0	9	0	0	1	9900.0	10000.0	9880.0	9868.0	0	0	547388.0	68718.0

61 rows × 15 columns

Figura 5.1: Dataset con feature per analisi su Dozza

## Modello di regressione

La feature `date` è stata convertita in formato numerico per poter essere utilizzata correttamente dal modello di regressione. Successivamente, il dataset è stato suddiviso in due sottoinsiemi: *training set* (70%) e *test set* (30%), variando il seed casuale per ottenere valutazioni più robuste e generalizzabili.

Non si è scelto di utilizzare simultaneamente tutte le feature disponibili, poiché l'inclusione di un numero eccessivo di variabili avrebbe potuto introdurre rumore, riducendo la capacità predittiva del modello. Le feature sono state quindi testate in gruppi, al fine di individuare le combinazioni più efficaci.

Per valutare la stabilità e l'accuratezza del modello, è stato eseguito un ciclo di addestramento e test su diversi seed casuali. Per ciascuna esecuzione sono stati calcolati due indicatori di errore:

- **MAE** (Mean Absolute Error): errore assoluto medio;
- **MAPE** (Mean Absolute Percentage Error): errore percentuale assoluto medio.

I dati in ingresso sono stati normalizzati tramite `MinMaxScaler`, mentre i valori target sono stati trasformati utilizzando il logaritmo naturale, al fine di ridurre l'impatto dell'elevata asimmetria nella distribuzione.

L'algoritmo scelto per la previsione è stato il *Gradient Boosting Regressor*, ottimizzato tramite ricerca iperparametrica con `GridSearchCV`. I valori esplorati per ciascun iperparametro sono stati i seguenti:

- `n_estimators`: [100, 300, 500];
- `learning_rate`: [0.01, 0.1, 0.2];
- `max_depth`: [3, 5, 7].

Infine, le predizioni ottenute sono state riconvertite alla scala originale tramite l'inverso della trasformazione logaritmica e confrontate con i valori osservati, al fine di calcolare le metriche finali. Tutti i risultati sono stati salvati e analizzati per identificare le configurazioni più performanti.

## Risultati

FEATURES	METRIC	SEED										MEAN
		0	7	13	21	42	99	123	34	67	80	
date	MAE	590	713	643	663	808	802	644	733	718	663	698
	MAPE (%)	6.803	9.492	7.642	8.312	9.420	9.828	7.484	8.306	7.824	7.483	8.259
date, weekday, week, weekend	MAE	667	747	873	637	777	832	683	695	786	696	739
	MAPE (%)	7.407	9.390	10.301	7.601	8.849	10.254	7.904	7.763	8.516	7.685	8.567
date, fenomeni, evento	MAE	596	740	629	632	880	807	823	781	752	721	736
	MAPE (%)	6.849	9.742	7.455	7.914	10.155	9.908	8.810	8.891	8.540	8.151	8.741
date, bologna, imola	MAE	601	663	762	768	986	793	830	663	827	519	718
	MAPE (%)	7.255	8.845	8.931	9.449	12.341	10.397	9.803	7.367	9.319	5.694	8.940
date, festivo, month	MAE	569	712	615	669	756	804	642	739	705	648	686
	MAPE (%)	6.565	9.462	7.360	8.374	8.779	9.864	7.459	8.287	7.697	7.332	8.117
date, lag	MAE	717	977	762	719	835	1029	912	925	917	714	850
	MAPE (%)	8.649	12.824	9.050	8.940	9.428	12.750	11.176	10.172	10.222	8.037	10.125
date, weekday, week, festivo, month	MAE	655	732	833	596	797	839	672	722	763	702	732
	MAPE (%)	7.401	9.211	9.900	7.116	9.089	10.322	7.862	8.054	8.263	7.769	8.498
date, festivo	MAE	569	712	615	663	754	803	640	739	705	649	684
	MAPE (%)	6.569	9.462	7.360	8.320	8.744	9.857	7.442	8.283	7.693	7.345	8.107

Figura 5.2: Risultati confronto feature

I risultati ottenuti dall'applicazione del modello *Gradient Boosting Regressor* per la previsione dei flussi di mobilità nel comune di Dozza (BO), valutati utilizzando le metriche **MAE** (Mean Absolute Error) e **MAPE** (Mean Absolute Percentage Error), hanno evidenziato un buon livello di accuratezza del modello, con errori assoluti medi contenuti e percentuali di errore relative basse.

L'analisi comparativa tra i diversi gruppi di feature ha mostrato che, anche utilizzando unicamente la variabile **date**, il modello è stato in grado di ottenere risultati già molto soddisfacenti, con valori medi di MAE e MAPE competitivi rispetto a combinazioni più articolate.

L'aggiunta di feature come **evento**, **fenomeni** e valori di tipo **lag** (ritardi temporali), contrariamente alle aspettative, ha spesso introdotto rumore nei dati, portando a un peggioramento delle prestazioni predittive.

Un miglioramento più evidente è stato osservato invece con l'inserimento della feature **festivo**, probabilmente perché, nel periodo analizzato (agosto-settembre), i flussi di mobilità sono fortemente influenzati dalle ferie estive e dalle ricorrenze nazionali, che incidono in modo significativo sulla presenza o meno di persone nel territorio.

## 5.3 Utilizzo del dataset completo per la stima dei flussi

Come **secondo approccio**, si è scelto di utilizzare **l'intero dataset**, anziché limitarsi ai soli dati relativi all'ACE target. In questo modo, il numero di osservazioni è passato da 61 a **28.548 righe**, ottenute combinando i dati giornalieri per tutte le 468 ACE disponibili (61 giorni × 468 ACE).

	<b>date</b>	<b>toid</b>	<b>datavalue</b>	<b>weekday</b>	<b>week</b>	<b>weekend</b>	<b>month</b>	<b>festivo</b>	<b>lag_1</b>	<b>lag_2</b>	<b>lag_3</b>	<b>lag_7</b>
<b>0</b>	2019-08-01	08 033 001 000 000	3649	3	0	0	0	0	0.0	0.0	0.0	0.0
<b>1</b>	2019-08-01	08 033 002 000 000	7193	3	0	0	0	0	0.0	0.0	0.0	0.0
<b>2</b>	2019-08-01	08 033 003 000 000	2245	3	0	0	0	0	0.0	0.0	0.0	0.0
<b>3</b>	2019-08-01	08 033 004 000 000	6263	3	0	0	0	0	0.0	0.0	0.0	0.0
<b>4</b>	2019-08-01	08 033 005 000 000	3571	3	0	0	0	0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
<b>28543</b>	2019-09-30	08 099 024 000 000	4078	0	9	0	1	0	4863.0	4672.0	3431.0	3181.0
<b>28544</b>	2019-09-30	08 099 025 000 000	10383	0	9	0	1	0	10122.0	10054.0	9097.0	8889.0
<b>28545</b>	2019-09-30	08 099 026 000 000	2919	0	9	0	1	0	4304.0	3761.0	3149.0	2776.0
<b>28546</b>	2019-09-30	08 099 027 000 000	2315	0	9	0	1	0	2293.0	2454.0	2255.0	2187.0
<b>28547</b>	2019-09-30	08 099 999 000 255	55142	0	9	0	1	0	44427.0	48376.0	56540.0	47501.0

28548 rows × 12 columns

Figura 5.3: Dataset che ricopre tutti gli ACE

La suddivisione in training e test è stata eseguita a livello di giorno, mantenendo il 70% delle giornate per l'addestramento e il 30% per il test. Tuttavia, ai fini della valutazione, **sono stati considerati solo i dati del test set relativi all'ACE target**, per mantenere la coerenza con l'obiettivo di previsione. Per tenere traccia dell'identità spaziale di ciascun record, è stata aggiunta al dataset la feature **toid**, convertita poi in formato numerico, così da permettere al modello di distinguere le diverse aree censuarie. Anche in questo caso, è stato utilizzato il modello di Gradient Boosting già descritto in precedenza, testando diversi gruppi di feature per valutarne l'impatto sulle prestazioni predittive.

## Risultati

FEATURES	METRIC	SEED										MEAN
		0	7	13	21	42	99	123	34	67	80	
date	MAE	583	618	732	652	797	899	687	882	598	546	699
	MAPE (%)	6.726	7.708	8.237	8.211	9.584	10.593	8.434	9.324	6.936	6.197	8.342
date, weekday, week, weekend	MAE	779	506	762	655	807	732	681	654	730	771	708
	MAPE (%)	8.718	6.355	8.686	7.886	9.076	9.038	8.272	7.257	7.961	8.600	8.297
date, festivo, month	MAE	678	605	590	652	694	852	691	730	778	652	683
	MAPE (%)	7.955	7.394	6.895	6.652	8.840	10.687	8.551	8.197	8.832	7.226	8.062
date, lag	MAE	768	674	1079	744	1289	851	1331	1338	1522	696	1032
	MAPE (%)	8.965	8.472	11.604	8.742	14.523	10.450	14.345	14.553	16.300	7.645	11.604
date, weekday, week, festivo, month	MAE	682	557	697	591	828	754	691	660	673	656	675
	MAPE (%)	7.906	6.837	8.142	6.801	9.725	9.240	8.305	7.315	7.459	7.489	8.041
date, festivo	MAE	672	598	590	539	832	850	710	786	777	662	702
	MAPE (%)	7.896	7.319	6.895	6.949	9.731	10.667	8.345	8.804	8.805	7.324	8.223

Figura 5.4: Risultati Gradient Boosting su Dozza secondo approccio

Anche utilizzando l'intero dataset, i risultati nella previsione del numero di persone in ingresso nell'*ACE target* si sono confermati accurati e stabili, con metriche di errore comparabili o, in alcuni casi, superiori rispetto a quelle ottenute con l'approccio locale.

Alcuni gruppi di feature, come `fenomeni` ed `evento`, sono stati esclusi a causa della mancanza di copertura uniforme su tutte le ACE. Inoltre, le variabili relative a `bologna` e `imola` si sono rivelate ridondanti, poiché l'identità spaziale è già espressa in modo sufficiente dalla feature `toid`.

Dal confronto tra i due approcci è emerso che il miglior risultato globale è stato ottenuto con il set di feature `date`, `weekday`, `week`, `festivo`, `month`, che ha raggiunto un MAE medio pari a 675 e un MAPE dell'8.04%. Lo stesso set, applicato nel contesto locale, aveva fatto registrare un MAE di 732 e un MAPE dell'8.49%, mostrando quindi un chiaro miglioramento in termini di accuratezza nel secondo approccio.

Il peggior risultato, in entrambi i casi, è stato osservato nei set che includevano la feature `lag`, confermando che l'introduzione di valori ritardati ha spesso introdotto rumore nei dati, peggiorando la qualità delle previsioni.

In sintesi, entrambi gli approcci si sono dimostrati efficaci e hanno prodotto risultati soddisfacenti, con buoni livelli di accuratezza nella stima dei flussi in ingresso all'`ACE target`.

## 5.4 Generalizzazione del modello

Infine, si è scelto di estendere il primo approccio locale a tutti gli ACE della regione Emilia-Romagna (468 in totale), con l'obiettivo di individuare quale set di feature risultasse più adatto in un contesto generalizzato.

Per ciascun ACE è stato costruito un sottoinsieme di dati composto da 61 righe, rappresentanti il numero di ingressi giornalieri. Su ciascun sottoinsieme è stato applicato lo stesso processo di modellazione, adottando una suddivisione temporale

del dataset in *training set* (i primi 41 giorni, pari al 70% del totale) e *test set* (i restanti 20 giorni, pari al 30%).

Sono stati confrontati tre differenti set di feature:

- **set1**: date, weekday, weekend, week;
- **set2**: date, festivo, month;
- **set3**: date, lag\_1, lag\_2, lag\_3, lag\_7.

Per ogni ACE è stato calcolato il valore di MAPE (Mean Absolute Percentage Error) corrispondente a ciascun set di feature. Successivamente, è stata calcolata la media dei MAPE su tutti gli ACE per ogni set.

I risultati ottenuti hanno mostrato che il **set3**, basato su feature di tipo *lag temporale*, ha fornito le migliori prestazioni complessive, con un MAPE medio pari al 9.92%, rispetto ai valori di 12.08% per **set1** e 12.47% per **set2**.

## Distribuzione dell'errore

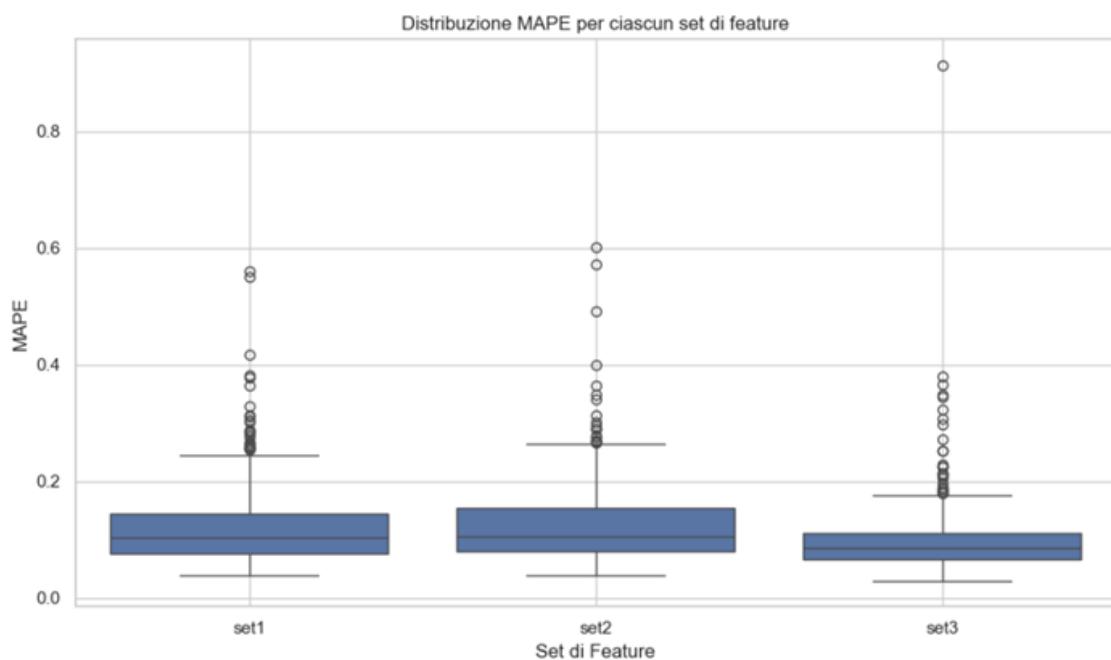


Figura 5.5: Distribuzione Mape nei tre set

La figura mostra la distribuzione del MAPE (Mean Absolute Percentage Error) per ciascun set di feature considerato nell'esperimento di generalizzazione.

Si osserva che il **set3**, basato su feature di tipo *lag temporale*, non solo presenta la media più bassa del MAPE (9.9%), ma evidenzia anche una distribuzione più compatta, con minore variabilità e un numero inferiore di outlier rispetto agli altri

set. Questo risultato suggerisce una maggiore stabilità e affidabilità del modello su ACE differenti.

Al contrario, i set **set1** (basato su variabili temporali standard) e **set2** (contenente informazioni su festività e mese) mostrano una maggiore dispersione nei valori di errore, con alcuni ACE caratterizzati da prestazioni predittive significativamente inferiori.

Nel complesso, l'analisi conferma che l'impiego di feature storiche (**lag**) permette al modello di generalizzare meglio su territori eterogenei, limitando i casi in cui l'errore predittivo supera soglie elevate.

## Affidabilità del modello per ACE

L'analisi dei risultati ottenuti con il set di feature basato sui lag (**set3**) ha permesso di identificare gli ACE in cui il modello risulta più affidabile e quelli in cui, al contrario, incontra maggiori difficoltà predittive.

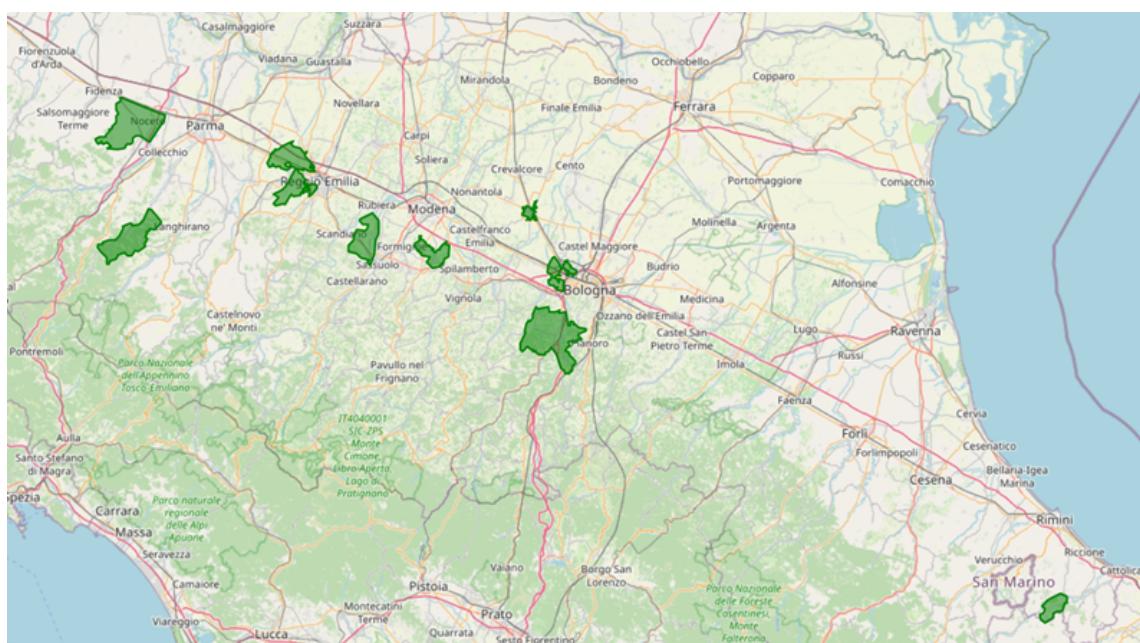


Figura 5.6: ACE più prevedibili

Le **aree più semplici da prevedere**, caratterizzate da un *MAPE molto basso* (evidenziate in verde nella figura), si concentrano prevalentemente in **zone urbane o industriali**, come i comuni di Bologna e Modena. In questi contesti, i flussi di mobilità risultano regolari e ripetitivi, spesso associati ad abitudini lavorative e pendolarismo. La stagionalità settimanale e la memoria dei flussi precedenti costituiscono, in tali casi, elementi altamente predittivi.

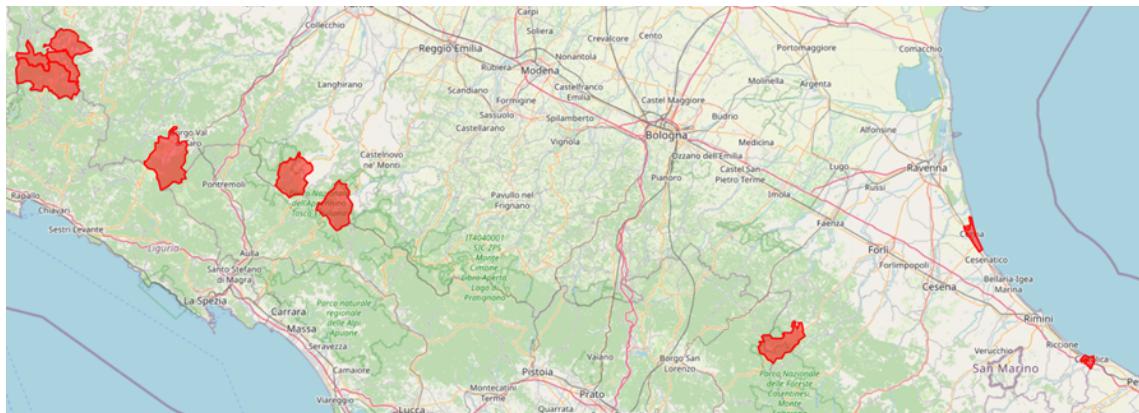


Figura 5.7: ACE meno prevedibili

Al contrario, gli **ACE più difficili** da prevedere (MAPE elevato, evidenziati in rosso) si trovano prevalentemente in **zone turistiche o montane**, come l'Appennino tosco-emiliano, la costa romagnola e l'entroterra di Rimini e Cesena. Qui i flussi di mobilità sono fortemente influenzati da fattori esterni e discontinui, come eventi stagionali, condizioni meteo o vacanze, difficili da catturare con feature puramente temporali o storiche.

Questa distinzione evidenzia come **la natura del territorio influenzi fortemente la predicitività dei flussi**: dove c'è routine (lavoro, scuola, pendolarismo), il modello performa molto bene; dove prevalgono dinamiche legate al turismo o all'intrattenimento, è invece più complesso ottenere previsioni accurate senza integrare ulteriori informazioni contestuali.

# Capitolo 6

## Mobility api

Infine, poiché l’obiettivo finale del progetto è rendere accessibile il modello predittivo attraverso un’interfaccia utilizzabile da altri sistemi o applicazioni, è stata sviluppata un’API per la previsione dei flussi di mobilità in Emilia-Romagna.

È importante sottolineare che la Mobility API rappresenta un **lavoro preliminare**, concepito come **base tecnica per sviluppi futuri**. L’obiettivo di questa componente non è fornire un sistema già pronto per la produzione, ma piuttosto dimostrare la fattibilità dell’integrazione del modello predittivo all’interno di un servizio accessibile via web.

### 6.1 Implementazione con FastAPI e Uvicorn

Per garantire che il modello predittivo sviluppato fosse accessibile e integrabile con altri sistemi, la Mobility API è stata realizzata utilizzando **FastAPI** [7] e **Uvicorn** [8] come server per l’esecuzione del servizio.

FastAPI, un moderno framework web per la costruzione di API in Python, è stato scelto per le sue elevate prestazioni e la sua facilità d’uso. Questo framework offre una serie di vantaggi significativi, tra cui la gestione automatica della validazione dei dati (tramite Pydantic) e la generazione automatica di documentazione interattiva (tramite Swagger UI e ReDoc).

Uvicorn, d’altra parte, è un server ASGI (Asynchronous Server Gateway Interface) estremamente veloce e performante. La sua integrazione con FastAPI consente all’API di gestire un elevato numero di richieste concorrenti in modo efficiente, garantendo reattività e scalabilità al servizio.

In sintesi, l’engine è progettato per ricevere richieste HTTP, elaborarle e fornire previsioni in modo rapido e affidabile, fungendo da interfaccia tra i sistemi client e il modello di machine learning sottostante. Questa architettura robusta e scalabile assicura che l’API possa supportare efficacemente scenari di utilizzo intensivo e diverse applicazioni.

## 6.2 Modello predittivo

Il modello utilizzato è basato sul Gradient Boosting Regressor, ottimizzato secondo la stessa metodologia descritta nei capitoli precedenti: trasformazione logaritmica del target, normalizzazione con MinMaxScaler e ottimizzazione degli iperparametri tramite GridSearchCV. In questo caso, i dati sono stati divisi considerando il 70% dei giorni per l'addestramento e il restante 30% per il test, utilizzando come feature: date, toid, weekday, week e weekend. Questo ha portato a un **MAE di circa 969** e un **MAPE di circa 8.92%**, confermando una buona capacità predittiva del modello.

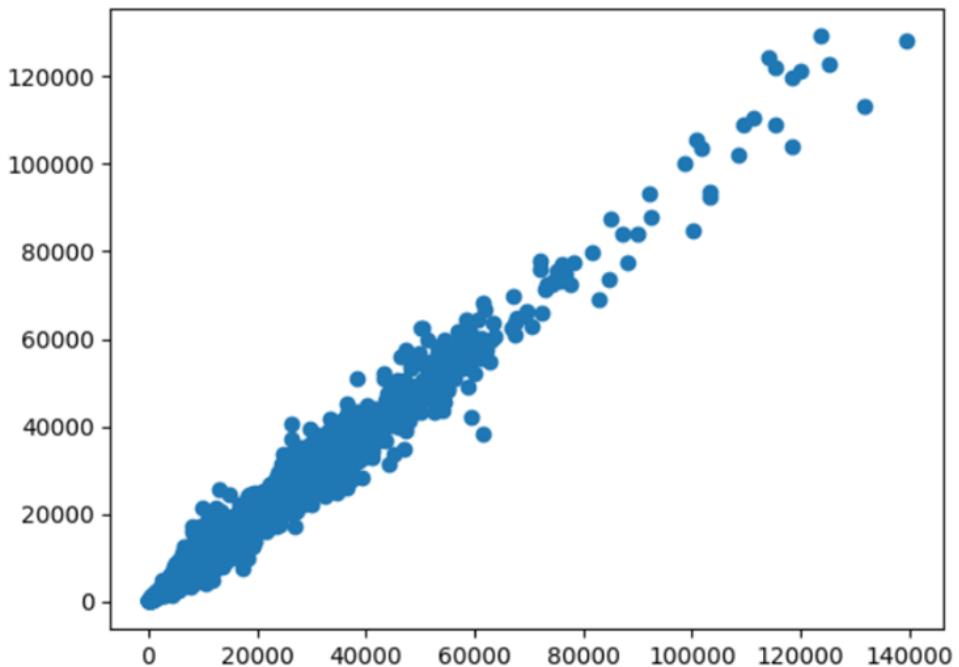


Figura 6.1: Risultati modello api

## 6.3 Funzionamento engine

Per assicurare un'integrazione fluida e un riutilizzo efficiente senza la necessità di riaddestrare il modello ad ogni richiesta, il modello stesso, insieme allo scaler e all'encoder (componenti essenziali per le trasformazioni dei dati), sono stati serializzati e salvati in formato .pkl tramite la libreria **joblib** [9]. Questo consente all'engine di ricaricare rapidamente questi componenti in memoria al momento della richiesta di una previsione, ottimizzando le prestazioni e la disponibilità del servizio.

Il funzionamento pratico dell'engine è intuitivo ed efficiente, concepito per una facile interazione tramite richieste HTTP. Quando l'API riceve una richiesta di previsione, attende due input fondamentali nel corpo della richiesta JSON: una **data** specifica (nel formato "YYYY-MM-DD") e il **layerid** di un'Area Censuaria Elementare (ACE).

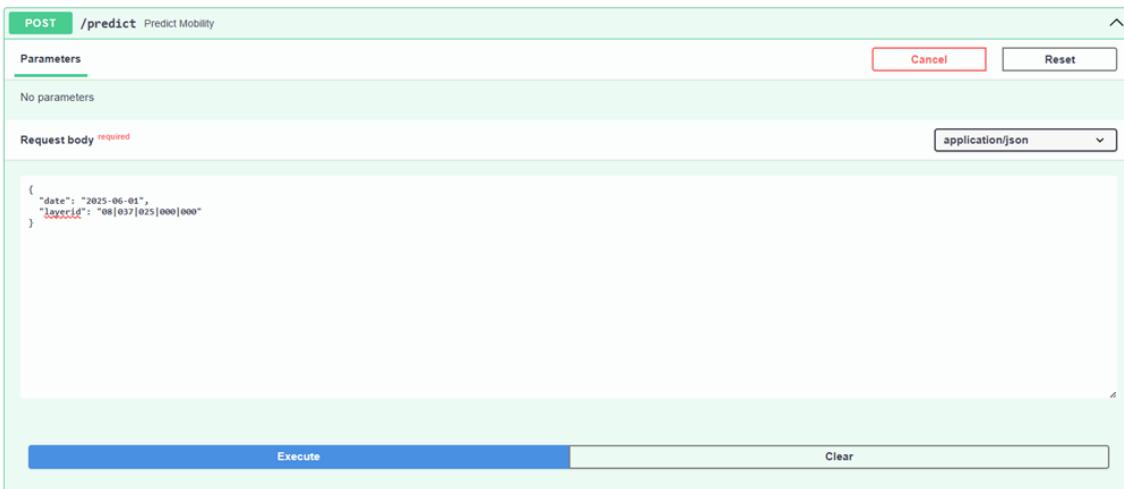


Figura 6.2: Esempio di richiesta

Sulla base di questi input, il modello predittivo viene interrogato. Il processo prevede che i dati in ingresso vengano pre-processati utilizzando lo scaler e l'encoder precedentemente salvati. Il modello serializzato (.pkl) viene poi utilizzato per generare la previsione del numero di persone in ingresso nell'area specificata per il giorno richiesto. L'API restituisce quindi questa stima come risposta.



Figura 6.3: Esempio di risposta

Questa visualizzazione mostra chiaramente come i dati di input vengano inviati all'endpoint /predict e come l'API risponda con il valore di previsione calcolato.

In conclusione, la Mobility API rappresenta un'implementazione chiave del progetto, trasformando un modello predittivo complesso in un servizio facilmente accessibile e integrabile. Questa soluzione non solo facilita l'aggiornamento dei dati e la scalabilità del servizio, ma apre anche nuove opportunità per l'integrazione in strumenti di supporto decisionale.

# Conclusioni

L'elaborato ha affrontato il problema dell'analisi e della previsione dei flussi di mobilità urbana nella regione Emilia-Romagna, dimostrando come l'unione di tecniche statistiche, strumenti geografici e modelli predittivi di machine learning possa fornire un valido supporto alla comprensione e alla gestione dei fenomeni di spostamento su scala regionale.

A partire da un dataset complesso e dettagliato, ottenuto dai segnali della rete mobile e arricchito da dimensioni spaziali, temporali e demografiche, sono state condotte analisi esplorative mirate a identificare pattern ricorrenti, anomalie legate ad eventi straordinari e differenze tra categorie socio-demografiche. Le rappresentazioni geografiche hanno inoltre permesso di evidenziare le disomogeneità territoriali e il ruolo dei centri attrattori nei flussi.

L'implementazione di modelli di regressione basati su Gradient Boosting ha dimostrato la possibilità di stimare con buona accuratezza i flussi di ingresso in ciascuna ACE, con performance particolarmente positive in aree urbane e industriali. In particolare, l'uso di feature temporali storiche (lag) ha mostrato la migliore capacità di generalizzazione su scala regionale, consentendo al modello di adattarsi a territori con caratteristiche diverse.

L'integrazione del modello predittivo all'interno di un'API realizzata con FastAPI ha infine completato l'obiettivo applicativo del progetto, rendendo disponibile uno strumento scalabile e accessibile per la previsione in tempo reale dei flussi di mobilità.

## Sviluppi futuri

Numerose sono le possibili direzioni di approfondimento e miglioramento:

- **Estensione temporale del dataset:** l'inclusione di dati relativi ad altri mesi o anni permetterebbe una maggiore robustezza nella modellazione, evidenziando fenomeni stagionali e cicli annuali.
- **Arricchimento delle feature:** l'integrazione di informazioni esterne come eventi programmati, condizioni meteo o orari del trasporto pubblico potrebbe migliorare la predittività in contesti complessi o turistici.
- **Modelli alternativi:** l'utilizzo di approcci alternativi potrebbe catturare meglio le dipendenze spazio-temporali nei flussi di mobilità.

- **Analisi in tempo reale:** un'evoluzione dell'API con supporto a dati in *streaming* consentirebbe applicazioni in scenari di *smart city* o gestione dinamica del traffico.

In conclusione, questo progetto rappresenta un primo passo verso una mobilità urbana più consapevole, adattiva e *data-driven*, in cui l'analisi dei dati diventa uno strumento centrale per il supporto alle decisioni e la pianificazione sostenibile del territorio.

# Bibliografia

- [1] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, 2016.
- [2] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2 edition, 2019.
- [3] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2 edition, 2017.
- [4] Kelsey Jordahl et al. Geopandas: Python tools for geographic data, 2023.
- [5] Folium Developers. Folium documentation, 2023.
- [6] PostgreSQL Global Development Group. Postgresql documentation, 2023.
- [7] Sebastián Ramírez. Fastapi documentation, 2023.
- [8] Uvicorn Developers. Uvicorn: Asgi web server implementation for python, 2023.
- [9] Joblib Developers. Joblib: running python functions as pipeline jobs, 2023.