UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

TESI DI LAUREA

Implementazione di Modelli di Machine Learning in MLOps per la Pianificazione urbana delle Smart City: Un Caso di Studio su Breda e ’s-Hertogenbosch, Paesi Bassi

RELATORE

Prof. Fabio Palomba

Università degli Studi di Salerno

Dott. Fabiano Pecorelli

Jheronimus Academy Of Data Science

CANDIDATO

Marco Costante

Matricola: 0522501330

Anno Accademico 2022-2023

Questa tesi è stata realizzata durante un traineeship presso il

in collaborazione con il

"In God we trust, all others must bring data."

W. E. Deming

Abstract

La crescita continua del tasso di urbanizzazione nei Paesi Bassi rappresenta una sfida significativa per le città della nazione. Esse infatti sono costantemente tenute ad affrontare problematiche legate alla pianificazione urbana, la congestione del traffico, la sicurezza pubblica e le esigenze energetiche.

Per far fronte a questa crescente tendenza, le amministrazioni comunali di Breda e 's-Hertogenbosch hanno deciso di aderire al progetto **Smart City Monitor**, un'iniziativa data-driven, che coinvolge partner provenienti dall'industria e dal governo volta a migliorare la qualità della vita dei cittadini.

Il seguente lavoro di tesi è stato svolto nel contesto del suddetto progetto, con l'obiettivo principale di analizzare dati real-time, provenienti dalle amministrazioni comunali in esame, e implementare pipeline di addestramento automatico, scalabili e generalizzabili, per modelli di Machine Learning applicati a serie temporali.

Questo è risultato nello sviluppo di modelli predittivi in grado di generare predizioni in tempo reale di vari fenomeni critici in ambito Smart Cities.

I risultati ottenuti hanno mostrato performance generalmente accurate, indicando l'efficacia di tale approccio per pianificare e ottimizzare la gestione delle città, in particolare per affrontare problemi come la congestione del traffico pedonale e la gestione della disponibilità di parcheggi.

Indice

Elenco delle figure

Elenco delle tabelle

CAPITOLO 1

Introduzione

1.1 Contesto applicativo

La densità di popolazione dei Paesi Bassi è una tra le più alte dell'Unione Europea, il che ha portato negli ultimi anni, ad una notevole crescita del suo tasso di urbanizzazione [1]. Come si nota in Figura 1.1, a partire da primi anni '60 c'è stato un notevole aumento della popolazione urbana, fino al 2021 anno in cui circa il 92,57% della popolazione nazionale risiedeva in città.

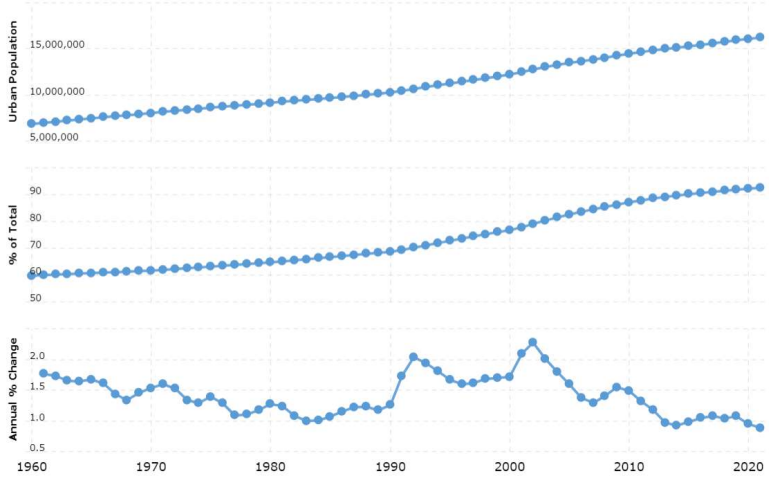

Figura 1.1: Popolazione urbana dei Paesi Bassi nel periodo 1960-2023.

Difronte a questa crescente tendenza, alcune amministrazioni comunali dei Paesi Bassi hanno deciso di affrontare il fenomeno attraverso l'uso di approcci data-driven, al fine di offrire una gestione efficiente dei centri urbani garantendo sicurezza pubblica, gestione della mobilità e partecipazione attiva dei cittadini.

Entra in gioco il concetto di **Smart City**: *"una visione di sviluppo urbano che integra in modo sicuro la tecnologia dell'informazione e della comunicazione (ICT) e la tecnologia dell'Internet delle cose (IoT) al fine di gestire gli asset di una città"* [2].

Il governo dei Paesi Bassi sta esplorando soluzioni basate su nuove tecnologie per affrontare le suddette sfide. Il Ministero dell'Infrastruttura e della Gestione delle Acque si sta impegnando a creare un sistema di mobilità che sia facile da usare, offrendo opzioni di trasporto ottimali. Per raggiungere questo obiettivo, è fondamentale coinvolgere attivamente i cittadini stessi, spronandoli a prendere decisioni consapevoli come evitare sia gli orari di punta che le strade con elevato traffico pedonale e considerare alternative come il trasporto pubblico o il bike renting. Inoltre, si cerca di migliorare la qualità del trasporto e ridurre gli ingorghi stradali, contribuendo anche alla protezione dell'ambiente [3].

1.2 Motivazioni e obiettivi

L'urbanizzazione e la densità di popolazione significativa nei Paesi Bassi presentano una serie di sfide uniche, tra cui la congestione del traffico dovuta a strade affollate e una serie di problemi legati alla sicurezza urbana. In particolare, la diffusa pratica dell'utilizzo delle biciclette, con una media di circa 1,3 biciclette per cittadino in una popolazione di circa 17 milioni, ha portato ad una saturazione dei parcheggi delle biciclette e ha suscitato preoccupazioni riguardo ai furti e ai danneggiamenti. La popolarità delle biciclette come mezzo di trasporto quotidiano è un riflesso della cultura dei Paesi Bassi, promuovendo uno stile di vita sano e sostenibile. Tuttavia, questa pratica ha anche generato una sfida amministrativa nel fornire adeguati parcheggi sicuri, poiché spesso le biciclette vengono abbandonate o esposte agli agenti atmosferici.

In aggiunta, la densità di popolazione e il turismo portano spesso a strade molto affollate e, in alcuni casi, a situazioni di borseggio nelle aree più congestionate durante eventi pubblici frequenti, i quali rappresentano un aspetto intrinseco della vivace vita sociale della nazione. La gestione di tali problematiche diventa cruciale per garantire un ambiente urbano sicuro e sostenibile nei Paesi Bassi.

Per far fronte ai problemi precedentemente introdotti nasce il progetto Smart City Monitor [4] [5], un'iniziativa che coinvolge partner provenienti dall'industria, dal governo e dal settore dell'istruzione, volta a migliorare la qualità della vita nelle città, della regione del Brabante, di Breda e 's-Hertogenbosch.
Questa tesi è il risultato di 4 mesi di lavoro e partecipazione attiva al progetto, presso la Jheronimus Academy of Data Science, uno dei principali partner coinvolti.

La fase preliminare del lavoro di tesi è stata caratterizzata da un'analisi approfondita dello stato dell'arte nel campo di studio.
Questo approfondimento ha rivelato una serie di limitazioni sostanziali nel contesto delle attuali ricerche, poiché gran parte degli studi si sono concentrati principalmente sulla prevenzione degli incidenti stradali, sull'analisi in ambienti chiusi e su categorie protette. L'assenza di un approccio mirato alla progettazione e alla gestione degli spazi urbani rappresenta una lacuna importante, considerando l'importanza di sviluppare strategie preventive e soluzioni ottimali per una vasta gamma di questioni legate alla pianificazione e ottimizzazione urbana.
Segue una fase di esplorazione approfondita dei fenomeni rilevanti nei due contesti urbani di interesse. Questa ha fornito una base per la successiva progettazione delle pipeline di addestramento per modelli di Machine Learning applicati alle serie temporali. Nel corso di questa analisi, è stata dedicata particolare attenzione all'identificazione di features esogene, statistiche e temporali, mirando a selezionare quelle più rilevanti per la generazione di previsioni attendibili in tempo reale.
La sperimentazione è proseguita con la valutazione di diversi modelli, compresi modelli statistici e di ensemble, al fine di identificare quelli più accurati attraverso un confronto dei risultati su diversi esperimenti.

La fase finale ha visto l'integrazione di queste conoscenze nell'implementazione di pipeline MLOps sottoforma di APIs, garantendo che i modelli addestrati potessero essere incorporati in modo fluido nella piattaforma Smart City Monitor. La frequenza delle predizioni in tempo reale è stata definite per avvenire ogni 2 ore, consentendo un monitoraggio costante e tempestivo del traffico pedonale e della disponibilità di parcheggi.

1.3 Risultati ottenuti

Il risultato di questo lavoro di tesi è stato quello di esplorare, sperimentare e valutare diverse strategie di Machine Learning al fine di identificare le più idonee e accurate per modellare fenomeni temporali in una Smart City.
Attraverso il processo di ricerca, sono state sviluppate e testate diverse pipeline di addestramento, integrate con un approccio operativo, al fine di generare deliverables di Machine Learning sottoforma di APIs, pronti per la creazione di predizioni integrabili in vari contesti, come ad esempio la creazione di una dashboard.
Sebbene il modello ARIMA abbia dimostrato generalmente buone performance in vari contesti, è emerso che l'utilizzo di XGBoost addestrato su un dataset composto sia da variabili temporali autoregressive che da variabili esogene, come i dati meteorologici, permette di creare modelli più avanzati sia in termini di accuratezza che di ottimizzazione delle risorse temporali e di memoria.
Particolarmente sorprendenti sono stati i risultati ottenuti nei casi specifici del traffico pedonale a 's-Hertogenbosch e della disponibilità dei parcheggi per biciclette a Breda. Per la previsione del traffico pedonale a 's-Hertogenbosch, si è registrato un errore medio di circa 70 pedoni, considerando che la media per connessione stradale è di circa 270 pedoni passanti per ora. Analogamente, per la previsione di disponibilità di parcheggi per biciclette a Breda, si è ottenuto un errore medio di 39 unità, su parcheggi che possono arrivare fino a 1384 posti di disponibilità. Il root mean squared error per i parcheggi auto a Breda e 's-Hertogenbosh è stato rispettivamente di 1200 e 1446 (su parcheggi auto che arrivano ad avere circa 15mila slot disponibili), evidenziando ulteriormente la precisione e l'efficacia dei modelli implementati.

I risultati ottenuti sono stati molto incoraggianti, mostrando performance generalmente accurate, suggerendo l'efficacia di un simile approccio per migliorare e gestire in maniera ottimale la vita cittadina

1.4 Struttura della tesi

Gli argomenti introdotti sono trattati in dettaglio nei seguenti capitoli che compongono la tesi:

- **Capitolo 2:** offre una panoramica del background sul quale questo lavoro è basato, definendo il dominio del problema e i concetti fondamentali da comprendere;
- **Capitolo 3:** presenta un'analisi dello stato dell'arte sul tema delle Smart City, analizzando lavori correlati di natura nazionale e non;
- **Capitolo 4:** introduce la metodologia adottata per il raggiungimento degli obiettivi preposti, sintetizzati attraverso la formulazione di Research Questions;
- **Capitolo 5:** descrive gli approcci utilizzati e analizza i risultati ottenuti dalle sessioni sperimentali eseguite sul caso di studio riguardante la città di 's-Hertogenbosch.
- **Capitolo 6:** descrive gli approcci utilizzati e analizza i risultati ottenuti dalle sessioni sperimentali eseguite sul caso di studio riguardante la città di Breda;
- **Capitolo 7:** mette in evidenza i risultati più significativi rispondendo alle Research Questions definite nel Capitolo 4 e riporta alcune limitazioni che possono influenzare la validità futuro di questo lavoro.
- **Capitolo 8:** fornisce le conclusioni del lavoro e definisce alcuni sviluppi futuri;

CAPITOLO 2

Background

L'aumento dell'urbanizzazione rappresenta una sfida sempre più grande per le amministrazioni comunali, che devono affrontare problemi come la congestione del traffico, la sicurezza pubblica, la gestione dei rifiuti e le esigenze energetiche. L'evoluzione digitale sta fornendo nuove soluzioni per affrontare questi problemi e gli enti locali stanno rispondendo a queste sfide adottando il concetto di Smart City, utilizzando tecnologie avanzate per migliorare la qualità della vita dei cittadini e la gestione delle risorse urbane. In questa sezione, si fornirà una panoramica aggiornata del background sul quale questo lavoro di tesi è basato, mettendo in luce i principali aspetti tecnologici e sociali che caratterizzano questa innovativa concezione urbana.

2.1 Data Science e Machine Learning

La Data Science [6] è un campo in continua evoluzione e sempre più importante per l'analisi e l'utilizzo dei dati in molti settori, non solo legati all'informatica. Il suo obiettivo principale è quello di estrarre informazioni significative dai dati al fine di sviluppare soluzioni innovative. Per raggiungere il suddetto obiettivo, la Data Science utilizza un insieme di tecniche e teorie multidisciplinari, come la matematica, la statistica, l'informatica e l'ingegneria informatica.

Attualmente ci troviamo nella cosiddetta "Rivoluzione Industriale dei Dati" [7], in cui la maggior parte dei dati è generata da macchine come, ad esempio, registrazioni software, telecamere, smartphone o reti di sensori wireless. Questi dispositivi producono dati che possono essere memorizzati a basso costo e sfruttati per ottenere utili informazioni e per effetto della legge di Moore, il tasso di produzione di questi dati aumenta e aumenterà in modo esponenziale.

Affianco al concetto di Data Science spesso entra in gioco il concetto di "Big Data". I Big Data sono un insieme di dati di grandi dimensioni e complessità che rappresentano una sfida per le tradizionali applicazioni di elaborazione dati. La loro dimensione è tale da richiedere l'utilizzo di tecniche e strumenti specifici per la loro analisi e gestione.
La crescente frequenza con cui vengono prodotti i dati, ad esempio con l'avvento dell'IoT, fa si che i dati diventino molto velocemente obsoleti, il che richiede l'aggiornamento costante delle informazioni per garantire decisioni sempre più accurate.

La piramide DIKW (Data-Information-Knowledge-Wisdom) di Ackoff (1989) [8] è un modello concettuale nato per aiutare a comprendere come i dati possono essere utilizzati per creare valore attraverso l'elaborazione e l'interpretazione. La piramide, osservabile in Figura 2.1, è composta da quattro livelli:

- **Data:** il livello più basso della piramide, rappresenta i fatti grezzi e non elaborati;
- **Information:** i dati vengono organizzati e interpretati per creare informazioni significative;
- **Knowledge:** le informazioni vengono utilizzate per costruire la conoscenza, ovvero la comprensione dei fatti e dei concetti;
- **Wisdom:** il livello più alto della piramide rappresenta l'abilità di utilizzare la conoscenza per prendere decisioni sagge e basate su una visione globale del problema;

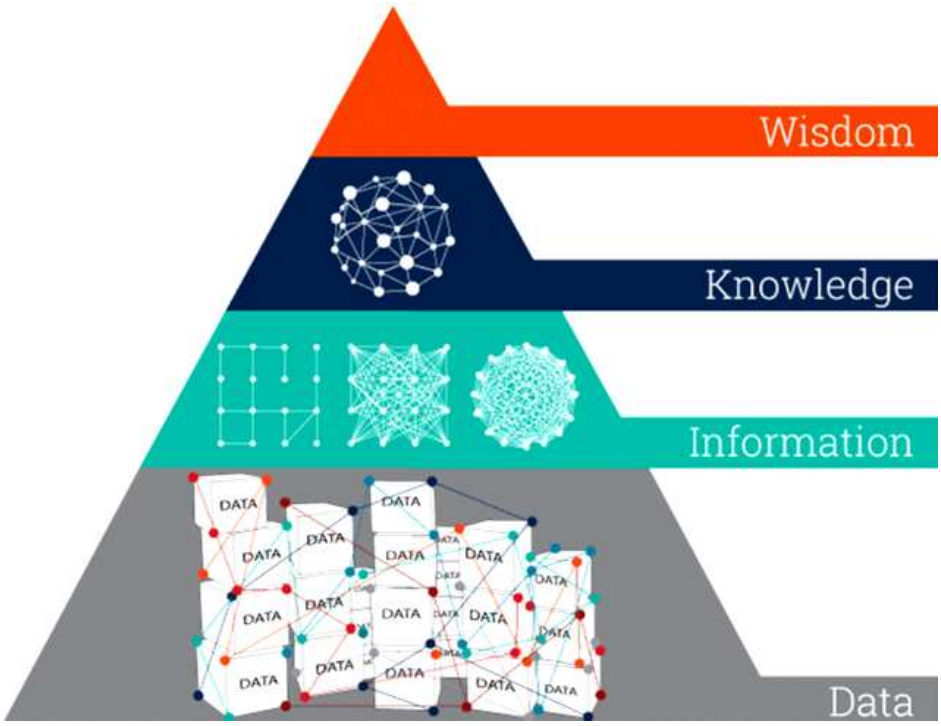

Figura 2.1: Piramide dikw

Uno dei modelli più diffusi, invece, per approcciarsi a problemi di Data Science è il CRoss Industry Standard Process for Data Mining [9], che nonostante sia stato concepito nel 1996, risulta essere ancora oggi largamente accettato.
Il CRISP-DM, come si vede in Figura 2.2, si articola in sei fasi interconnesse:

- **Business Understanding:** identificare il problema e definire gli obiettivi del progetto;
- **Data Understanding:** acquisire i dati necessari per il progetto;
- **Data Preparation:** eseguire la pulizia, la trasformazione e l'integrazione dei dati raccolti;
- **Modeling:** selezionare le tecniche di modellizzazione appropriate e creare il modello di analisi dei dati;
- **Evaluation:** valutare la qualità del modello creato;
- **Deployement:** implementare il modello creato e fornire risultati all'utente finale;

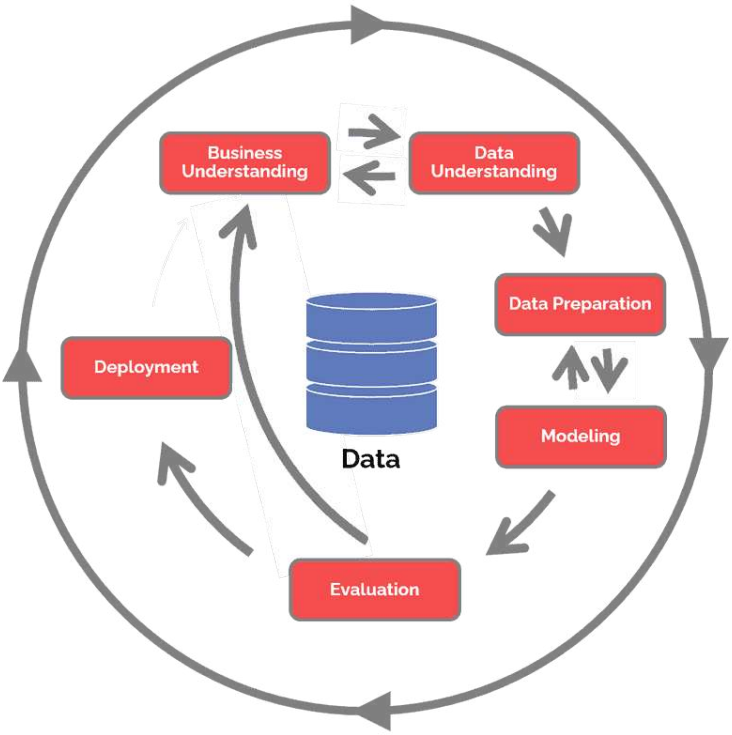

Figura 2.2: Modello CRISP-DM

Il Machine Learning (ML) [10] rappresenta la pratica di far agire i computer in modo autonomo senza bisogno di una programmazione esplicita.
Esso si suddivide in tre tipologie principali:

- **Apprendimento supervisionato:** si basa sull'individuazione di una corrispondenza tra variabili dipendenti e fattori indipendenti, utilizzando esempi etichettati con la risposta corretta. Un tipico esempio è la classificazione;
- **Apprendimento non supervisionato:** mira a individuare i modelli e pattern presenti nei dati senza l'utilizzo di etichette. Un'applicazione tipica di questa tecnica è il clustering;
- **Apprendimento per rinforzo:** si basa sull'utilizzo di un concetto di ricompensa e penalità per quantificare le performance del sistema di ML, come per l'addestramento di un agente in un videogioco;

Sono molteplici le sfide da affrontare quando ci si approccia ad un problema di Machine Learning a partire dall'analisi, la documentazione e i processi per identificare i requisiti del problema.

Lo sviluppo e l'implementazione dei modelli di Machine Learning sono affetti da molte sfide critiche, che devono essere affrontate per garantire il successo dei progetti e l'efficacia dei modelli implementati [11]. Ad esempio, è importante considerare l'impatto dei modelli sui destinatari in modo da evitare discriminazioni o altri effetti negativi. Il costo e la complessità del processo di sviluppo possono anche rappresentare un aspetto cruciale, specialmente per le aziende che devono bilanciare le proprie esigenze con la qualità dei modelli. Inoltre, è importante selezionare obiettivi appropriati e formare team di lavoro competenti e collaborativi.
Spesso i progetti di Machine Learning falliscono a causa di una mancanza di comprensione del dominio del problema, dati di bassa qualità, scelta di modelli inadeguati e mancanza di un'adeguata gestione del ciclo di vita del modello. Affrontare queste sfide richiede una combinazione di competenze tecniche e conoscenza del dominio del problema, oltre ad una gestione del processo di sviluppo e un monitoraggio del modello dopo l'implementazione. In definitiva, è evidente come sia necessario un approccio ingegneristico al problema.

2.2 Software engineering for AI: MLOps

Costruire un prodotto di Machine Learning è una sfida che richiede la considerazione di una molteplicità di aspetti, non solo legati strettamente al mondo dei dati. Non è infatti sufficiente costruire un modello statico per affrontare e risolvere questioni in un'ottica aziendale.
I modelli di Machine Learning hanno bisogno di costante "manutenzione", al fine di aggiornarne i parametri in base alle continue evoluzioni delle informazioni e valutarne l'effettiva efficacia ed efficienza. Dal punto di vista pratico, poi, è necessario sviluppare un prodotto finito e scalabile che possa essere utilizzato per fornire predizioni e risultati validi ai suoi fini.

DevOps (Development+Operations) include un insieme di protocolli al fine di offrire un passaggio fluido dallo sviluppo di un software al suo deployment in un ambiente di produzione, che possa essere offerto agli utenti finali, proponendo quindi un metodo per connettere figure responsabili di diversi aspetti.

La gestione di un progetto software che includa una componente ML richiede particolare attenzione dato che è necessario integrare codici e dipendenze in una maniera scalabile e portabile. Risulta quindi fondamentale una collaborazione tra la figura del Data Scientist e quella del Software Engineer.

MLOps è una metodologia per il Machine Learning engineering che unifica lo sviluppo della componente ML con la componente operazionale. Supporta l'automazione dei passaggi critici della costruzione di un sistema di Machine Learning e fornisce una serie di processi standardizzati e funzionalità tecnologiche per la creazione, l'implementazione e messa in funzione di sistemi ML, in modo rapido e affidabile. [12]

Figura 2.3: Ciclo di vita di MLOps

Il ciclo di vita di MLOps può essere riassunto in 7 fasi, mostrate in Figura 2.3, interconnesse tra di loro:

- **Ml development:** riguarda la sperimentazione e lo sviluppo di una procedura di addestramento dei modelli che sia robusta e riproducibile. Essa consiste in molteplici attività, dalla definizione dei task alla preparazione e trasformazione dei dati, all'addestramento, fino ad arrivare alla valutazione dei modelli;
- **Training operationalization:** riguarda l'automazione del processo di building, testing e deployment di pipeline di addestramento ripetibili e affidabili;
- **Continuous training:** riguarda l'esecuzione ripetuta della pipeline di addestramento in risposta a nuovi dati o a modifiche del codice, che possono avvenire ad intervalli regolari attraverso invocazioni manuali oppure in seguito al verificarsi di determinati eventi scatenanti;
- **Model deployment:** riguarda il "packaging", la verifica e la distribuzione di un modello in un ambiente di servizio per la sperimentazione online e la produzione;
- **Prediction serving:** riguarda la fase durante la quale il modello è pronto per accettare richieste e produrre predizioni;
- **Continous monitoring:** riguarda il controllo dell'efficacia e dell'efficienza di un modello in produzione;
- **Data and model managment:** è una funzione centrale per la gestione degli artefatti di ML per supportare la capacità di revisione, la tracciabilità e la conformità. La gestione dei dati e dei modelli può anche promuovere la condivisibilità, la riusabilità e la scopribilità degli asset di ML;

2.3 Serie temporali

Una serie temporale non è nient'altro che una raccolta di informazioni ottenute in un dato periodo di tempo a intervalli più o meno regolari.

Esse vengono tipicamente rappresentate attraverso l'uso di grafici caratterizzati da un asse temporale sul quale vengono mostrati i dati e la loro evoluzione, attraverso l'uso di segmenti che uniscono le varie osservazioni al fine di visualizzare in maniera più chiara l'andamento del fenomeno in esame.

Il motivo per cui è stato necessario approfondire il tema delle serie temporali in questo lavoro di tesi è che tutti i fenomeni affrontati ed analizzati, come la predizione del traffico pedonale, sono basati su rilevazioni di dati real-time, distribuite nel tempo. Si è reso dunque necessario approfondire il dominio al fine di identificare le tecniche e le pratiche più idonee a modellare il problema.

In linea generale si possono avere diverse tipologie e distinzioni di serie temporali, non necessariamente mutualmente esclusive tra di loro:

- **Metriche:** misurazioni e osservazioni raccolte a intervalli regolari;
- **Eventi:** misurazioni e osservazioni raccolte in modo irregolare, che rendono la predizione di eventi futuri molto difficoltosa;

Un'altra distinzione comune è quella tra serie temporali lineari e non lineari:

- **Lineari:** ogni punto può essere visto come combinazione lineare dei valori passati o futuri;
- **Non Lineari:** costituite da variabili che non possono essere caratterizzate da processi lineari;

Le classi comuni di serie temporali sono invece:

- **Serie temporali classiche:** un insieme di osservazioni, di una singola variabile, lungo il tempo;
- **Cross-sectional:** un insieme di osservazioni, per un gruppo di variabili, lungo il tempo;
- **Panel:** un insieme di osservazioni, per un gruppo di variabili, in molteplici intervalli temporali;

Una serie temporale è ovviamente definita da un ordine, che rappresenta la chiave per l'interpretazione e l'analisi del fenomeno che mostra. Ciò significa che una serie temporale deve essere immutabile e le nuove osservazioni devono essere successive a quelle pregresse.

L'analisi di una serie temporale può essere utile per valutare come una o più variabili cambiano nel tempo e permette di estrarre statistiche e caratteristiche dei dati in esame. Uno degli aspetti più importanti legati a questo tema è quello della **stazionarietà**. Esso entra in gioco molto spesso quando ci si approccia a modelli statistici e di Machine Learning per l'analisi e la predizione.
In sintesi una serie temporale si dice stazionaria se è caratterizzata da una media, varianza e auto-correlazione costante, ovvero se la distribuzione congiunta di $(Y_{s+1}, Y_{s+2}, \ldots, Ys + T)$ non dipende da s.
In particolare la media si definisce come la somma di tutte le osservazioni diviso il numero totale di osservazioni:

$$E(X) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{2.3.1}$$

La varianza rappresenta quanto le osservazioni si discostano dalla media:

$$VAR(X) = \sigma_X^2 = \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)^2\Big]. \tag{2.3.2}$$

Infine per auto-correlazione si intende la correlazione di una serie con i propri valori ritardati:

$$\operatorname{cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \tag{2.3.3}$$

$$autocorrelazione = corr(Y_t, Y_{t-j}) = \frac{cov(Y_t, Y_{t-j})}{\sqrt{var(Y_t) * var(Y_{t-j})}} \tag{2.3.4}$$

Le serie temporali possono inoltre essere caratterizzate da una serie di fattori:

- **Trend:** qualsiasi cambiamento sistematico in un livello di una serie. Vi sono due tipi di trend:
  - **Deterministico:** che è una funzione non aleatoria del tempo;
  - **Stocastico:** che è aleatorio e varia nel tempo;
- **Stagionalità:** uno schema ripetuto di aumento e diminuzione nei valori della serie, che si verifica costantemente per tutta la sua durata. La stagionalità è comunemente considerata come un modello ciclico o ripetitivo all'interno di un periodo stagionale di un anno con stagioni mensili o stagionali;
- **Residui:** ciò che resta dopo aver rimosso la stagionalità e il trend dai dati.

Per quanto riguarda, infine, i metodi di analisi delle serie temporali, esistono diversi criteri di classificazione:

- **Metodi frequency-domain:** incentrati sull'analisi della frequenza, invece che del tempo, di segnali o funzioni matematiche;
- **Metodi time-domain:** focalizzati sull'analisi di segnali o funzioni matematiche in relazione al tempo;
- **Metodi parametrici:** presuppongono che il processo stocastico stazionario sottostante abbia una certa struttura che può essere descritta utilizzando un limitato numero di parametri;
- **Metodi non parametrici:** richiedono di stimare esplicitamente la covarianza o lo spettro del processo, senza assumere che il processo abbia una struttura particolare.

2.4 Modelli per time-series forecasting

La previsione di eventi e fatti futuri sulla base di dati presenti e passati è un problema particolarmente complesso. Le serie temporali sono fenomeni estremamente variabili, che possono essere influenzati da numerosi fattori esterni. Tipicamente, inoltre, è difficile avere a che fare con serie caratterizzate sempre dallo stesso andamento nel tempo. Basti pensare alla pandemia di COVID-19, durante la quale tutte le attività globali si sono fermate, e che ha reso l'uso dei dati rilevati durante lo stesso periodo sostanzialmente inutili per l'analisi di fenomeni post-pandemia. Risulta pertanto fondamentale analizzare i numerosi modelli presenti in letteratura, al fine di utilizzare quelli che maggiormente possono catturare gli schemi presenti nei fenomeni in esame.
L'elemento che sicuramente caratterizza questa classe di modelli è il fatto che essi prendono in input dati caratterizzati da un ordine cronologico. A seconda dello specifico modello utilizzato è possibile poi includere variabili esogene, ovvero fattori indipendenti e trasversali, che facilitano la modellazione del problema.
In questa sezione verranno descritti i principali modelli per la previsione di serie temporali analizzati e sperimentati in questo lavoro di tesi.

Seppure l'obiettivo finale di questo lavoro non sia quello di offrire una panoramica degli algoritmi presenti in letteratura, suddetta descrizione è fondamentale al fine di comprendere e motivare le differenze prestazionali in fase di analisi dei risultati.

Tutti i modelli utilizzati durante la sperimentazione utilizzano un approccio gerarchico. Si tratta di un metodo per fare previsioni per un gruppo di elementi con una struttura gerarchica.
Per creare una previsione accurata, il metodo prevede la scomposizione della gerarchia in diversi livelli e la creazione di una previsione per ogni livello. In questo modo si possono cogliere le relazioni e le dipendenze tra i diversi livelli della gerarchia.
Questa tecnica verrà descritta in maniera più chiara nei capitoli che seguiranno.

2.4.1 Extreme Gradient Boosting Regressor

XGBoost [13] è un potente algoritmo di Machine Learning utilizzato per task di regressione e classificazione, che risulta particolarmente adatto anche per la modellazione di serie temporali. Esso è sostanzialmente un'implementazione efficiente dell'algoritmo di Gradient Boosting [14], una classe di algoritmi di Ensamble Learning basati sull'uso di alberi decisionali, dove ogni albero viene aggiunto all'ensamble al fine di correggere l'errore fatto dal predittore precedente, il tutto utilizzando una funzione di perdita che sia differenziabile e idonea all'applicazione della tecnica del Gradient Boosting come, ad esempio, l'errore medio quadratico. Al fine di utilizzare suddetto algoritmo per la modellazione di serie temporali è fondamentale trasformare il problema in un task di apprendimento supervisionato e utilizzare un approccio di validazione diverso dalla classica cross-validation.
Essendo una serie temporale caratterizzata da una sequenza temporale di osservazioni che ne rappresenta l'identità, non è pensabile dividere il set di addestramento e test in maniera casuale, ma bisogna conservarne l'ordine. A tal proposito è stato utilizzata una strategia di validazione incrociata chiamata "Expanding Window Splitter", che tiene conto dell'ordine temporale delle osservazioni. Nello specifico la dimensione dell'insieme di addestramento viene aumentata progressivamente simulandone l'andamento reale. I parametri utilizzati durante l'addestramento del suddetto modello sono riassunti in Tabella 2.1.

Tabella 2.1: Griglia di parametri per l'addestramento del modello XGB

Parametro	Valori
`nthread`	4
`objective`	`reg:squarederror`
`learning_rate`	0.06, 0.07, 0.08
`max_depth`	8, 9, 10
`min_child_weight`	4
`subsample`	0.7
`n_estimators`	300, 500

2.4.2 ARIMA

ARIMA (Autoregressive Integrated Moving Average) è un modello statistico utilizzato per l'analisi e la previsione delle serie temporali. Il modello prende in considerazione i valori passati di una serie temporale per prevedere i valori futuri, combinando l'autoregressione (AR), con la media mobile (MA). La parte integrata (I) di ARIMA fa riferimento ai termini di differenziazione, utilizzati per rimuovere le tendenze o le componenti stagionali dalla serie al fine di renderla costante nel tempo. Per autoregressione (AR) si intende il processo con il quale vengono utilizzati i valori passati di una serie temporale, denominati lag, per modellare quelli futuri:

$$Y = B_0 + B_1 * Y_{lag_1} + B_2 * Y_{lag_2} + ... + B_n * Y_{lag_n} \tag{2.4.1}$$

Quindi il valore attualmente osservato di Y è una funzione lineare dei suoi n valori precedenti (dove B_0, B_1 ecc. sono i coefficienti di regressione usati durante l'addestramento del modello). Per Integrazione si intende il processo di applicare un termine di differenziazione ai dati, nello specifico:

$$Y_{next} - Y = B_0 + B_1 * (Y - Y_{lag_1}) + B_2 * (Y_{lag_1} - Y_{lag_2}) + ... + B_n * (Y_{lag_{n-1}} - Y_{lag_n}) \tag{2.4.2}$$

Con il termine di media mobile (MA) vengono utilizzati gli errori di previsione passati per aggiustare le previsioni future:

$$Y = B_0 + B_1 * E_{lag_1} + B_2 * E_{lag_2} + ... + B_n * E_{lag_n} \tag{2.4.3}$$

dove E rappresenta le deviazioni residue casuali tra il modello e la variabile target.
I modelli di regressione con errori, come ARIMA, possono diventare piuttosto complessi a causa delle numerose operazioni matriciali che essi comportano. È, dunque, fondamentale cercare il modello e i parametri più parsimoniosi.
Per trovare il giusto ordine di integrazione, bisogna valutare se la serie è stazionaria attraverso l'uso di un test statistico.
In letteratura sono presenti diverse proposte, ma le metodologie utilizzate in questo lavoro sono state due:

- **Augmented Dickey Fuller (ADF) test:** l'ipotesi nulla del test ADF è che la serie temporale non sia stazionaria.

Quindi, se il $p - value$ del test è inferiore al livello di significatività (0,05), si rifiuta l'ipotesi nulla e si deduce che la serie temporale è effettivamente stazionaria, se il $p - value > 0.05$ si procede con la ricerca dell'ordine di differenziazione;

- **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:** l'ipotesi nulla del test KPSS è che la serie temporale sia stazionaria. Se il $p - value$ del test è inferiore al livello di significatività (0,05), allora si rifiuta l'ipotesi nulla e si deduce che la serie temporale non è stazionaria e si procede con la ricerca dell'ordine di differenziazione, se il $p - value$ è maggiore di 0,05 si deduce che la serie è stazionaria.

Per identificare il giusto numero di termini di autoregressione si può utilizzare il grafico dell'autocorrelazione parziale (PACF). Il grafico misura la correlazione tra una serie temporale e i suoi ritardi, eliminando gli effetti dei ritardi intermedi.
Se il grafico PACF mostra una correlazione significativa ad un particolare lag, indica che il lag corrispondente è necessario per il termine AR. Al contrario, se non c'è una correlazione, suggerisce che il ritardo non è necessario nel modello.

Per determinare il numero di termini di media mobile si può, invece, utilizzare il grafico dell'autocorrelazione. Il grafico dell'autocorrelazione è uno strumento utilizzato per determinare se le proprietà statistiche della serie rimangono costanti nel tempo. Il grafico mostra la correlazione della serie temporale con i suoi valori passati, o lag, in diversi punti nel tempo.
L'asse delle ascisse del grafico rappresenta il tempo di ritardo, mentre l'asse delle ordinate mostra il coefficiente di correlazione che varia da -1 a 1. Un coefficiente di correlazione pari a 0 indica l'assenza di correlazione, mentre i coefficienti di -1 o 1 indicano rispettivamente una perfetta correlazione negativa o positiva. Se il grafico mostra un trend o un andamento periodico, suggerisce la presenza di un trend o di una stagionalità nella serie temporale.
Infine, se nell'utilizzare un modello ARIMA vengono incluse anche delle variabili esogene il modello prende il nome di ARIMAX.

I parametri utilizzati durante l'addestramento del suddetto modello sono riassunti in Tabella 2.2.

Tabella 2.2: Griglia di parametri per l'addestramento del modello ARIMA

Parametro	Valori
`p (termine AR)`	1,2,3,4
`d (termine I))`	0
`q (termine MA)`	1,2,3

2.4.3 SARIMA

SARIMA, o ARIMA stagionale, è un'estensione di ARIMA progettata per gestire dati di serie temporali con una componente stagionale. SARIMA introduce tre nuovi iperparametri per modellare le componenti di autoregressione (AR), differenziazione (I) e media mobile (MA) stagionale delle serie, insieme a un parametro per determinare la lunghezza del ciclo stagionale.
Se combinato con caratteristiche esogene, SARIMA prende il nome di SARIMAX. Ci sono quattro elementi stagionali che non fanno parte di ARIMA e devono essere configurati, ovvero l'ordine autoregressivo stagionale, l'ordine di differenza stagionale, l'ordine di media mobile stagionale e il numero di passaggi temporali per un singolo periodo stagionale. Per rimuovere gli effetti stagionali additivi, un modello ARIMA stagionale utilizza la differenziazione con un ritardo pari al numero di stagioni (s). Questa procedura, simile alla differenziazione con un ritardo di 1 per eliminare una tendenza, introduce un termine di media mobile con un ritardo s. Pertanto, il modello ARIMA stagionale comprende termini autoregressivi e di media mobile con un ritardo s.

2.5 Smart cities

Sono molteplici in letteratura le definizioni di "Smart City".
A. M. Nagy et al. [15] con il termine "Smart City" definiscono l'uso delle tecnologie dell'informazione e della comunicazione per rilevare, analizzare e integrare informazioni chiave dai sistemi centrali delle città in funzione. Allo stesso tempo, i servizi delle città intelligenti possono fornire risposte intelligenti a diversi tipi di esigenze in termini di vita quotidiana, protezione dell'ambiente e sicurezza pubblica, nonché alle attività delle strutture e dell'industria commerciale della città.

Tra gli obiettivi principali delle smart cities vi è la creazione di sistemi di trasporto e di gestione urbana intelligenti, in quanto questi hanno un potenziale impatto significativo sulla vita dei cittadini. K. R. Kunzmann [16] sottolinea come il concetto delle smart cities miri a coinvolgere e motivare i cittadini a partecipare in modo attivo alla vita della comunità, fornendo loro la possibilità di dare feedback sulla qualità dei servizi, migliorare lo stato dell'ambiente urbano, adottare uno stile di vita più sostenibile e partecipare a iniziative sociali o di sostegno a gruppi minoritari.

L'accelerata urbanizzazione ha modernizzato la vita di molte persone, ma ha anche portato molti problemi, come l'eccessivo traffico stradale con un conseguente aumento del consumo di carburante e all'emissione di enormi quantità di inquinanti. Questi fenomeni hanno un grande impatto sulla salute e sulla qualità della vita degli abitanti delle città. Secondo alcuni studi di laboratorio [17], l'inquinamento dell'aria legato al trasporto può aumentare il rischio di sviluppare allergie e può aggravare i sintomi, in particolare in sottogruppi suscettibili. Difronte a queste realtà è diventato sempre più evidente che le soluzioni tradizionali non sono più sufficienti. In questo contesto, tecnologie come il Machine Learning possono svolgere un ruolo fondamentale nell'aiutare le città a gestire meglio le loro risorse e fornire servizi più efficienti ai loro cittadini. Così facendo si è in grado di raccogliere e analizzare grandi quantità di dati acquisiti in tempo reale dalle città, identificare modelli e tendenze e, quindi, utilizzare queste informazioni per prendere decisioni informate e migliorare la qualità della vita urbana.

I Cyber-Physical Systems (CPS) sono una tecnologia innovativa che sta guadagnando sempre più attenzione nel contesto delle smart cities [18].
Si tratta di sistemi integrati composti da componenti fisici e digitali che interagiscono in modo intelligente e autonomo, utilizzando tecnologie avanzate come il Machine Learning, l'Internet of Things (IoT) e l'Intelligenza Artificiale (AI).
Grazie a questa integrazione, i CPS possono monitorare e analizzare il comportamento dei sistemi fisici, generare dati in tempo reale e fornire informazioni preziose per la gestione e l'ottimizzazione delle risorse urbane. I CPS possono essere generalmente caratterizzati come *"sistemi fisici ed ingegnerizzati il cui funzionamento è monitorato, controllato, coordinato e integrato da un nucleo di calcolo e comunicazione"* [19] e per queste caratteristiche intensive di interoperabilità, soprattutto in connessione con i loro corrispettivi nel mondo fisico, pongono diverse sfide di sicurezza [20], ma svolgono anche un ruolo chiave nell'interazione tra i luoghi fisici e cyber, dove entrambi si influenzano reciprocamente.

L'impiego dei Cyber-Physical Systems rappresenta un aspetto chiave per garantire la sicurezza degli spazi pubblici e migliorare la qualità della vita nelle smart cities di nuova generazione. Questi sistemi informatici, integrati nell'ambiente fisico, sono in grado di monitorare e controllare in tempo reale il mondo circostante, permettendo alle autorità di prevenire e gestire attacchi terroristici, furti, molestie e altre forme di violenza urbana. Grazie alla loro efficacia nel controllo del territorio, i CPS rappresentano una risorsa preziosa per garantire la sicurezza e il benessere della comunità.

La caratteristica predominante dei sistemi cyber-fisici è la loro possibilità di interagire con altri CPS oltre i confini del proprio sistema [21]. Questi sistemi sono in grado di collegare a loro volta diversi sottosistemi. Senza uno standard di networking, le possibilità per i sistemi cyber-fisici sono limitate. I CPS decentralizzati raccolgono dati dalle applicazioni reali, li elaborano in algoritmi complessi, trasferiscono i risultati ad altri sistemi incorporati e a grandi strutture, con elevata flessibilità, di elaborazione centrale.

Allo stesso modo, i dati vengono ricevuti da reti di computer ad alte prestazioni, database o altri sistemi incorporati.
Tra le molte applicazioni dei CPS, una delle più comuni è quella che riguarda la sicurezza. Grazie alla capacità di monitorare e controllare il mondo fisico in tempo reale, questi sistemi possono essere utilizzati per prevenire e mitigare problemi come incidenti stradali, congestione del traffico, eccessiva affluenza a eventi pubblici e minacce alla sicurezza pubblica.

2.5.1 Smart city monitor

Il progetto Smart City Monitor [4] [5] è un'iniziativa innovativa volta a migliorare la qualità della vita in alcune città dei Paesi Bassi, come Breda e 's-Hertogenbosch, con un occhio attento alla privacy dei dati.
L'obiettivo del progetto è concentrato su tre obiettivi principali:

- la ripresa economica delle città durante e dopo la pandemia COVID-19;
- la creazione di centri urbani più accessibili e attrattivi;
- la promozione di città pulite, salutari e sostenibili.

La piattaforma si basa sull'utilizzo di dati real-time, ma tutti i dati raccolti sono anonimi e non possono essere associati a individui specifici. Non vengono raccolte informazioni personali sensibili, come informazioni sulla salute o sulle abitudini di consumo. Inoltre, ogni flusso di dati viene controllato e approvato sia dal proprietario del dato (il comune o una terza parte) che dal team di Smart City Monitor per garantire la sicurezza dei dati e la privacy dei cittadini. Ciò significa che alcuni dati potrebbero non essere accessibili se non soddisfano il requisito di non tracciabilità. Tuttavia, ciò non influisce sulla qualità dei servizi che Smart City Monitor intende fornire. Questo permette di garantire il rispetto delle tre proprietà della "CIA Triad":

- **Confidenzialità:** solo personale autorizzato può accedere a dati sensibili;
- **Integrità:** solo personale autorizzato può modificare dati sensibili;
- **Disponibilità:** i servizi sono accessibili e disponibili quando gli utenti ne fanno richiesta.

L'iniziativa coinvolge sei partner di progetto provenienti dall'industria, dal governo e dall'istruzione nella regione del Brabante. La piattaforma offre anche dati sulla qualità dell'aria e sulla percezione dei visitatori che vengono collegati in un cosiddetto Smart City Digital Twin. Smart City Monitor è in grado di fornire informazioni importanti per fare previsioni e trovare soluzioni adeguate alle esigenze della città. Ad esempio, può essere utilizzato per individuare le zone più frequentate della città e migliorare l'accessibilità a queste zone, la gestione intelligente dei parcheggi oppure per individuare le aree in cui la qualità dell'aria è più bassa e trovare modi per ridurne l'impatto. In questo modo, fornisce informazioni preziose per la pianificazione urbana e può aiutare le autorità locali a prendere decisioni più informate e mirate.

Smart City Monitor risulta particolarmente utile anche per gestire eventi sociali. In particolare, il sistema ha raccolto dati sulle festività dell'11 novembre 2022 a 's-Hertogenbosch [22], durante le quali si attendevano molti visitatori. Questi dati sono stati archiviati affinchè potessero essere utilizzati per creare simulazioni delle folle e dei flussi di visitatori che verranno utilizzati per eventi futuri come il carnevale del 2023. Durante grandi eventi pubblici come il Carnevale, possono verificarsi lunghe code e alcuni luoghi potrebbero diventare molto affollati e pericolosi. I dati provenienti dal sistema Smart City Monitor possono pertanto aiutare a prevedere e prevenire tali situazioni.

CAPITOLO 3

Stato dell'arte

La letteratura sulle smart cities è vasta e in continua evoluzione. Analizzando lo stato dell'arte è possibile trovare molti studi e progetti che riguardano le smart cities, tipicamente da un punto di vista di pubblica sicurezza. Le metodologie più diffuse, che prevedono l'uso combinato di sistemi IoT e tecniche di Intelligenza Artificiali, possono essere riassunte cosi:

- Algoritmi di motion tracking basati tipicamente su sistemi video-based;
- Algoritmi di object identification utilizzati per task di monitoraggio del traffico automobilistico;
- Analisi del comportamento delle folle per la gestione del traffico pedonale;
- Analisi della traiettoria per prevenire incidenti stradali e pedonali.

Molti studi si sono concentrati principalmente sulla previsione del movimento dei pedoni in aree chiuse o dei percorsi che questi seguono in strade pubbliche. È importante, però, sottolineare che monitorare il livello di traffico pedonale in strada è altrettanto cruciale per garantire la sicurezza nelle smart cities. Le applicazioni delle tecnologie CPS possono essere utilizzate per rilevare i flussi di pedoni e prevenire incidenti in zone ad alta densità di traffico.

L'analisi dei dati sul traffico pedonale può essere utile per ottimizzare le infrastrutture e i servizi pubblici, come la disposizione di attraversamenti pedonali, la segnalazione luminosa e l'accesso alle aree pedonali. Il conteggio dei pedoni in città è influenzato da diversi fattori urbani come il clima, gli eventi speciali e la affidabilità dei servizi di trasporto pubblico. C'è quindi bisogno di una strategia a lungo termine per la pianificazione e la gestione urbana e per migliorare il trasporto urbano.
Altro tipico problema delle smart cities moderne è la gestione ottimale dei parcheggi. Tipicamente i cittadini delle smart city possono sapere quanti posti auto liberi ci sono sia per strada che in parcheggi grazie all'uso di sensori.
L'integrazione della tecnologia IoT può migliorare la previsione della disponibilità di parcheggio raccogliendo e utilizzando i risultati dell'analisi dei dati sulle abitudini di parcheggio dei cittadini.
J. Mihelj et al. [23] sostengono che l'integrazione di sistemi di controllo del traffico e di rilevamento delle violazioni stradali con informazioni provenienti da diverse fonti, come i sensori nelle piste ciclabili, i semafori intelligenti e le telecamere, nonché le biciclette pubbliche basate su dispositivi IoT, potrebbe migliorare potenzialmente le condizioni del traffico in un ambiente di smart city.
Per garantire un'efficace organizzazione del traffico in linea con le esigenze dei cittadini, le smart cities devono seguire una sequenza di compiti ben definiti.
In questo contesto, la gestione urbana può garantire un alto livello di responsabilità e protezione dei dati dei cittadini durante la fase di formulazione delle politiche, considerando la molteplicità di flussi dati provenienti da varie fonti di raccolta. Tuttavia, per raggiungere l'obiettivo di una mobilità multifunzionale, è necessario tener conto anche degli aspetti etici e della privacy relativi all'IoT per la mobilità, con l'obiettivo di aiutare cittadini, automobilisti e pedoni a gestire il traffico cittadino [24].
Da D. Mavrokapnidis et al. [25] viene introdotta una tecnica di monitoraggio e previsione basato sulla visione artificiale per consentire l'integrazione del comportamento collettivo nelle valutazioni spazio-temporali dell'esposizione allo stress termico urbano nei Smart City Digital Twins (SCDT), rappresentazioni digitali delle città che vengono costantemente arricchite con dati spazio-temporali provenienti da sistemi umani e infrastrutturali, con l'obiettivo di trasformare le sfide dell'urbanizzazione in opportunità per migliorare la qualità della vita.

Essi descrivono sostanzialmente un modello digitale in cui vengono raccolti dati sulle attività umane e infrastrutturali presenti nella città, con l'obiettivo di migliorare la qualità della vita dei cittadini attraverso una gestione più efficiente e centrata sui loro bisogni. Il metodo proposto utilizza anche un modello di previsione in grado di prevedere l'esposizione della comunità allo stress termico in tutta la città e ha mostrato come gli SCDT possano rendere la dinamica della comunità più accessibile ai responsabili della città. Questo consente ai responsabili della città di identificare le fluttuazioni spazio-temporali del livello di esposizione della comunità a condizioni avverse di comfort termico e ad altri pericoli. In questo modo, i responsabili della città possono identificare eventuali anomalie nascoste o imminenti (ad esempio, un trend crescente di stress termico può essere più dannoso durante i periodi di maggior affluenza), e utilizzare gli SCDT come strumento per la gestione pro attiva della città.
Presso l'università di Delft è stato effettuato uno studio con l'obiettivo di sviluppare dei percorsi ottimizzati per pedoni non vedenti, al fine di creare percorsi che tengano conto delle loro specifiche esigenze, garantendo un percorso sicuro e agevole [26]. In particolare, la ricerca si concentra sull'utilizzo di tecnologie avanzate e di strumenti di analisi dei dati per creare percorsi intelligenti che tengano conto delle diverse condizioni ambientali e di traffico. In questo modo, si cerca di migliorare l'esperienza di mobilità dei non vedenti rendendo più efficiente e accessibile il loro spostamento all'interno della città.
Sono stati creati due differenti sistemi: la prima tecnica prevede l'analisi visiva del percorso pedonale in confronto agli altri mezzi di trasporto; la seconda tecnica, invece, si basa sull'analisi quantitativa del percorso pedonale.
La ricerca si focalizza su aree di particolare interesse come la stazione centrale di 's-Hertogenbosch. Le stazioni rappresentano, infatti, un ambiente particolarmente pericoloso per i non vedenti. Le barriere architettoniche, l'assenza di segnali acustici e tattili possono rendere il loro percorso difficoltoso, mettendoli a rischio di cadute o incidenti. Inoltre, le stazioni sono spesso affollate e caotiche, aumentando il livello di stress e di difficoltà nella navigazione per le persone non vedenti.
Tale tema è stato oggetto anche di uno studio condotto da A. Cohen et al. [27], che si concentra sulle sfide che i pedoni non vedenti affrontano durante la navigazione e l'orientamento nello spazio urbano.

La ricerca dimostra la fattibilità dell'utilizzo di algoritmi di pianificazione del percorso basati su dati di mappatura di OpenStreetMap per calcolare percorsi personalizzati per i pedoni non vedenti. Tuttavia, questi algoritmi non tengono conto dei flussi di traffico pedonale e veicolare che possono influenzare le scelte di percorso dei pedoni non vedenti. L'obiettivo della ricerca è stato quello di sviluppare un modello per la previsione dei flussi di traffico pedonale che possa integrare questi dati mancanti. I risultati hanno mostrato che gli algoritmi di Machine lLearning, come le Random Forest, possono generare dati temporali necessari per arricchire il database di OpenStreetMap.

J. Bakerman [28] si concentra sull'analisi delle aree urbane della città di Den Bosch ('s-Hertogenbosch), come la stazione centrale, la cattedrale e la Maarketstraat, attraverso l'uso di Geographic Information Systems. L'approccio presentato utilizza un ambiente dati ibrido, con dati vettoriali e raster, per calcolare un percorso ottimale come parte della pianificazione del percorso pedonale. L'obiettivo della ricerca è quindi quello di calcolare il percorso ottimale per i pedoni in un ambiente urbano, basandosi su un dataset ibrido che contiene dati di rete vettoriali per rappresentare i percorsi a forma di corridoio e dati raster per rappresentare lo spazio pedonabile libero. Il metodo proposto, tuttavia, non prevede un controllo di coerenza per i percorsi fissi e le aree pedonali, il che causa un'imperfezione nella rete. Inoltre, il calcolo del percorso raster utilizzato non riesce a produrre i percorsi desiderati.
In un articolo di H. O. Jacobs et al. [29] viene presentato un nuovo algoritmo di previsione probabilistica per i pedoni basato su equazioni differenziali ordinarie e informazioni storiche delle traiettorie.
X. Wang et al. [30] si concentrano, invece, sulla creazione di un modello e di un sistema per prevedere il flusso pedonale in una città. Vengono utilizzati dati storici di conteggio dei pedoni registrati da sensori termici e laser installati in vari punti della città. Viene proposto un sistema di previsione robusto in grado di gestire diversi pattern temporali di flusso pedonale. L'analisi sperimentale mostra che il modello ARIMA proposto è efficace nella modellizzazione dei pattern dei giorni feriali e festivi, superando altri modelli all'avanguardia per la previsione a breve termine del conteggio dei pedoni.

Il sistema, che è stato valutato utilizzando un dataset fornito dalla città di Melbourne, utilizza solo i dati relativi al numero di pedoni rilevati dai sensori, senza considerare altri fattori esterni, come le condizioni ambientali.

F. Van den Bossche et al.[31] hanno analizzato l'effetto di diversi fattori esplicativi sulla sicurezza del traffico, al fine di migliorare la comprensione degli sviluppi nella sicurezza stradale in Belgio. A tal fine, sono stati sviluppati dei modelli per spiegare e prevedere la frequenza e la gravità degli incidenti. Nella ricerca sono stati sviluppati modelli di regressione con errori, come ARIMA, per indagare l'impatto del tempo, delle leggi e dei regolamenti e delle condizioni economiche sulla frequenza e gravità degli incidenti in Belgio. Se tutti i presupposti statistici sono soddisfatti, la combinazione di regressione e analisi di serie temporali offre un potente strumento per indagare l'effetto di diversi fattori sulla sicurezza del traffico. I risultati dimostrano che le condizioni meteorologiche e alcune normative hanno un effetto significativo sulla sicurezza stradale, mentre l'impatto delle condizioni economiche non è significativo. Le previsioni sono state plausibili e abbastanza accurate, ma mostrano la volatilità intrinseca presente nei risultati della sicurezza stradale, sottolineando l'importanza di un approccio statistico all'analisi degli incidenti.

CAPITOLO 4

Metodologia di sviluppo

Lo scopo di questa sezione è quella di descrivere le metodologie adottate al fine di modellare i fenomeni previsti dal progetto Smart City Monitor.
Le sfide affrontate possono essere riassunte da due fondamentali domande di ricerca, esposte nel dettaglio in questo capitolo.

4.1 Research Questions

RQ1. Quali sono le features più adatte a descrivere un problema di predizione di serie temporale in un'ottica di smart city?

Per rispondere a questa domanda, è stato condotto un approfondito esame dei dati che caratterizzano i diversi fenomeni del caso di studio e delle diverse classi di features a disposizione.
Le sperimentazioni sono partite da un'analisi empirica, testando l'utilizzo esclusivo di variabili esogene e componenti autoregressive. Successivamente, sono state esplorate features più complesse ottenute combinando entrambe le categorie, fino ad includere variabili statistiche su diverse finestre temporali.

L'obiettivo è stato individuare le caratteristiche più rilevanti per modellare in maniera accurata i fenomeni temporali nelle smart city.

RQ2. Quali sono le strategie e i modelli di Machine Learning più adatti per gestire e modellare un problema di smart city forecasting?

La risposta a questa domanda è strettamente collegata alla RQ1.
Diverse configurazioni di features hanno richiesto l'impiego di vari modelli di Machine Learning. Attraverso un approccio sperimentale e confrontando i risultati ottenuti, sono stati testati modelli statistici e non.
Il processo di selezione del modello più idoneo è stato iterativo, adattandosi alle specifiche esigenze di ciascuna configurazione di features. Questo approccio ha permesso di identificare l'insieme ottimale di features e il modello di machine learning più efficace per la previsione dei fenomeni nelle smart city.

È emerso che la natura temporale dei dati nelle smart city richiede un'attenzione particolare alla modellazione come serie temporali. Questa scelta è stata giustificata dalla necessità di catturare le dinamiche temporali e le relazioni tra le variabili che caratterizzano gli ambienti urbani in evoluzione continua.

La sfida maggiore è stata quella di identificare tecniche e strategie sufficientemente generalizzabili ai vari casi d'uso. Infatti, la necessità di fronteggiare aspetti come la gestione ottimale dei parcheggi e la predizione di traffico pedonale, in due diversi comuni di una nazione cosi vasta, solleva diverse problematiche legate alla differenza intrinseca dei dati e dei fenomeni che si verificano nelle varie città.
Si è reso cruciale un approfondito studio preliminare dell'andamento e della caratterizzazione dei due fenomeni nei comuni di Breda e 's-Hertogenbosch, per trovare gli approcci migliori, in termini di qualità dei risultati, ma anche di risorse computazionali da istanziare.

4.2 Metriche di valutazione

Per validare l'accuratezza dei modelli utilizzati durante la sperimentazione sono state utilizzate tre metriche largamente diffuse in letteratura [32]:

- **Mean Absolute Error:** attribuisce lo stesso peso ad errori grandi ed errori bassi e si definisce come la media in valore assoluto degli errori di previsione:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - m_i| \tag{4.2.1}$$

- **Mean Squared Error:**; si definisce come la media degli errori di previsione al quadrato, poiché gli errori positivi e negativi tendono a cancellarsi reciprocamente:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_i)^2 \tag{4.2.2}$$

- **Root Mean Square Error:** definito come la radice quadrata dell'errore quadratico medio, viene utilizzato per convertire l'unità di misura dell'errore all'unità di misura dei dati:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - m_i)^2} \tag{4.2.3}$$

In tutte le definizioni x_i rappresenta l'osservazione i-esima della serie e m_i l'i-esima predizione del modello.

4.3 Panoramica dei dati

I dati fondamentali utilizzati durante tutto il lavoro sono stati acquisiti attraverso l'uso di numerosi sensori presenti nelle città in esame.
Tali sensori permettono di acquisire e monitorare in tempo reale un elevato numero di fenomeni, come l'uso di biciclette pubbliche, il conteggio delle auto attraverso l'uso di semafori "intelligenti", gli spostamenti delle folle, il traffico pedonale e il tasso di occupazione dei parcheggi comunali, ma anche numerosi fattori esterni come le condizioni meteorologiche e la presenza di eventi pubblici.

Se da un lato la disponibilità di dati real-time offre una preziosa risorsa per costruire modelli accurati, dall'altro gestire tale flusso di dati potrebbe essere particolarmente impegnativo, dato che spesso i sensori distribuiti in città sono soggetti a guasti e malfunzionamenti. Pertanto diventa essenziale adottare tecniche specifiche per far fronte a tali situazioni.
Molte delle informazioni raccolte dai sistemi di rilevatori sono state utilizzate durante il processo di features selection, al fine di estrarre le informazioni più rilevanti per gestire il problema d'interesse.
Nello specifico i dati utilizzati sono stati:

- **Dati meteo:** più precisamente il tasso di irraggiamento, pioggia, neve, temperatura e velocità del vento;
- **Eventi pubblici:** festival, eventi, concerti, parate che si sono svolti in corrispondenza dei giorni delle varie osservazioni;
- **Giorni di vacanza:** booleano per indicare feste nazionali e fine settimana;
- **Mese:** mese dell'anno da gennaio a dicembre;
- **Giorno della settimana:** giorno della settimana da lunedì a domenica;
- **Ora ciclica:** considerando che le 24 ore della giornata sono in realtà cicliche e non lineari, è stata effettuata una trasformazione sull'orario delle osservazioni. L'idea è quella di convertire la variabile oraria in due diverse variabili (due dimensioni) attraverso delle trasformazioni goniometriche[1]:

$$x_{sin} = sin\frac{2 * \pi * x}{max(x)} \tag{4.3.1}$$

$$x_{cos} = sin\frac{2 * \pi * x}{max(x)} \tag{4.3.2}$$

  dove $max(x)$ rappresent l'ora massima rilevata, ovvero mezzanotte (24).
  Così facendo si evita di avere una discontinuità di salto alla fine di ogni giornata, quando il valore dell'ora passa da 23 : 00 a 00 : 00, e si ottiene una codifica in formato ciclico migliore per il modello.

[1] https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning

Le variabili dipendenti si distinguono a seconda del fenomeno preso in considerazione in:

- Numero di pedoni che attraversano determinate connessioni stradali di 's-Hertogenbosch;
- Disponibilità di posti auto in determinati parcheggi di 's-Hertogenbosch;
- Disponibilità di posti auto in determinati parcheggi di Breda;
- Disponibilità di posti in determinati parcheggi adibiti a biciclette di Breda.

Per ottenere una prima visione generale dell'andamento dei vari fenomeni sono state eseguite delle semplici analisi preliminari sintetizzate attraverso grafici e tabelle. Prima di tutto dai grafici a barre 4.1, 4.2, 4.3 e 4.4 si può notare il variare dei valori delle variabili dipendenti in funzione della presenza o meno di eventi particolari nei centri città.

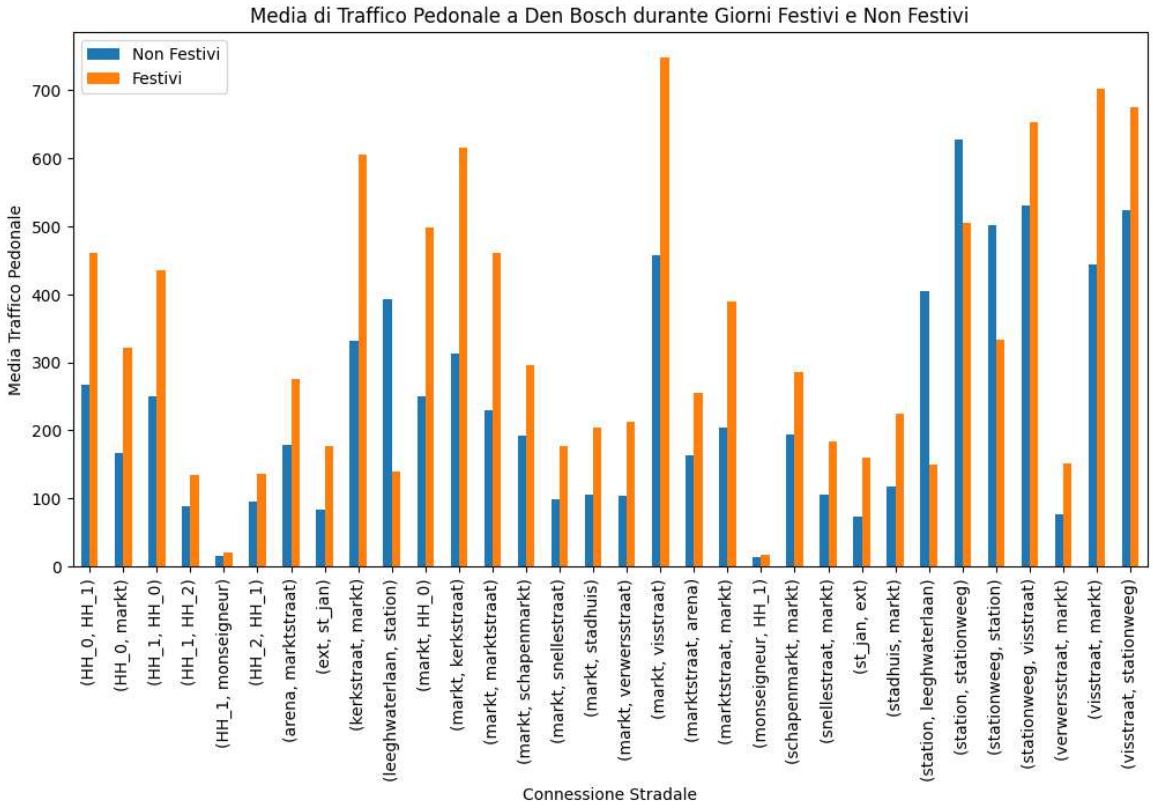

Figura 4.1: Media di traffico pedonale a Den Bosch durante giorni festivi e non festivi.

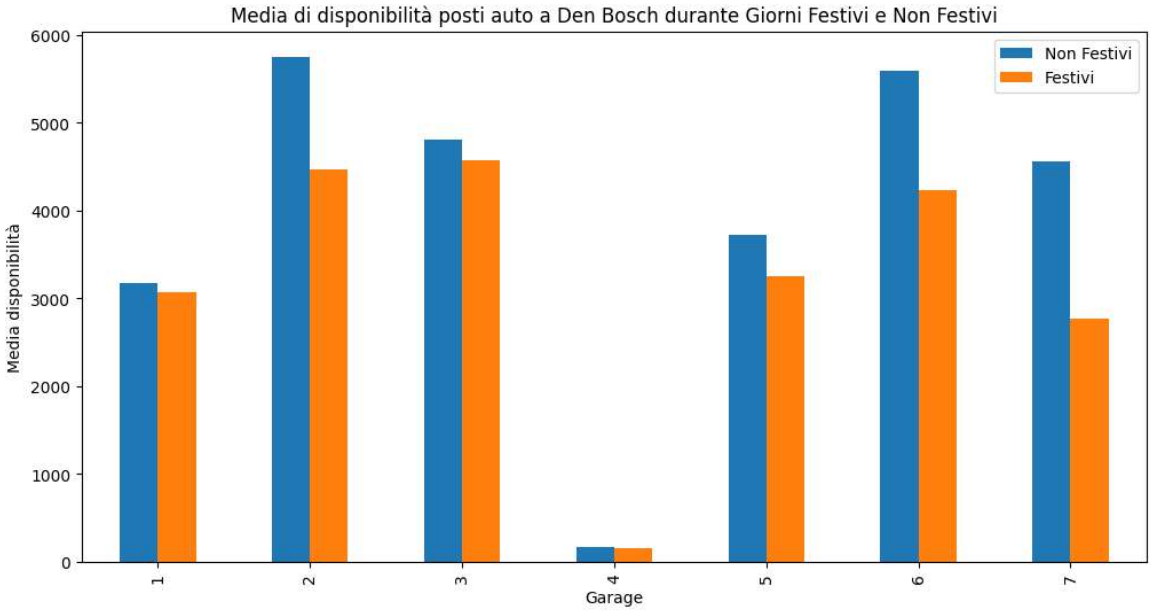

Figura 4.2: Media di disponibilità posti auto nei parcheggi di Den Bosch durante giorni festivi e non festivi.

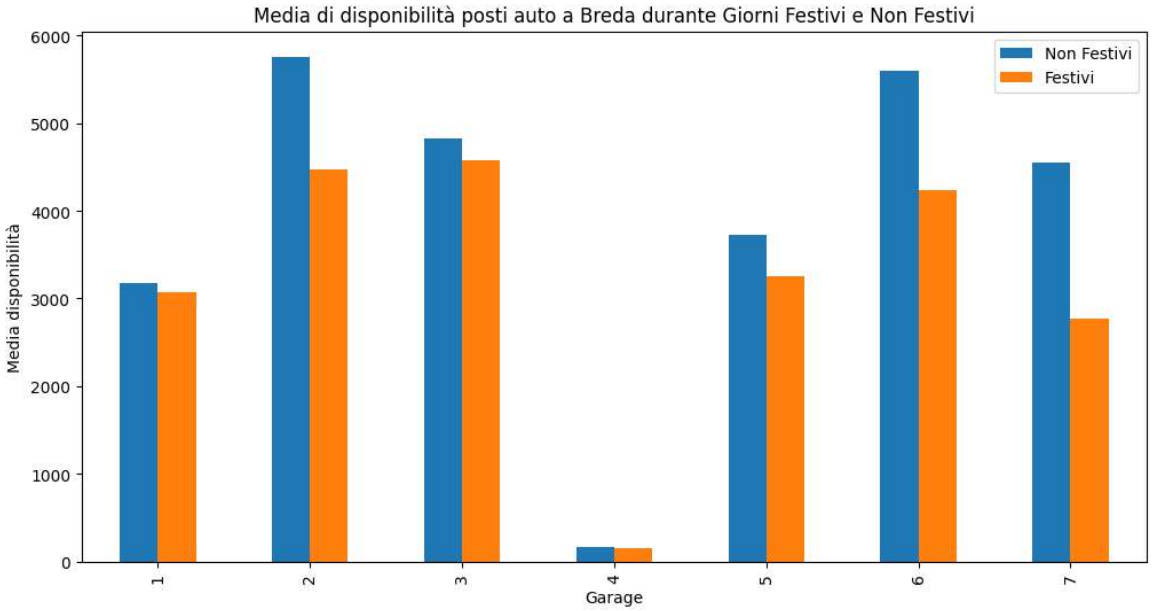

Figura 4.3: Media di disponibilità posti auto nei parcheggi di Breda durante giorni festivi e non festivi.

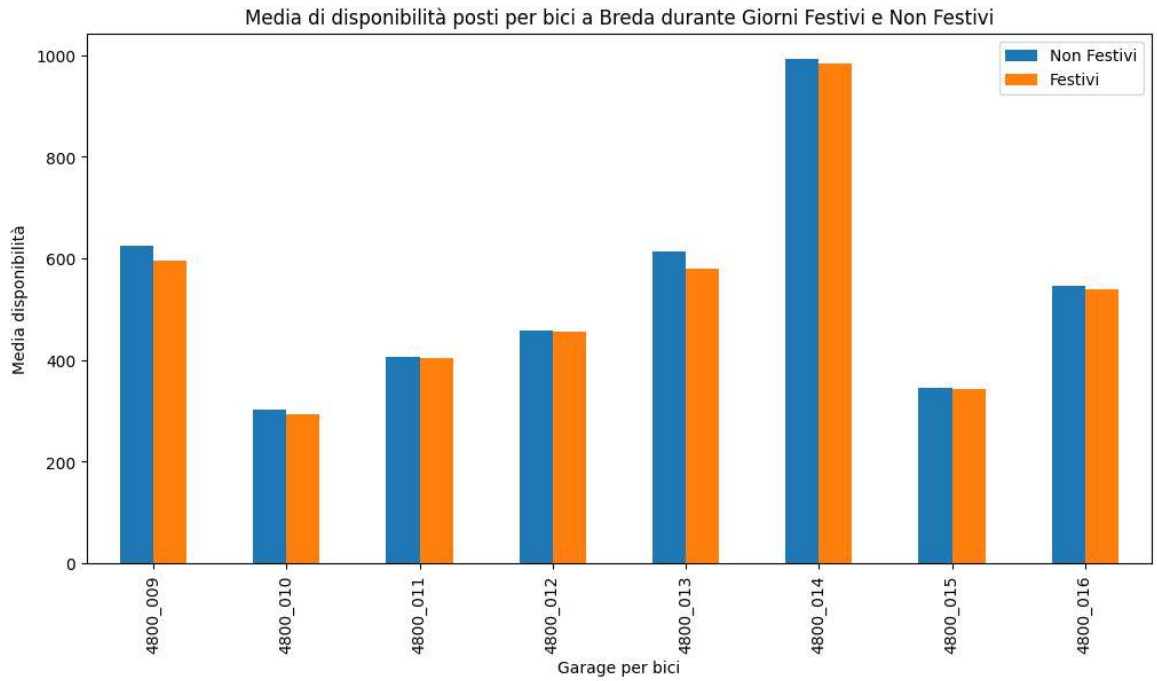

Figura 4.4: Media di disponibilità posti per biciclette nei parcheggi di Breda durante giorni festivi e non festivi.

Dai grafici si nota come il fenomeno che maggiormente risente della differenza tra giorni festivi e non sia il traffico pedonale, nello specifico nei tratti stradali che coinvolgono la zona del mercato. Vi è una differenza meno accentuata per quanto riguarda la disponibilità di posti auto, mentre non sembra sussistere una differenza rilevante per quanto concerne il tasso di occupazione dei parcheggi adibiti a sole biciclette di Breda.

Dopo la fase di feature engineering, che ha permesso di costruire i dataset di partenza, si sono rese necessarie alcune operazioni di cleaning al fine di rimuovere eventuali outliers e imputare i valori nulli in corrispondenza delle variabili indipendenti e della variabile target. La presenza di valori nulli all'interno dei dataset tipicamente crea molti problemi, essendo essi porzioni di informazione mancanti. Per imputazione di valori nulli si intende indicare il processo di riempimento dei campi vuoti con dei valori che preservino la consistenza semantica dell'intero dataset. Per comprendere l'andamento generale delle variabili sono state anche calcolate delle statistiche generali sintetizzate nelle tabelle 4.1, 4.2, 4.3, 4.4 e sulla base di queste ultime sono state scelte le tecniche per trasformare i dati dei dataset.

Tabella 4.1: Sintesi statistica sul traffico pedonale a Den Bosch.

	Pedestrians	Temperature	Wind Speed	Radiation	Rain	Snow
Mean	267.79	10.50	3.99	49.52	0.21	0.01
Std	468.27	7.11	2.16	82.04	0.41	0.12
Min	0.00	-8.50	0.00	0.00	0.00	0.00
Max	11076.00	31.50	14.00	344.00	1.00	1.00

Tabella 4.2: Sintesi statistica sulla disponibilità di parcheggi a Den Bosch.

	Available	Temperature	Wind Speed	Radiation	Rain	Snow
Mean	3715.61	8.86	3.64	51.55	0.22	0.02
Std	2942.74	4.88	2.00	75.67	0.42	0.12
Min	0.00	-2.80	0.00	0.00	0.00	0.00
Max	15035.00	22.90	11.00	344.00	1.00	1.00

Tabella 4.3: Sintesi statistica sulla disponibilità di parcheggi a Breda.

	Available	Temperature	Wind Speed	Radiation	Rain	Snow
Mean	3715.94	8.46	3.68	49.38	0.25	0.01
Std	2944.03	4.86	1.85	73.55	0.43	0.10
Min	0.00	-4.40	0.00	0.00	0.00	0.00
Max	15035.00	23.80	11.00	323.00	1.00	1.00

Tabella 4.4: Sintesi statistica sulla disponibilità di parcheggi per bici a Breda.

	Available	Temperature	Wind Speed	Radiation	Rain	Snow
Mean	495.04	7.65	3.76	38.92	0.23	0.01
Std	170.74	5.42	1.93	67.51	0.42	0.11
Min	5.00	-8.20	0.00	0.00	0.00	0.00
Max	1384.00	23.80	11.00	323.00	1.00	1.00

Una volta separate variabile dipendente e variabili esogene , è stato possibile applicare le pipeline di trasformazioni ad-hoc previste.
Le trasformazioni specifiche eseguite per ogni set di variabili sono state le seguenti:

- **Variabili indipendenti:** per trasformare le variabili esogene sono state attuate le seguenti trasformazioni:

  – i valori numerici mancanti sono stati imputati con la media e scalati con un approccio MinMax.
  In un approccio di min-max scaling le variabili vengono normalizzate in un intervallo $[0, 1]$, mediante la seguente formula:

  $$x_{scalato} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.3.3}$$

  dove x_{max} e x_{min} sono rispettivamente il valore massimo e minimo che la variabile di riferimento assume;

  – i valori booleani mancanti sono stati imputati con il valore più frequente, essendo gli eventi speciali e i giorni di festa meno comuni, rispetto quelli feriali;

  Su tutti gli attributi categorici, come il giorno della settimana o il mese, è stata poi applicata la trasformazione nota come "One-Hot-Encoding", con la quale si definisce, per ogni possibile valore dell'attributo, una variabile binaria che assume valore 1 solo in corrispondenza dello specifico attributo categorico che essa rappresenta. Tale trasformazione è necessaria per rendere i dati idonei a modelli di Machine Learning che non supportano l'uso di variabili nominali;

- **Variabile dipendente:** per tutte le variabili target i valori mancanti sono stati imputati utilizzando la media per evitare l'alterazione dei fenomeni e i valori 0 sono stati sostituiti con un valore positivo molto piccolo pari a 1, per poter successivamente applicare una trasformazione BoxCox.
  Una trasformazione BoxCox permette di i rendere dati distorti in una distribuzione normale, tuttavia essa non è applicabile quando sono presenti valori negativi o pari a 0, ma può migliorare l'accuratezza delle predizioni effettuate attraverso l'uso di modelli regressivi.
  La tabella 4.5 sintetizza le percentuali di valori nulli in corrispondenza delle variabili target dei fenomeni analizzati durante questo lavoro di tesi.

Tabella 4.5: Percentuali valori nulli su variabili dipendenti.

Fenomeno	**Variabile**	**Valori Mancanti**
Traffico Pedonale (Den Bosch)	Pedestrians	18.37%
Disponibilità Parcheggi (Den Bosch)	Available	24.42%
Disponibilità Parcheggi (Breda)	Available	24.42%
Disponibilità Parcheggi per Bici (Breda)	Available	25.56%

4.4 Ciclo di vita

Portare un sistema di Machine Learning in produzione è una fase cruciale, dalla quale derivano le reali probabilità di successo di un progetto. Tipicamente, dopo aver distribuito un progetto di questo tipo, si possono verificare diminuzioni delle prestazioni del modello che rendono necessari addestramenti e aggiornamenti periodici e continui per mantenerlo stabile nel tempo.
Nella sezione 2.2 è stato introdotto il concetto di MLOps, una metodologia per il Machine Learning engineering che unifica lo sviluppo della componente ML con la componente operazionale. Applicare tale metodologia non è sempre facile perché richiede la configurazione di strumenti adatti a diversi fattori come il caso d'uso, le competenze del team, le normative e le risorse disponibili. Inoltre, richiede la collaborazione tra ruoli con competenze diverse e comporta sfide nell'interazione tra gli strumenti utilizzati all'interno del flusso di lavoro. Nello specifico, per integrare le componenti di Machine Learning in una prima versione dell'ambiente di produzione, nel quale offrire degli output utili e validi, è stato utilizzato MLFlow. L'obiettivo finale è quello non solo di fornire un modello addestrato, ma rendere codici ed esperimenti riproducibili, gestire in maniera semi-automatica le variazioni delle prestazioni, automatizzare l'utilizzo in produzione della miglior pipeline sperimentata e separare logicamente la fase di sperimentazione da quella di produzione.
MLFlow [33] è una piattaforma open-source multi-linguaggio che consente la gestione dei flussi di lavoro e degli artefatti durante l'intero ciclo di vita di un progetto di Machine Learning. È possibile ricorrere ad MLFlow per registrare i parametri e le metriche di valutazione dei vari esperimenti condotti, facilitando l'analisi dei risultati e l'esplorazione delle soluzioni tramite un'interfaccia utente intuitiva. Inoltre, la stessa interfaccia permette di confrontare le prestazioni di diversi modelli e selezionare il migliore per il deployment, attraverso la registrazione del modello nel MLflow Registry per monitorarne le prestazioni in produzione. Infine, permette ai Data Scientist e Software Engineer di organizzare il proprio codice in maniera ottimale, semplificando la condivisione e l'esecuzione, consentendo di "impacchettare" una componente di Machine Learning in modo riutilizzabile e riproducibile, promuovendo una migliore collaborazione all'interno del team.

Per garantire una gestione completa, MLflow fornisce un Model Store centrale per la gestione dei modelli condivisi. In particolare, per archiviare e tracciare il modello, è necessario registrare l'ambiente in cui è stato addestrato e le sue dipendenze, nonché definire un nome univoco, le versioni e altre informazioni aggiuntive come lo stage (ad esempio per rappresentare la fase di produzione).

Il ciclo di vita del software, le modalità in cui esso opera e il processo di automazione della pipeline d'inferenza può essere espressa in maniera succinta attraverso la Figura 4.6.

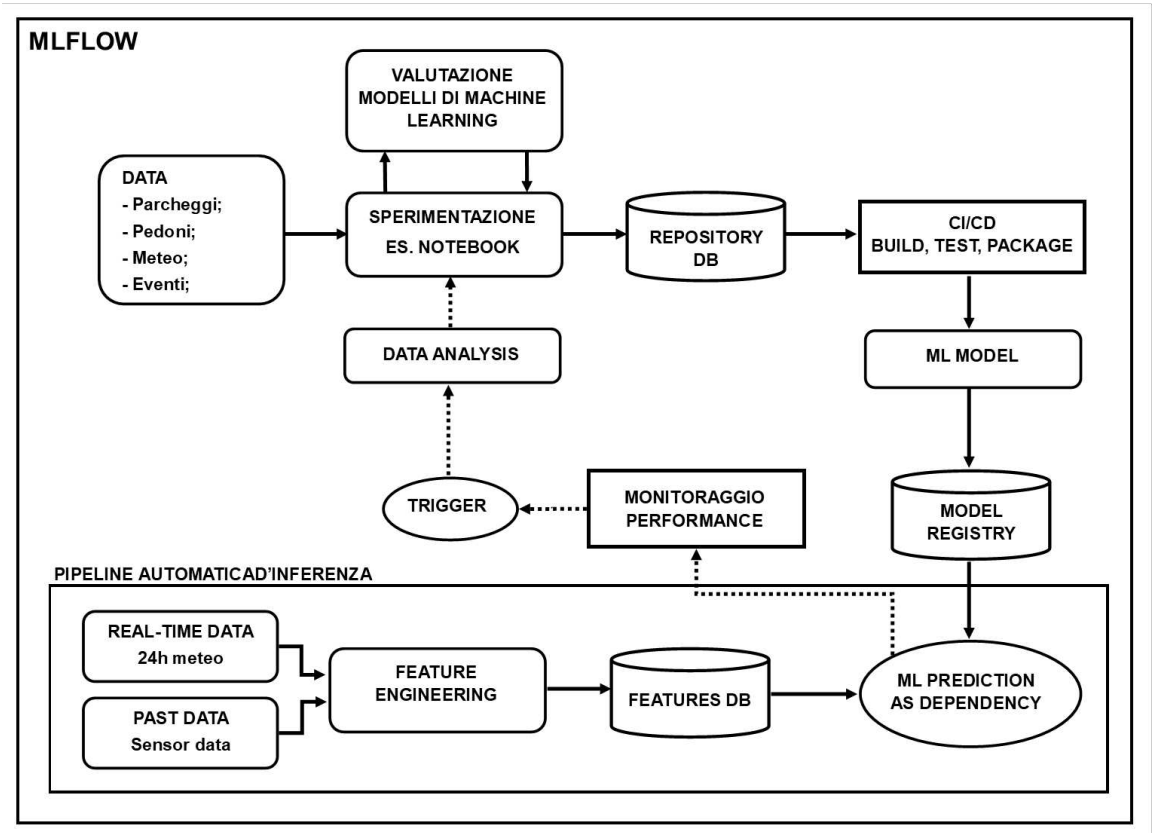

Figura 4.5: Architettura del sistema.

Il primo stadio del ciclo è costituito da un processo manuale di Data Science. Questa fase ha una natura sperimentale ed iterativa ed è quella che richiede maggior tempo. I dati a disposizione vengono analizzati, processati e validati al fine di configurare e testare diverse soluzioni di Machine Learning.

Le sessioni sperimentali vengono tracciate per salvare tutte le informazioni e le pipeline eseguibili in una repository comune per favorire la riproduzione, riesecuzione, condivisione e confronto dei diversi esperimenti, migliorando il processo di sviluppo. Si vanno, inoltre, a registrare alcuni elementi chiave tra i quali i parametri e le condizioni di addestramento, le metriche di valutazione e la localizzazione degli artefatti. Durante questa fase sono stati utilizzati strumenti di sviluppo rapidi, come i Jupiter Notebooks [34]. L'obiettivo è quello di costruire e testare pipeline di addestramento distribuibili in un ambiente specifico. Vengono effettuati dei test sulle singole componenti e sulla pipeline nella sua interezza (CI). Le pipeline testate vengono successivamente distribuite, in un ambiente target, prima di essere trasferite nell'ambiente di produzione (CD).

Completata la fase sperimentale, identificati e addestrati i modelli più adatti, questi ultimi sono pronti per essere memorizzati in un cosiddetto Model Registry.
Quest'ultimo conterrà, oltre al modello addestrato, la versione, lo stage, un alias e una serie di annotazioni aggiuntive. I modelli sono dunque pronti per il deployment. Durante il deployment dei sistemi, è importante considerare alcuni elementi cruciali e valutare la robustezza della strategia. Bisogna verificare se esiste già un modello in produzione da sostituire, se c'è un piano di rollback in caso di bug nel nuovo deployment e valutare quali sono le conseguenze negative se il modello non raggiunge l'accuratezza richiesta.

Nel processo di previsione, dopo che il modello viene distribuito nell'ambiente target, il servizio del modello inizia ad accettare richieste e a fornire risposte con previsioni. Viene quindi applicato il pattern del Model-as-Dependency, dove il modello viene incluso come una dipendenza all'interno dell'applicazione software che ne fa uso. Ad esempio l'applicazione utilizza il modello come una dipendenza convenzionale, invocando i metodi necessari per le previsioni e passando i parametri di input.

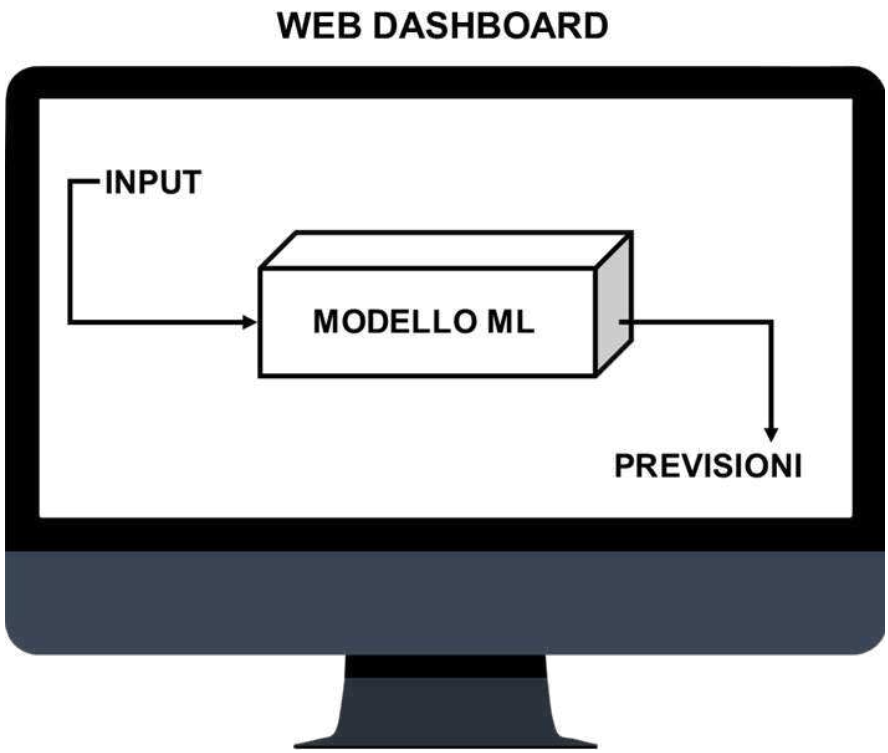

Figura 4.6: Model-as-Dependency.

In maniera parallela a quanto descritto vi è l'implementazione ed applicazione delle pipeline automatiche per l'esecuzione delle predizioni. Durante questa fase la miglior pipeline sperimentale è automatizzata e utilizzata in produzione per gestire eventuali richieste. Ogni qualvolta viene effettuata una richiesta di predizioni, vengono prima di tutto caricati i dati real-time. Viene effettuata una richiesta all'API per ottenere le previsioni meteo, che vengono poi processate e ripulite al fine di renderle conformi ai dati di training. Allo stesso tempo vengono caricati i dati pregressi dei fenomeni in esame, al fine di estrarre le feature temporali, per poi procedere con il processo di feature engineering, durante il quale viene eseguita la pipeline per costruire i dati di input, che vengono poi memorizzati in un database e utilizzati, insieme al modello, per servire risultati. Durante l'esecuzioni di tutte le fasi, si rende necessario un monitoraggio delle performance per verificare regolarmente che le prestazioni del modello non si deteriorino in maniera eccessiva. L'approccio utilizzato consiste nel verificare regolarmente e manualmente il comportamento dei modelli in modo da innescare un trigger on-demand, per richiedere la ri-esecuzione della pipeline di sperimentazione e addestramento.

CAPITOLO 5

's-Hertogenbosch

In questo capitolo verranno descritti gli approcci utilizzati e analizzati i risultati ottenuti dalle sessioni sperimentali eseguite sul caso di studio riguardante la città di 's-Hertogenbosch. Verranno indicati i modelli testati, le strategie adottate e le prestazioni ottenute in termini di accuratezza delle previsioni.

5.1 Contesto urbano

's-Hertogenbosch, tipicamente chiamata anche "Den Bosch", è la capitale della provincia del Brabante Settentrionale nei Paesi Bassi (Figura 5.1).
La città ha molti edifici storici, tra cui la Sint Jans Kathedraal, il palazzo comunale, monumenti storico-artistici e presenta svariate attività commerciali, rendendola una meta turistica molto amata.
La stazione di 's-Hertogenbosch ha una buona collocazione all'interno del sistema ferroviario olandese, ed è caratterizzata giornalmente da un elevato traffico pedonale, in modo particolare durante le ore di punta. Sono molti i parcheggi in città, sia al coperto che all'aperto, sono ben distribuiti ed è spesso la scelta più consigliata se si vuole raggiungere la città in auto.

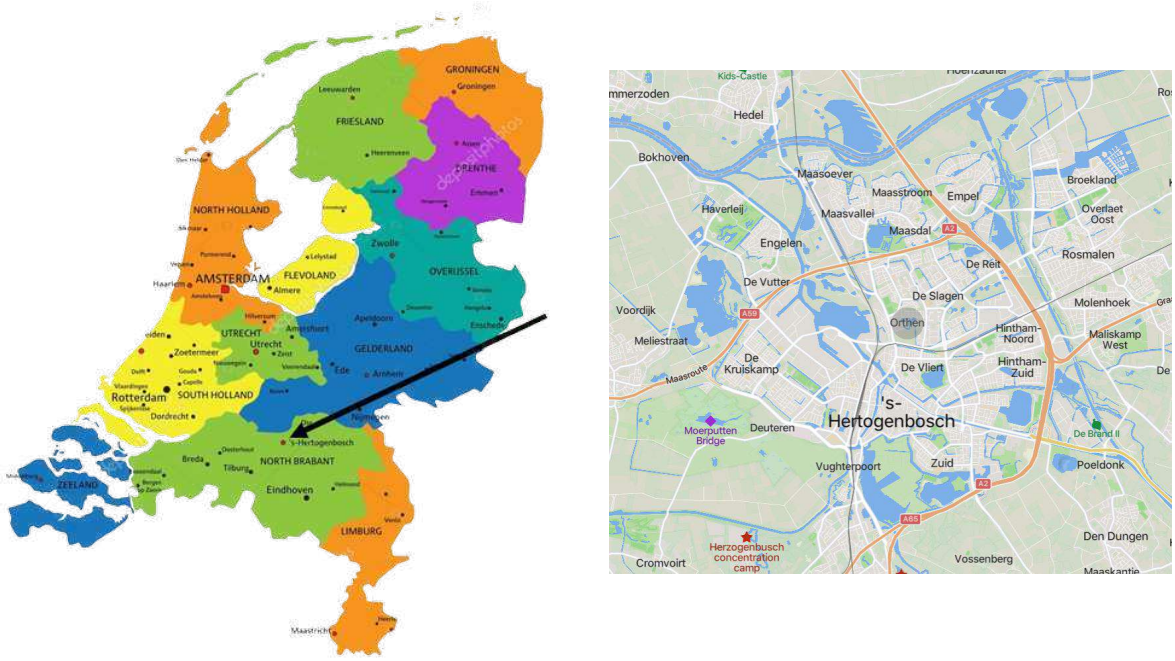

Figura 5.1: Posizione di Den Bosch rispetto i Paesi Bassi e mappa della città.

5.2 Traffico pedonale

Figura 5.2: Distribuzione delle telecamere per il conteggio pedonale nella città di 's-Hertogenbosch.

Le telecamere per il conteggio dei pedoni, la cui distribuzione è mostrata in Figura 5.2, sono state installate intorno al Grote Markt (la piazza principale) e in una serie di altri punti strategici, consentendo al Comune di vedere quante persone entrano ed escono dall'area. L'insieme di telecamere installate nella città formano una gerarchia mostrata in Figura 5.3. In particolare, ogni arco presente nel grafo in figura rappresenta l'orientamento dei sensori. Ad esempio un arco che punta da "Station" a "Stationweeg", indica la presenza di una telecamera che traccia il flusso pedonale passante per la suddetta direzione.

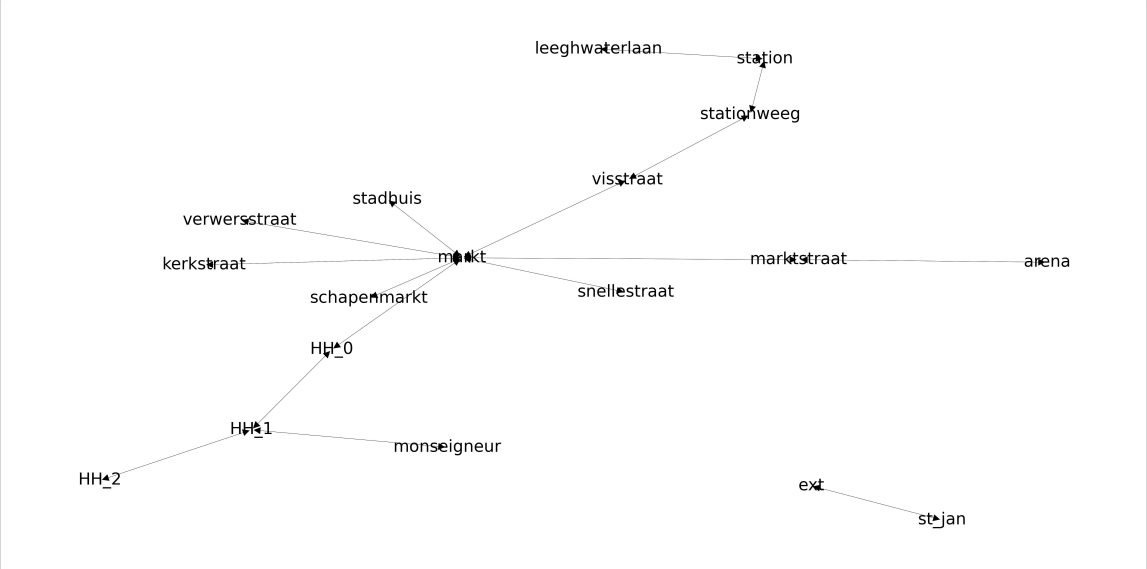

Figura 5.3: Gerarchia formata dalle telecamere per il tracciamento dei pedoni in 's-Hertogenbosch.

Nella totalità delle connessioni è importante sottolineare il legame tra le diverse connessioni. Difatti nel modellare il problema del traffico pedonale vanno considerate tutte le connessioni tra i vari nodi, ma a loro volta le connessioni che suddette hanno con altri nodi.

É, ad esempio, lecito pensare che nell'analizzare il traffico pedonale da "Station" a "Stationweeg", vada preso in considerazione anche quello da "Leeghwaterlaan" verso "Station" e così via.

Come già anticipato nella sezione 2.4 per gestire tale situazione è stata adottata la tecnica del "Hierarchical TimeSeries Reconciliation", che permette in fase di training di valutare, ad ogni istante di tempo, tutte le connessioni relative ad un dato tratto.

I dati relativi al traffico pedonale possono essere pensati come un ibrido tra i dati cosiddetti "Panel", descritti nella sezione dedicata alle serie temporali, e quelli "Gerarchici". Ogni nodo è un elemento del "Panel" e ogni arco forma un livello nella gerarchia del nodo, un esempio visivo in Figura 5.4.

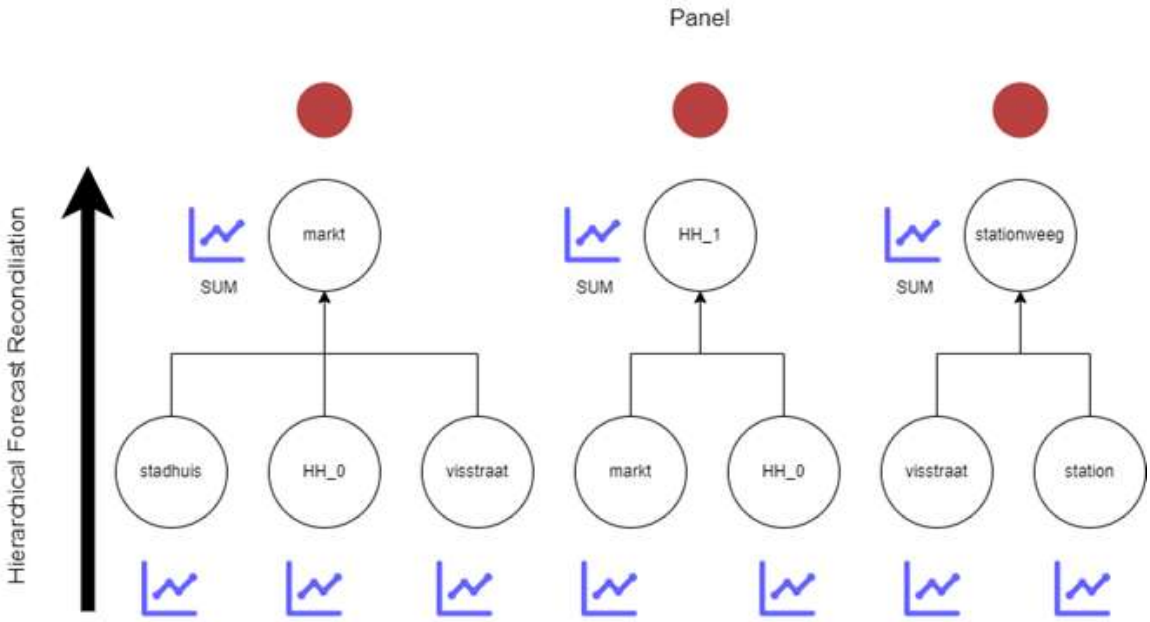

Figura 5.4: Esempio di Hierarchical TimeSeries Reconciliation per il traffico pedonale di 's-hertogenbosch.

Durante le varie sessioni di training è stata utilizzata la libreria Python *SKtime* [35], un framework open source facile da usare, flessibile e modulare per un'ampia gamma di attività di apprendimento automatico delle serie temporali. Offre interfacce compatibili con scikit-learn e strumenti di composizione del modello, con l'obiettivo di rendere l'ecosistema più utilizzabile e interoperabile nel suo insieme. Innanzitutto variabili dipendenti e indipendenti sono state aggregate, mantenendone i singoli livelli. Successivamente i dati sono stati divisi in insieme di addestramento e di test, ovviamente nel fare ciò è fondamentale non agire in maniera casuale, ma mantenere la caratteristica cronologica della serie.

Nel definire la pipeline di addestramento è stata considerata la struttura gerarchica dei dati, con l'obiettivo di applicare un regressore ad ogni serie della struttura in maniera ricorsiva, così che ogni previsione potesse essere usata per quelle future.
Le previsioni sono state poi riconciliate col fine di generare previsioni indipendenti sotto forma di serie MultiOutput, dove ogni output corrisponde ad uno specifico nodo della gerarchia.

5.2.1 XGBoost

Inizialmente è stato utilizzato Extreme Gradient Boosting per eseguire la previsione del traffico pedonale.
XGBoost offre una buona precisione su una grande varietà di previsioni di serie temporali, da qui l'idea di sperimentarne l'utilizzo.

L'aspetto indubbiamente più complesso che scaturisce dall'uso di un tale modello è la definizione degli iperparametri necessari ad addestrarlo.
Per applicare la cross-validation e la definizione degli iperparametri *SKtime* offre due utili classi:

- **ExpandingWindowSplitter:** per dividere ripetutamente le serie temporali in un insieme di allenamento crescente e in un insieme di test di dimensioni fisse, creando folds di cross-validation in maniera logica come mostrato in Figura 5.5:

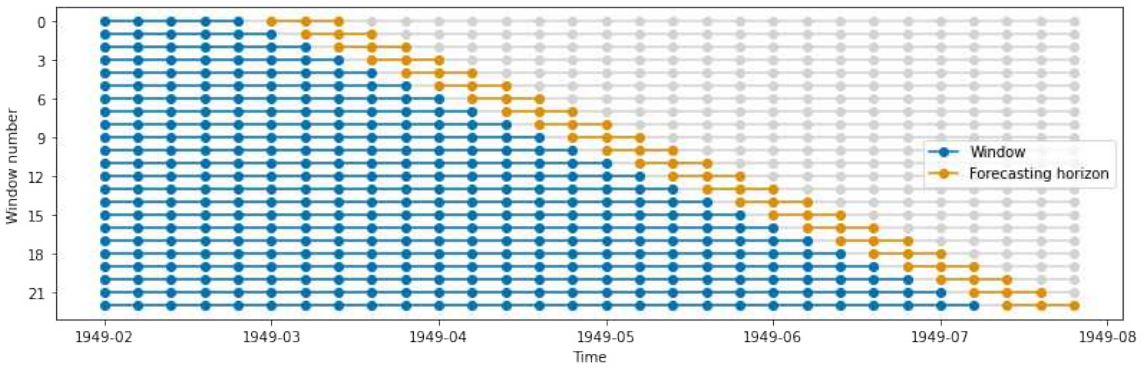

Figura 5.5: Esempio di ExpandingWindowSplitter

- **ForecastingGridSearchCV:** per applicare una grid-search in combinazione alla cross-validation, andando a definire una griglia di parametri candidati, utilizzando l'errore medio quadratico come metrica di valutazione;

Le sessioni di addestramento sono state effettuate su un intervallo di dati che copre gli ultimi 200 giorni a partire da quello corrente.
L'addestramento di un modello così definito può risultare computazionalmente costoso, in quanto necessita di molto tempo per eseguire tutti i sotto-task di training. Tra tutte le sessioni di addestramento quella che ha impiegato maggior tempo è stata di $1d\ 1h\ 12min\ 23s$ utilizzando 6 jobs.

5.2.2 ARIMA

L'approccio utilizzato per modellare il problema utilizzando un modello ARIMA è stato pressocchè simile a quello appena descritto, la differenza significativa è la modalità con cui sono stati definiti gli iperparametri.

Essendo ARIMA un tipo di modello lineare che si basa sui valori passati per fare previsioni sui valori futuri, è fondamentale un'analisi preliminare della serie per andare ad identificare i giusti termini di integrazione, media mobile e autoregressione. Un'approfondita descrizione del funzionamento del modello è stata già definita nella sezione 2.4.2, pertanto in questo paragrafo verranno descritti i valori ottenuti.

Le analisi preliminari, così come le sessioni di addestramento, sono state effettuate su un intervallo di dati che copre gli ultimi 200 giorni a partire da quello corrente.
I dati vengono estratti in tempo reale dal database con un intervallo di tempo orario.
Di seguito le informazioni estratte:

- **Termine di integrazione:** Eseguendo il $p - value$ del test Augmented Dickey Fuller pari a 0, pertanto inferiore al livello di significatività di 0.05, si può rigettare l'ipotesi nulla e inferire che la serie è già stazionaria.
  Il valore del termine di integrazione è definito pari a 0;

- **Termine di autoregressione:** Per trovare il numero di termini autoregressivi necessari al modello, si può osservare il grafico dell'autocorrelazione parziale (PACF) in Figura 5.6. Il PACF rappresenta la correlazione pura tra un lag e la serie, dopo aver escluso i contributi dei lag intermedi. Considerando la

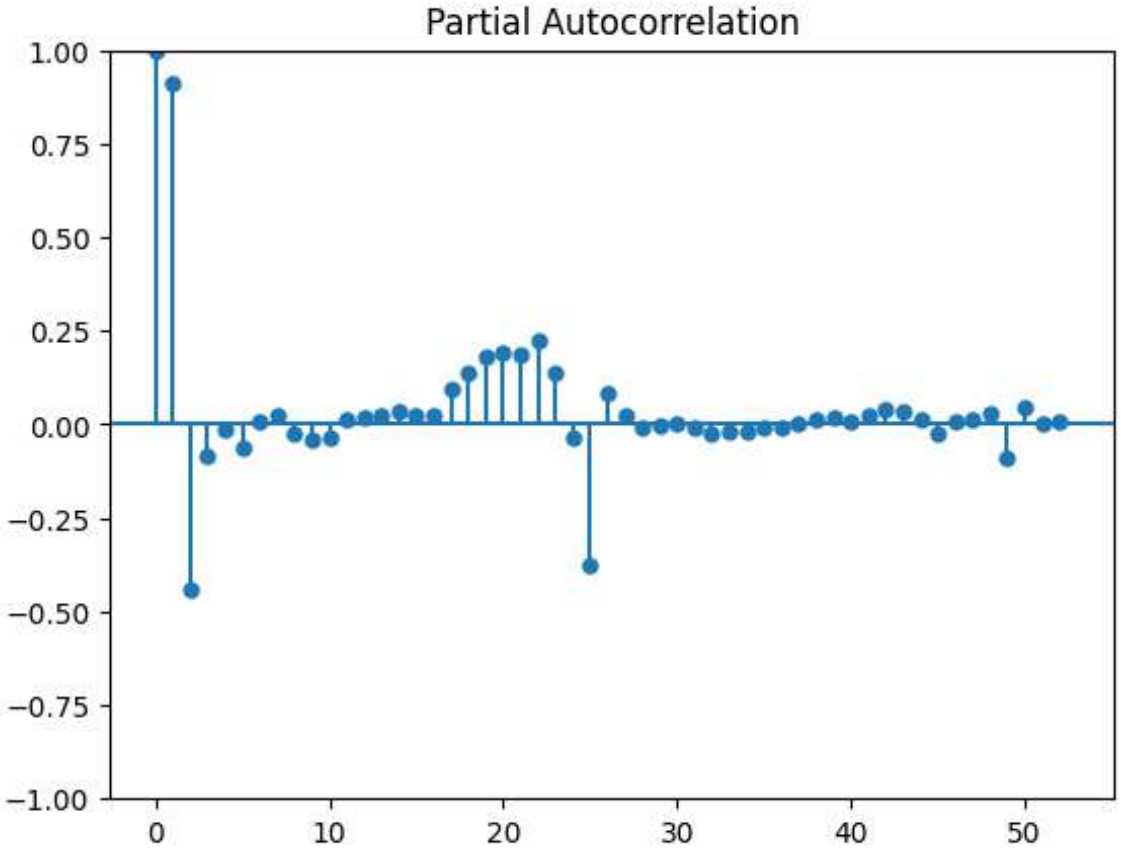

Figura 5.6: Grafico di autocorrelazione parziale per la definizione del termine di autoregressione di un modello ARIMA

linea retta al lag 1 nel grafico PACF, possiamo notare che esiste un'elevata correlazione tra la serie temporale e il suo primo valore ritardato. Ciò suggerisce che un modello autoregressivo potrebbe essere appropriato, e permette di restringere il range di valori candidati, come termini di auto regressione, ad un intervallo $1 - 4$;

- **Termine di media mobile:** Per trovare il numero di termini di media mobile si può fare riferimento al grafico di autocorrelazione in Figura 5.7. I valori positivi indicano una correlazione positiva tra la serie temporale e i suoi valori ritardati, mentre i valori negativi indicano una correlazione negativa.

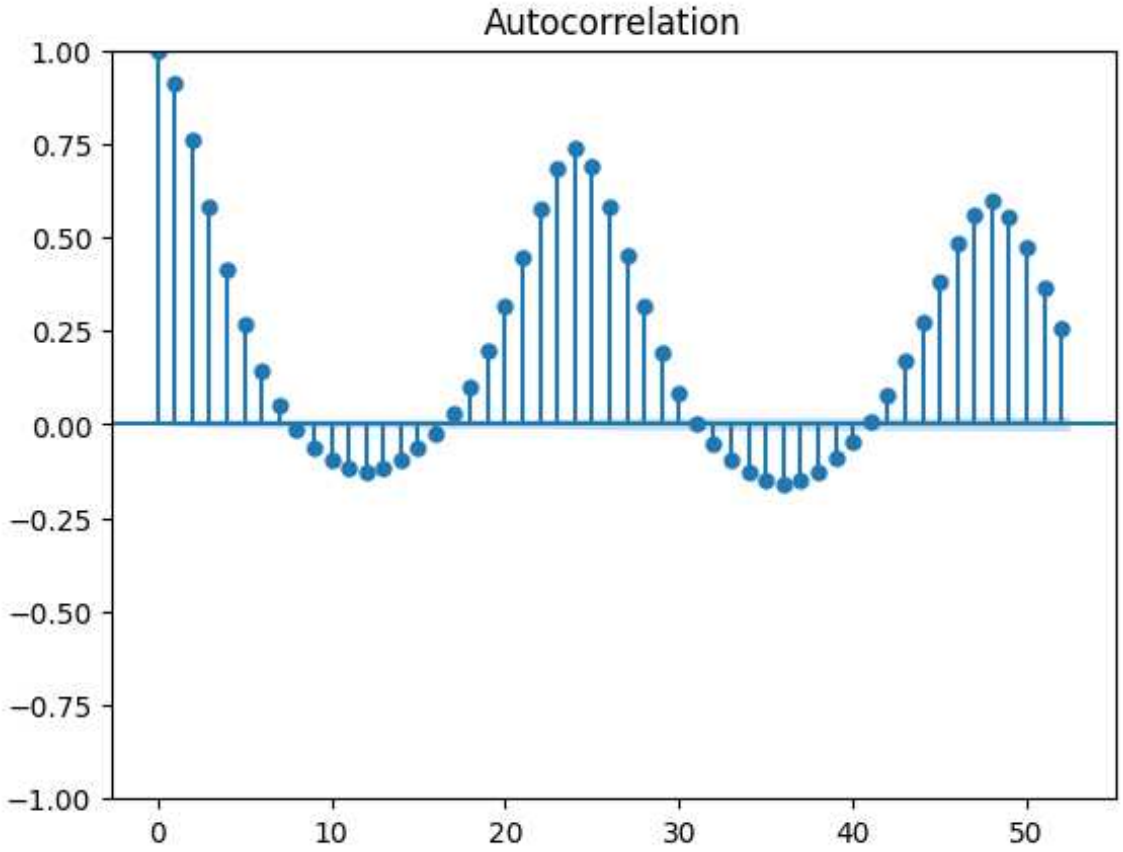

Figura 5.7: Grafico di autocorrelazione per la definizione del termine di media mobile di un modello ARIMA.

I punti positivi che vanno da 0 a 1 indicano una forte correlazione positiva tra la serie temporale e i suoi valori ritardati fino a un ritardo. I punti negativi che vanno da 0 a -0,2 indicano una correlazione negativa più debole tra la serie temporale e i suoi valori ritardati.
Il fatto che ci siano solo pochi punti negativi con valori relativamente piccoli suggerisce che la componente MA potrebbe non essere molto importante per la modellazione della serie temporale, permettendo di restringere il campo per la definizione del termine di media mobile ad un intervallo $1 - 2$. L'aspetto forse più interessante, però, è la forma assunta dal grafico. L'andamento sinusoidale delle autocorrelazioni mostra un evidente legame forte tra ogni osservazione e il valore di queste 24 ore prima (o 24 ore dopo).

Completato lo studio preliminare è stato possibile testarne il funzionamento attraverso lo stesso approccio gerarchico definito in precedenza.

Anche in questo caso è importante sottolineare come il training di un tale modello, che include anche l'uso di variabili esogene, necessiti di una sostanziale quantità di memoria.

SARIMAX

Sfortunatamente non è stato possibile effettuare la sperimentazione di un modello SARIMAX per mancanza di memoria disponibile. Infatti, lanciare l'addestramento di un tale modello ha richiesto più di 16GB di RAM, superando di gran lunga le risorse computazionali messe a disposizione per il progetto.

5.3 Disponibilità di parcheggi auto

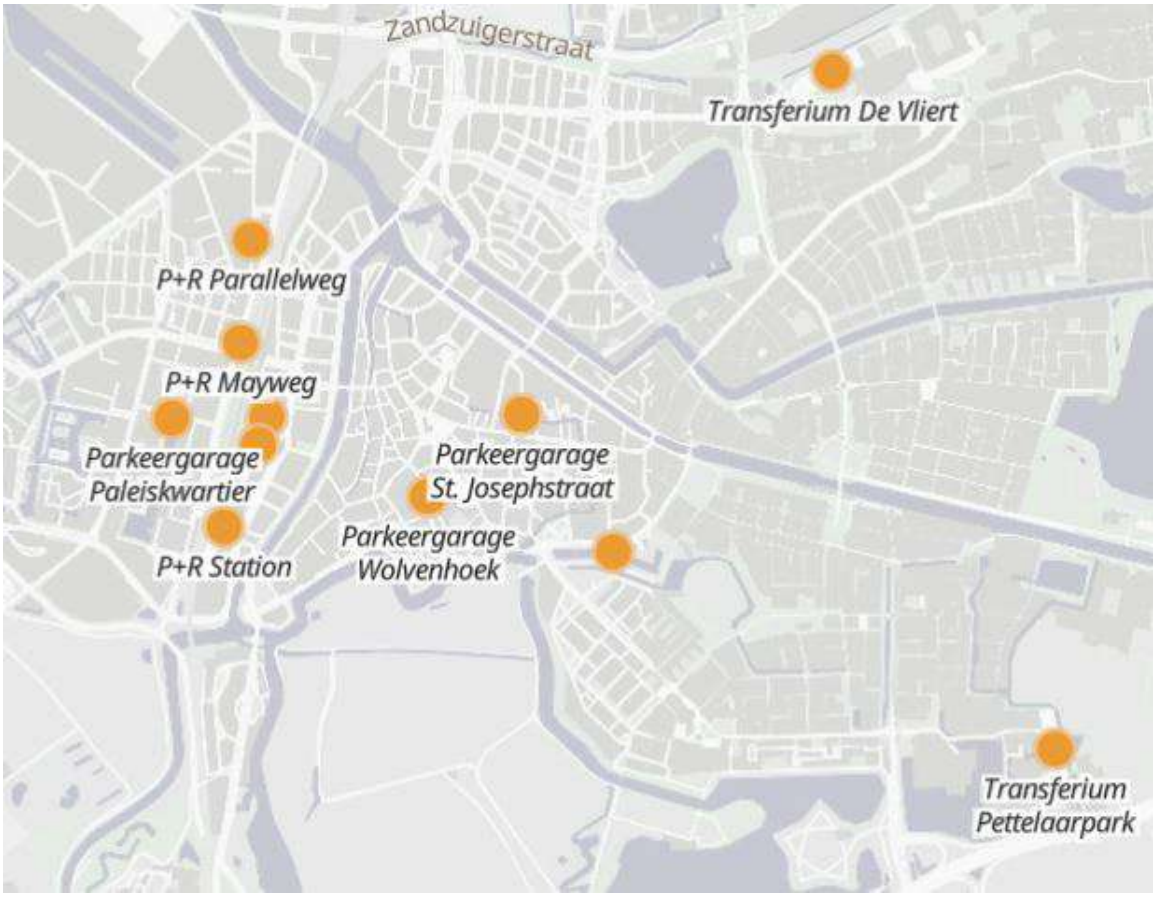

Figura 5.8: Distribuzione dei parcheggi pubblici nella città di 's-Hertogenbosch.

I parcheggi di 's-Hertogenbosch, Figura 5.8, sono maggiormente concentrati nelle aree che delimitano il centro storico, in particolare in prossimità della stazione centrale. Per la predizione della disponibilità di parcheggi a 's-Hertogenbosch, sono stati utilizzati i dati degli ultimi 90 giorni a partire da quello corrente. In Figura 5.9 è mostrata una panoramica dell'andamento delle serie temporali di ciascun parcheggio della città.

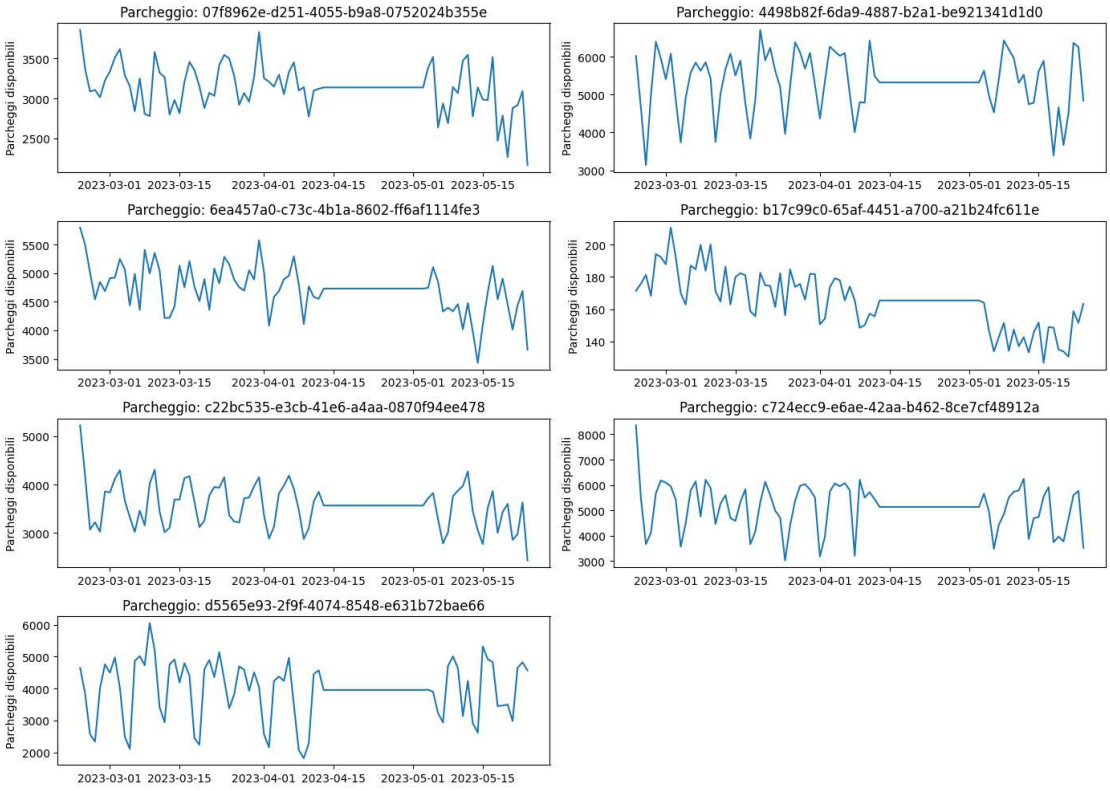

Figura 5.9: Serie temporale dei parcheggi pubblici a 's-Hertogenbosch.

Dai grafici è possibile notare innanzitutto la presenza di un periodo di osservazioni mancanti, che sono state poi imputate con la media e quindi caratterizzate dalla presenza di un segmento retto orizzontale.

Col fine di identificare il modello migliore per gestire il fenomeno in esame, è stato effettuato uno studio riguardo la caratterizzazione della serie temporale, in maniera analoga a quanto fatto per i pedoni.

Di sotto le informazioni ottenute:

- **Stazionarietà:** dal test di Adfuller risulta che la seria è stazionaria e non sono necessari termini di differenziazione;
- **Autocorrelazione parziale:** dal grafico dell'autocorrelazione parziale, in Figura 5.10, risulta una forte relazione tra la serie temporale e il suo primo lag ritardato e una correlazione più bassa col secondo, ma non c'è evidenza significativa di altre correlazioni;

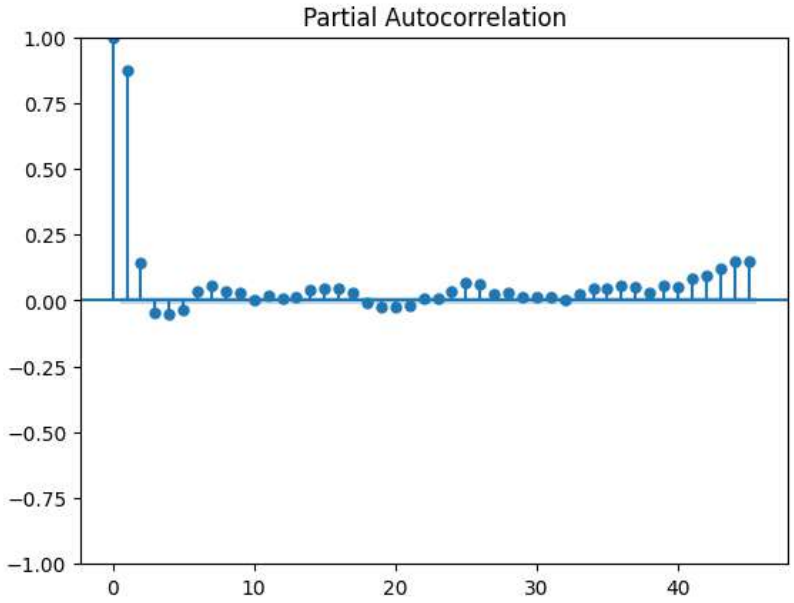

Figura 5.10: Grafico di autocorrelazione parziale dei parcheggi a 's-Hertogenbosch.

- **Autocorrelazione:** analizzando il grafico di autocorrelazione, in Figura 5.11, si nota una caratteristica forma a vasca da bagno, che mostra valori elevati di autocorrelazione all'inizio e alla fine del grafico, con valori bassi nel mezzo. Questa caratteristica suggerisce che i dati mostrano una certa stagionalità, con una ripetizione giornaliera (ogni 48 osservazioni di 30 minuti);

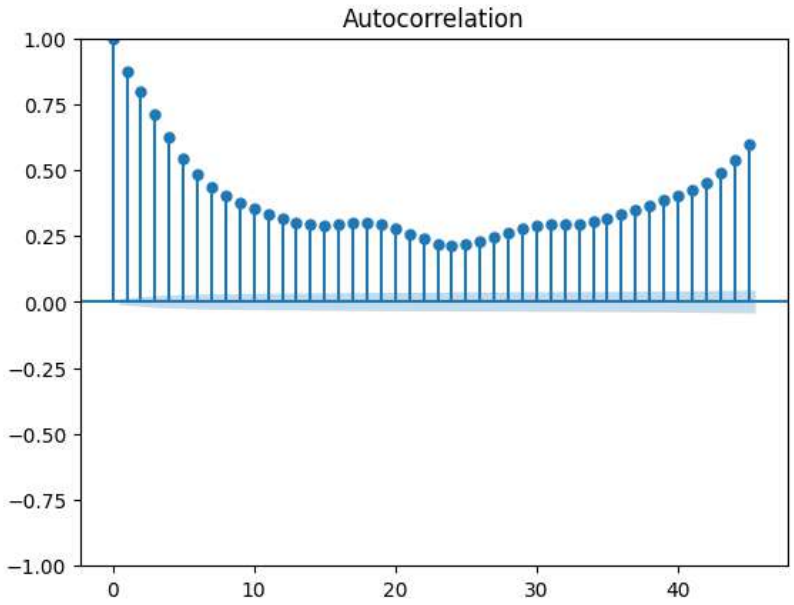

Figura 5.11: Grafico di autocorrelazione dei parcheggi pubblici a 's-Hertogenbosch.

Analizzando il trend della serie temporale, rappresentato in Figura 5.12, si può notare che effettivamente esista un certo pattern settimanale e giornaliero all'interno dei dati, suggerendo la possibilità di modellare il problema attraverso l'uso di un modello SARIMA.

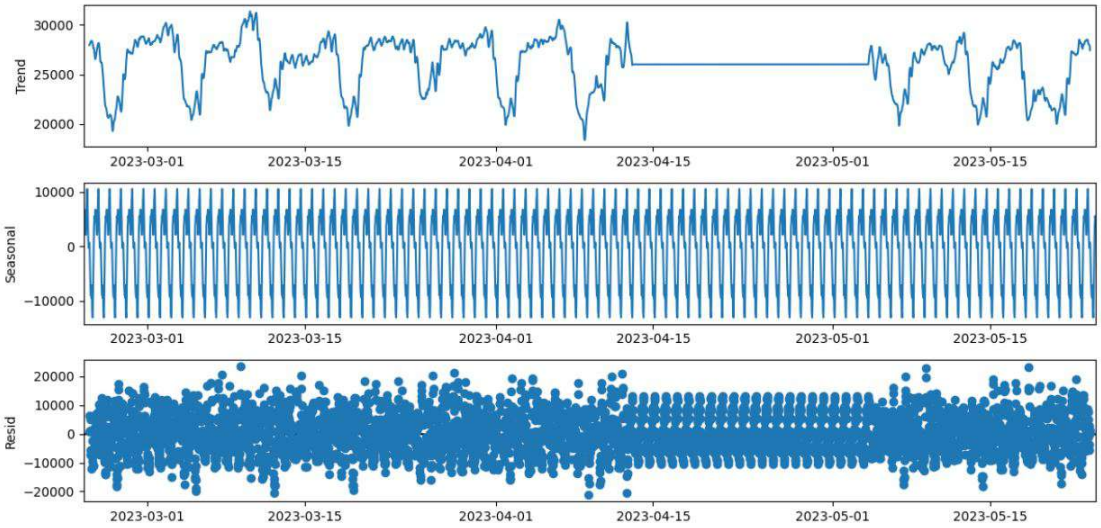

Figura 5.12: Trend della disponibilità di parcheggi in 's-Hertogenbosch.

Tutte sessioni di addestramento con ARIMA, SARIMA e XGBoost sono state eseguite con lo stesso approccio di fine-tuning descritto nella sezione riguardante il traffico pedonale.

5.4 Risultati predizione traffico pedonale

Anche se sono state testate, seguendo i risultati dell'analisi preliminare, diverse configurazioni, i parametri migliori per il modello ARIMA sono stati un termine di autoregressione pari a 4 e un valore di media mobile pari ad 1, producendo un Root Mean Squared Error (RMSE) complessivo di 199, 14.
L'Errore Assoluto Medio (MAE) delle previsioni è stato di 70, 67 pedoni.
Il Root Mean Squared Error complessivo commesso dal modello XGBoost è stato invece di 251, 87.
Per visualizzare la capacità del modello ARIMA di adattarsi alla serie temporale, può essere utile la Figura 5.13, nella quale troviamo sull'asse verticale il numero di pedoni e sull'asse orizzontale la totalità delle predizioni, per ogni connessione.

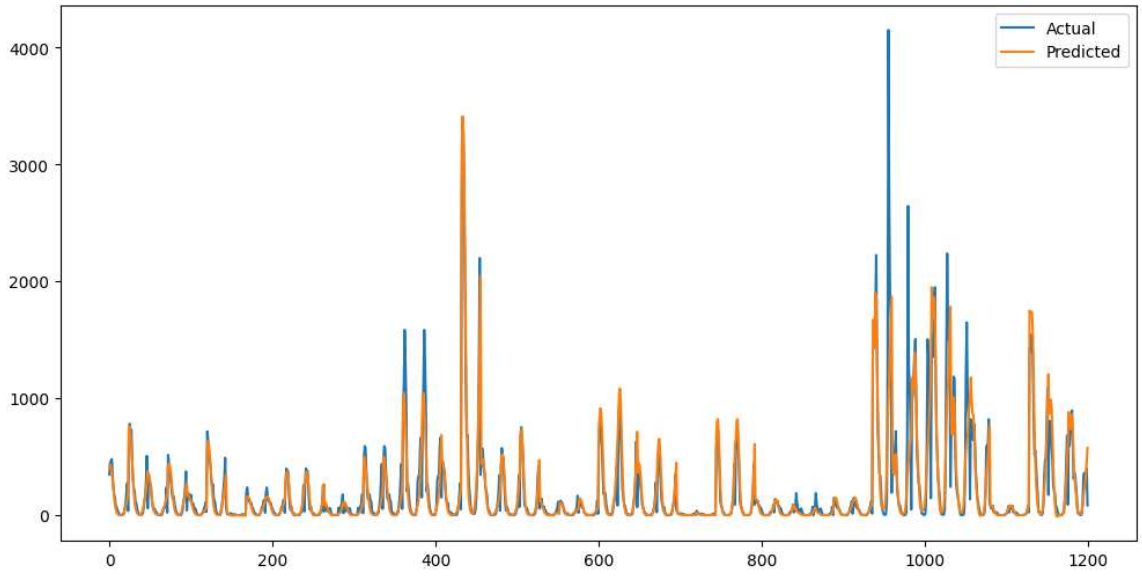

Figura 5.13: Risultati del modello ARIMA per la previsione del traffico pedonale di Den Bosch.

Come si può chiaramente notare, il modello ARIMA offre in generale un errore quadratico medio abbastanza basso e il modello sembra adattarsi abbastanza bene ai dati rispetto ad XGB, come si nota nelle Tabelle 5.1, 5.2 e 5.3, tuttavia ci sono alcuni spunti interessanti. Si può notare che in alcuni casi speciali il regressore XGB si comporta meglio.

Questi casi, evidenziati nelle tabelle, sono quelli in cui vengono coinvolti luoghi della città con un traffico pedonale molto elevato (Figura 5.14) come nei pressi della stazione centrale, della Stationweeg o della Sint-Janskathedraal.

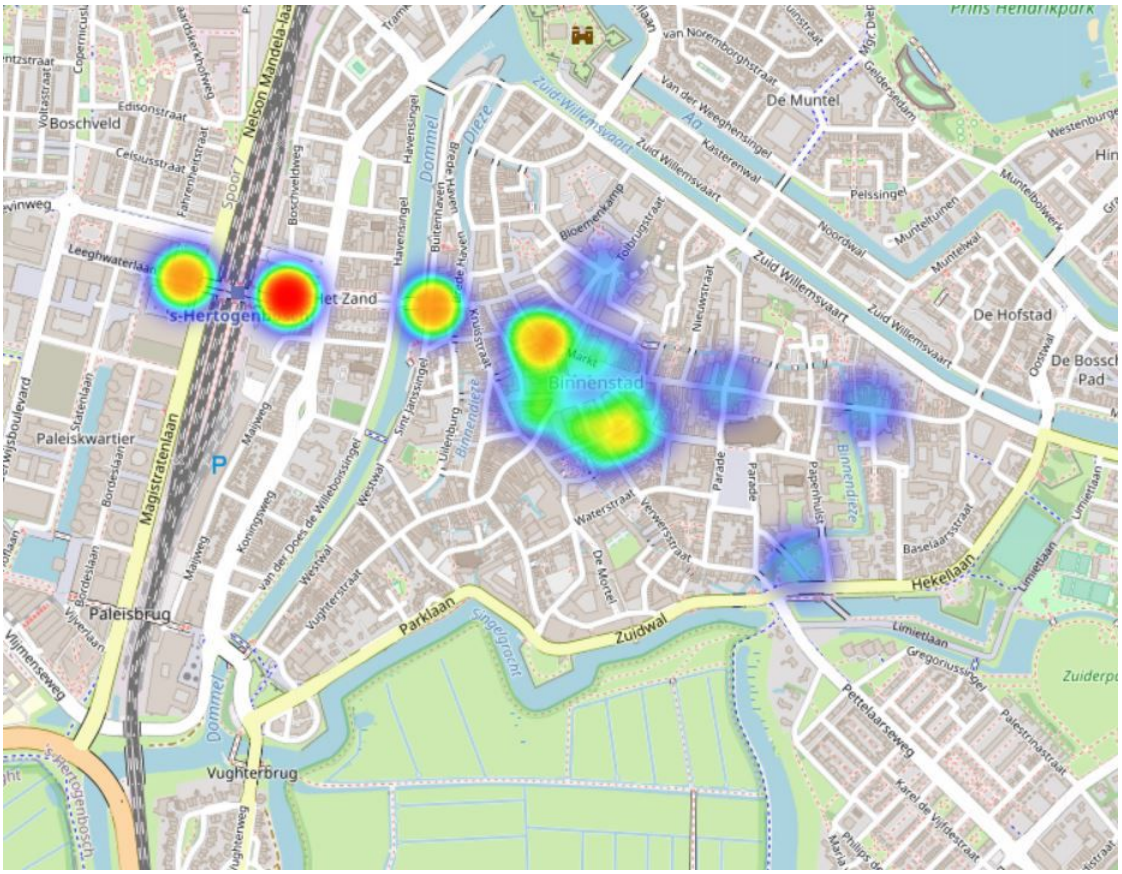

Figura 5.14: Aree di Den Bosch maggiormente affollate da pedoni.

Questo si può dedurre anche dal grafico 5.13, dove ci sono dei picchi di dati che non vengono colti a sufficienza dalle previsioni di ARIMA.

Difatti XGBoost tende ad essere più robusto, rispetto ad ARIMA, quando si tratta di gestire outliers o picchi molto alti, in quanto gli alberi decisionali che costituiscono il regressore possono adattarsi meglio a tali punti anomali.

Nelle Tabelle 5.1, 5.2 e 5.3 sono riportate le prestazioni di entrambi i modelli, utilizzando una granularità temporale, per l'addestramento e la previsione, di un'ora.

Tabella 5.1: Prestazioni di ARIMA e XGBoost sulla predizione di traffico pedonale nella città di 's-Hertogenbosch pt. 1.

node	connection	xgb_rmse	arima_rmse
stadhuis	__total	34.825168	30.615323
	markt	34.825168	30.615323
station	**__total**	454.914495	**812.44273**
	leeghwaterlaan	358.718169	523.825769
	stationweeg	254.026286	329.375033
stationweeg	**__total**	419.131606	**459.80807**
	station	190.628889	328.851227
	visstraat	243.158903	194.731397
verwersstraat	__total	32.70777	15.388099
	markt	32.70777	15.388099
visstraat	__total	394.624718	69.576286
	markt	191.288643	168.780583
	stationweeg	227.802989	131.377491

Tabella 5.2: Prestazioni di ARIMA e XGBoost sulla predizione di traffico pedonale nella città di 's-Hertogenbosch pt. 2.

node	connection	xgb_rmse	arima_rmse
HH_0	HH_1	123.491475	58.404649
	__total	209.77767	91.221112
	markt	91.421778	40.227868
HH_1	HH_0	154.468642	70.653731
	HH_2	36.997793	28.19069
	__total	185.881932	90.276371
	monseigneur	14.87713	8.131619
HH_2	HH_1	37.833186	34.09705
	__total	37.833186	34.09705
arena	__total	71.2849	58.133974
	marktstraat	71.2849	58.133974
ext	__total	28.191229	38.488388
	st_jan	28.191229	38.488388
kerkstraat	__total	125.761799	95.874767
	markt	125.761799	95.874767
leeghwaterlaan	__total	203.815648	239.280922
	station	203.815648	239.280922

Tabella 5.3: Prestazioni di ARIMA e XGBoost sulla predizione di traffico pedonale nella città di 's-Hertogenbosch pt. 3.

node	connection	xgb_rmse	arima_rmse
markt	HH_0	209.795631	48.154851
	__total	1274.741508	406.595365
	kerkstraat	236.870216	62.138256
	marktstraat	166.035126	68.393604
	schapenmarkt	156.794803	94.159337
	snellestraat	80.583609	27.775057
	stadhuis	81.733413	27.322408
	verwersstraat	103.715463	27.519914
	visstraat	283.339528	145.105406
marktstraat	__total	209.701001	174.461596
	arena	96.381161	64.748645
	markt	115.490256	112.812473
monseigneur	HH_1	6.617243	5.84809
	__total	6.617243	5.84809
schapenmarkt	__total	148.102407	121.022536
	markt	148.102407	121.022536
snellestraat	__total	32.026013	22.742506
	markt	32.026013	22.742506
st_jan	**__total**	20.30842	**40.171803**
	ext	20.30842	40.171803

Al fine di comprendere se le prestazioni di ARIMA sono significativamente migliori rispetto a quelle di XGB, è stato effettuato un test di significatività statistica.

Il test di Wilcoxon [36] è un test statistico che confronta due campioni correlati o accoppiati. È particolarmente utile quando i dati non seguono una distribuzione normale o quando la dimensione del campione è piccola. Per effettuare questo test, vengono calcolate le differenze tra le osservazioni accoppiate e vengono determinati i signed ranks di queste differenze. Il test verifica quindi se la mediana dei ranks è significativamente diversa da zero.

Nel confrontare le prestazioni di due modelli diversi sullo stesso set di dati, il test di Wilcoxon può essere utilizzato per determinare quale modello si comporta meglio. Esaminando le differenze tra i valori predetti di ciascun modello, è possibile scoprire se le prestazioni di un modello sono significativamente migliori dell'altro. Questa informazione può essere preziosa nel prendere decisioni su quale modello utilizzare per le esigenze specifiche.

L'ipotesi nulla per il test di Wilcoxon è che non vi sia differenza tra i due campioni correlati, quindi la mediana della differenza tra i due campioni è zero. L'ipotesi alternativa è che vi sia una differenza tra i due campioni correlati. Per verificare il valore p del test di Wilcoxon, calcoliamo innanzitutto la statistica del test, ovvero la somma dei ranks delle differenze assolute tra i campioni.
Il $p - value$ è la probabilità di osservare una statistica del test altrettanto estrema o più estrema della statistica del test osservata, considerando l'ipotesi nulla come vera. Se il $p - value$ è inferiore al livello di significatività, solitamente $0,05$, rifiutiamo l'ipotesi nulla e concludiamo che vi sono prove di una differenza tra i due campioni correlati, altrimenti non rifiutiamo l'ipotesi nulla e concludiamo che non vi sono prove di una differenza tra i due campioni correlati.

In particolare, il $p - value$ confrontando le prestazioni dei due modelli è stato $p - value = 0,002$, pertanto possiamo concludere che in generale il modello ARIMA ha prestazioni migliori rispetto a XGB.

5.5 Risultati disponibilità di parcheggi auto

Tre sono stati i modelli testati durante questa fase: ARIMA, SARIMA, e XGBoost. Contrariamente a quanto successo per il traffico pedonale, nel caso della previsione della disponibilità di parcheggi il modello che offre risultati più accurati risulta essere XGB, con un root mean squared error generale pari a 1445.567, rispetto l'errore di ARIMA e SARIMA di circa 2737.364, come evidente in Tabella 5.4.

Tabella 5.4: Confronto dei valori RMSE tra ARIMA, SARIMA e XGB per i diversi parcheggi di 's-Hertogenbpsch.

Parcheggio	arima_rmse	sarima_rmse	xgb_rmse
07f8962e-d251-4055-b9a8-0752024b355e	1557.331	1663.212	986.584
4498b82f-6da9-4887-b2a1-be921341d1d0	3818.634	3951.852	2480.906
6ea457a0-c73c-4b1a-8602-ff6af1114fe3	3810.820	3705.691	1171.666
b17c99c0-65af-4451-a700-a21b24fc611e	530.945	498.569	188.122
c22bc535-e3cb-41e6-a4aa-0870f94ee478	2006.675	2040.334	1231.897
c724ecc9-e6ae-42aa-b462-8ce7cf48912a	2183.750	2164.754	1697.478
d5565e93-2f9f-4074-8548-e631b72bae66	3367.325	3427.479	1300.850

Per visualizzare graficamente le differenze prestazionali e le diverse modalità con cui i modelli testati si adattano alla serie temporale le Figure 5.15, 5.16 e 5.17 possono essere utili, in rosso sono rappresentati i valori predetti e in blu quelli reali.
Per interpretare al meglio le figure va sottolineato come siano state generate 24 previsioni ad intervalli di trenta minuti per ognuno dei sette garage in esame, per un totale di 168 previsioni concatenate sull'asse orizzontale dei grafici.
Generalmente XGB è in grado di catturare relazioni non lineari nei dati, modelli di tendenza più complessi e gestire meglio picchi di osservazioni, producendo così previsioni più precise. Inoltre, includere al suo interno features temporali, come i primi lag ritardati permette di incrementare notevolmente le prestazioni del modello, dando cosi la possibilità di catturare tendenze.

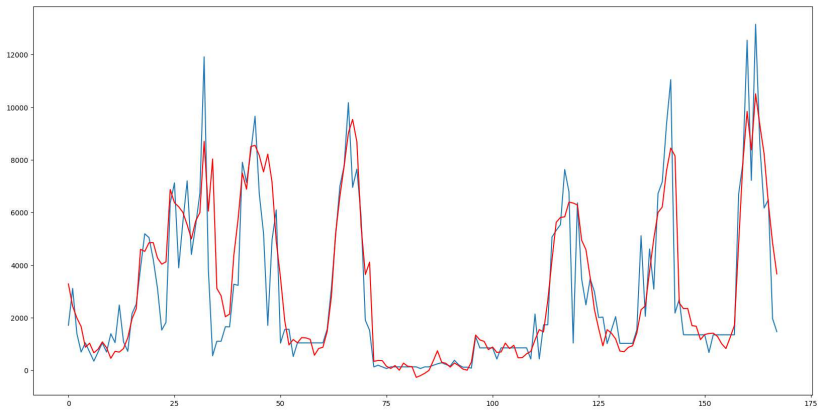

Figura 5.15: Risultati del modello XGB per la previsione della disponibilità di parcheggi di Den Bosch.

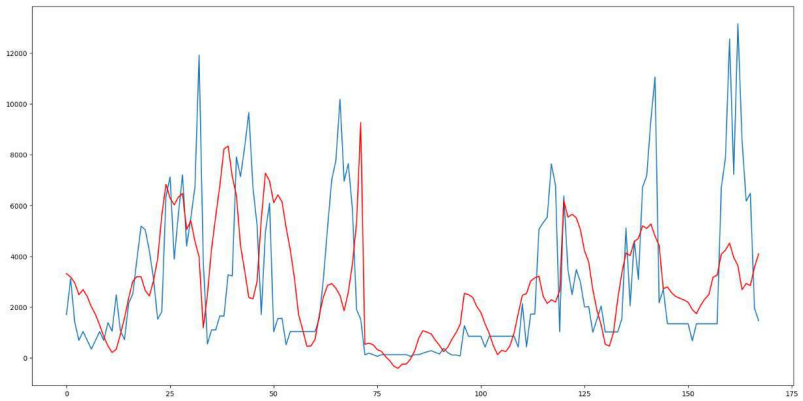

Figura 5.16: Risultati del modello ARIMA per la previsione della disponibilità di parcheggi di Den Bosch.

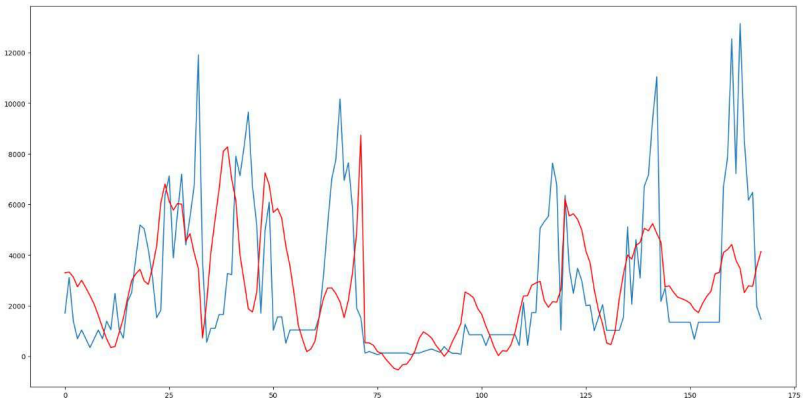

Figura 5.17: Risultati del modello SARIMA per la previsione della disponibilità di parcheggi di Den Bosch.

CAPITOLO 6

Breda

In questo capitolo verranno mostrati gli approcci utilizzati e analizzati i risultati ottenuti dalle sessioni sperimentali eseguite sul caso di studio riguardante Breda. Verranno indicati i modelli testati, le strategie adottate e le prestazioni ottenute in termini di accuratezza delle previsioni.

6.1 Contesto urbano

Breda 6.1, collocata nella provincia del Brabante settentrionale è una piccola città caratterizzata dalla presenza dei tipici canali che si distribuiscono attraverso i vicoli suggestivi del posto. Sono svariati i festival che si tengono in città, tra i quali uno dei più noti è il Breda Jazz Festival, così come le molteplici attività commerciali, che attirano turisti e visitatori da tutta Europa

Anche in questo caso la stazione ferroviaria è connessa in maniera estremamente efficace al resto della nazione e la città è facilmente raggiungibile in auto attraverso numerose autostrade.

Le strade maggiormente trafficate sono quelle che portano dalla stazione centrale al centro storico e i parcheggi pubblici sono distribuiti lungo il perimetro del centro.

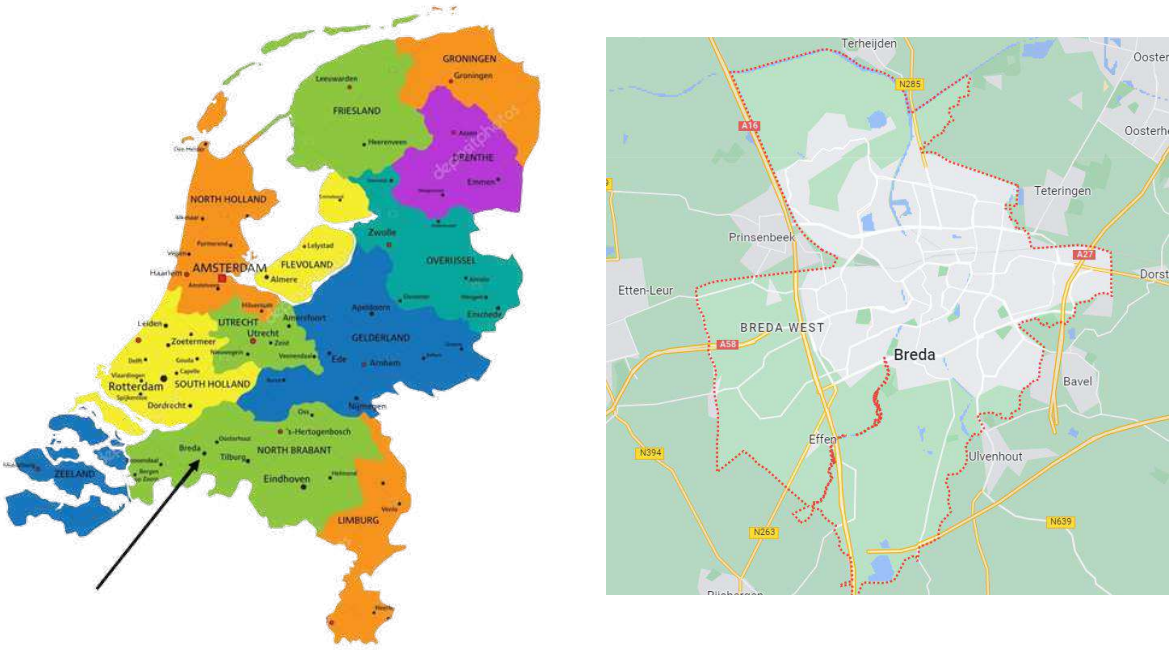

Figura 6.1: Posizione di Breda rispetto i Paesi Bassi e mappa della città.

6.2 Disponibilità di parcheggi auto

Per la predizione della disponibilità di parcheggi a Breda, sono stati utilizzati i dati degli ultimi 90 giorni a partire da quello corrente. In Figura 6.2 è mostrata una panoramica dell'andamento delle serie temporali di ciascun parcheggio della città. Anche in questo caso, come per Den Bosch, dai grafici è possibile notare innanzitutto la presenza di un periodo di osservazioni mancanti, che sono state poi imputate con la media e quindi caratterizzate dalla presenza di un segmento retto orizzontale.

In maniera analoga a quanto fatto per i parcheggi di 's-Hertogenbosch, è stata analizzata la correlazione della serie con i suoi valori ritardati. Dalle analisi è risultata una scarsa correlazione, se non con il primo valore ritardato e quello del giorno antecedente, suggerendo che un modello ARIMA potesse non essere la scelta migliore per descrivere il problema in esame.

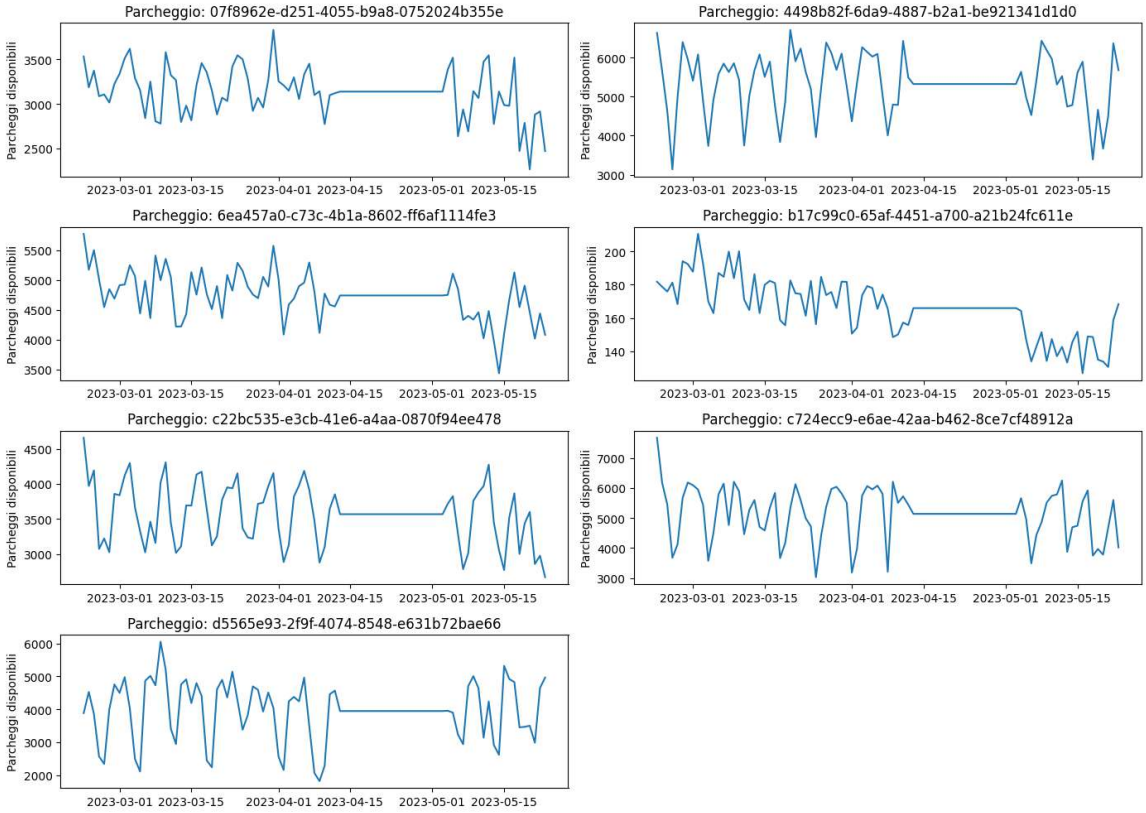

Figura 6.2: Serie temporale dei parcheggi pubblici a Breda.

Anche in questo caso quindi è stato utilizzato un regressore di tipo Extreme Gradient Boosting, i cui parametri migliori sono stati identificati con una grid search combinata a una cross-validation di tipo expanding window splitter, andando ad aggiungere alle features anche i valori passati della serie stessa.

6.3 Disponibilità di parcheggi per biciclette

Per la predizione della disponibilità di parcheggi per biciclette a Breda, sono stati utilizzati i dati degli ultimi 90 giorni a partire da quello corrente.
In Figura 6.3 è mostrata una panoramica dell'andamento delle serie temporali di ciascun parcheggio della città. Così come per i parcheggi auto, dalle analisi preliminari non è risultata evidenza significativa di correlazioni temporali, se non con il primo ritardo temporale, pertanto ancora una volta è stato utilizzato un modello XGB.

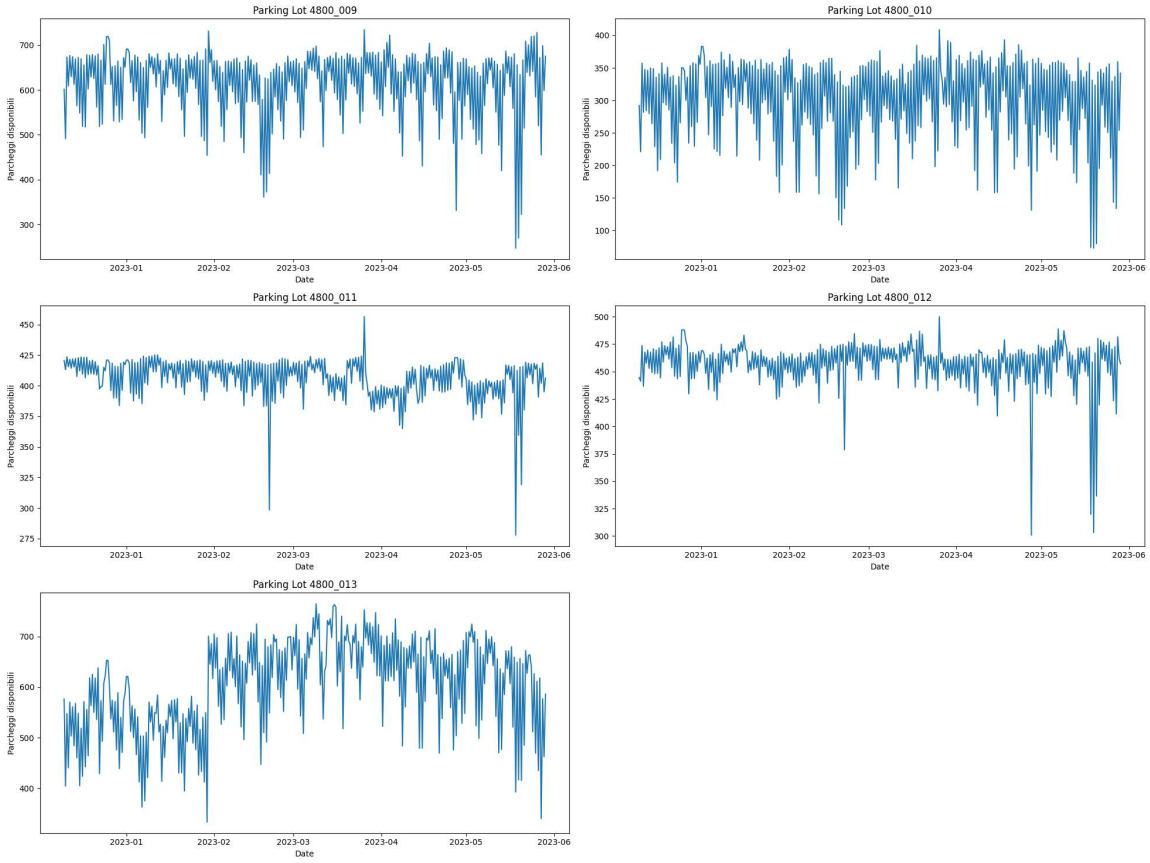

Figura 6.3: Serie temporale dei parcheggi pubblici a Breda.

6.4 Risultati disponibilità di parcheggi auto

Così come per 's-Hertogenbosch, il modello migliore è stato XGBoost, offrendo un Root Mean Sqaured Srror complessivo pari a 1199.566, e un errore per ogni parcheggio pari ai valori mostrati in Tabella 6.1.

Confrontando i risultati ottenuti attraverso l'uso di un modello ARIMA è stato evidente come quest'ultimo producesse errori molto più alti, non riuscendo a modellare a sufficienza gli andamenti irregolari della serie temporale.

Questa tendenza è stata confermata dall'esecuzione di un test di significatività statistica a controprova che effettivamente XGBoost sia un modello più adatto a descrivere il problema.

Una panoramica generale delle performance del modello è mostrata in Figura 6.4.

Dal grafico si vede come le predizioni, indicate in rosso, siano piuttosto fedeli ai valori di test, ad eccezione di alcuni casi in cui sono presenti picchi eccessivamente alti, che non vengono modellati perfettamente dal regressore.

Anche in questo caso, l'inclusione di lag temporali tra le features di input del modello XGB, permette di catturare tendenze e offrire una maggiore accuratezza generale.

Tabella 6.1: Valori di RMSE per un modello XGBoost per i diversi parcheggi di Breda.

Parcheggio	xgb_rmse
07f8962e-d251-4055-b9a8-0752024b355e	957.440
4498b82f-6da9-4887-b2a1-be921341d1d0	1261.077
6ea457a0-c73c-4b1a-8602-ff6af1114fe3	1032.860
b17c99c0-65af-4451-a700-a21b24fc611e	286.742
c22bc535-e3cb-41e6-a4aa-0870f94ee478	824.626
c724ecc9-e6ae-42aa-b462-8ce7cf48912a	1319.607
d5565e93-2f9f-4074-8548-e631b72bae66	1998.825

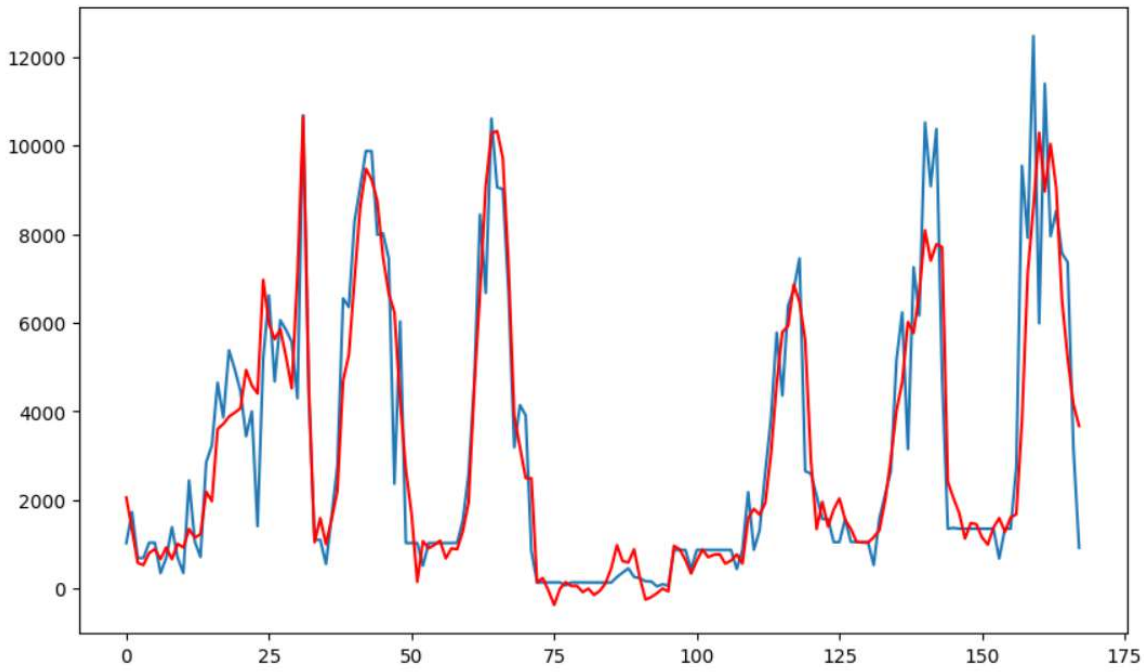

Figura 6.4: Risultati del modello XGBoost per la previsione della disponibilità di parcheggi di Breda.

6.5 Risultati disponibilità di parcheggi per biciclette

Una panoramica generale delle performance del modello è mostrata in Figura 6.5, nella quale sono indicate in rosso le predizioni e in blu i valori reali.

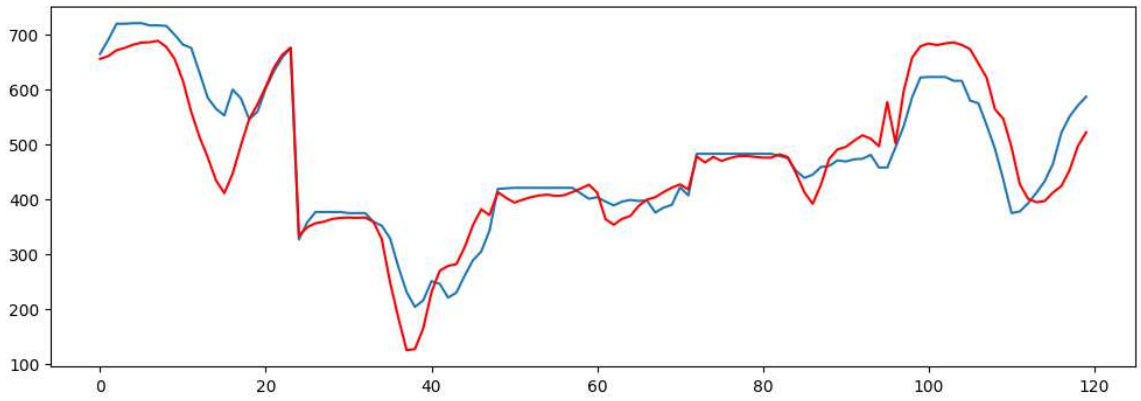

Figura 6.5: Risultati del modello XGBoost per la previsione della disponibilità di parcheggi per biciclette di Breda.

Il RMSE complessivo delle previsioni è stato pari a 53.23, con un errore assoluto medio di 39.40 unità. In tabella 6.2 è mostrato l'errore per ogni parcheggio.

Tabella 6.2: Valori di RMSE per un modello XGBoost per i diversi parcheggi per biciclette di Breda.

Parcheggio	xgb_rmse
4800_009	72.10
4800_010	48.81
4800_011	20.73
4800_012	32.99
4800_013	71.18

CAPITOLO 7

Analisi dei risultati e minacce alla validità

Al termine del lavoro è stato possibile avere una visione più chiara delle problematiche e delle sfide sorte inizialmente.

RQ1: Quali sono le features più adatte a descrivere un problema di predizione di serie temporale in un'ottica di smart city?

Alla luce di tutte le sessioni condotte durante questo lavoro di tesi, e attraverso una valutazione dell'importanza delle varie features utilizzate nei diversi modelli proposti, è stato abbastanza chiaro come le variabili fondamentali per descrivere i problemi siano quelle legate ai valori pregressi delle serie stesse, ovvero i lag temporali. Per avere conferma di ciò, alla luce del divario prestazionale tra i modelli ARIMA e XGB, evidenziato soprattutto dal fenomeno del traffico pedonale, è stato testato l'utilizzo anche di un modello XGB includendo alcune features e statistiche temporali, estratte dall'analisi della serie.
Il risultato di questa indagine ha mostrato come l'inserimento di alcune variabili, ad esempio il valore della serie alla stessa ora del giorno antecedente alla rilevazione o la media di una finestra temporale settimanale, permetta di aumentare notevolmente le prestazioni di XGB, offrendo in generale un errore quadratico medio molto vicino a quello di un modello ARIMA.

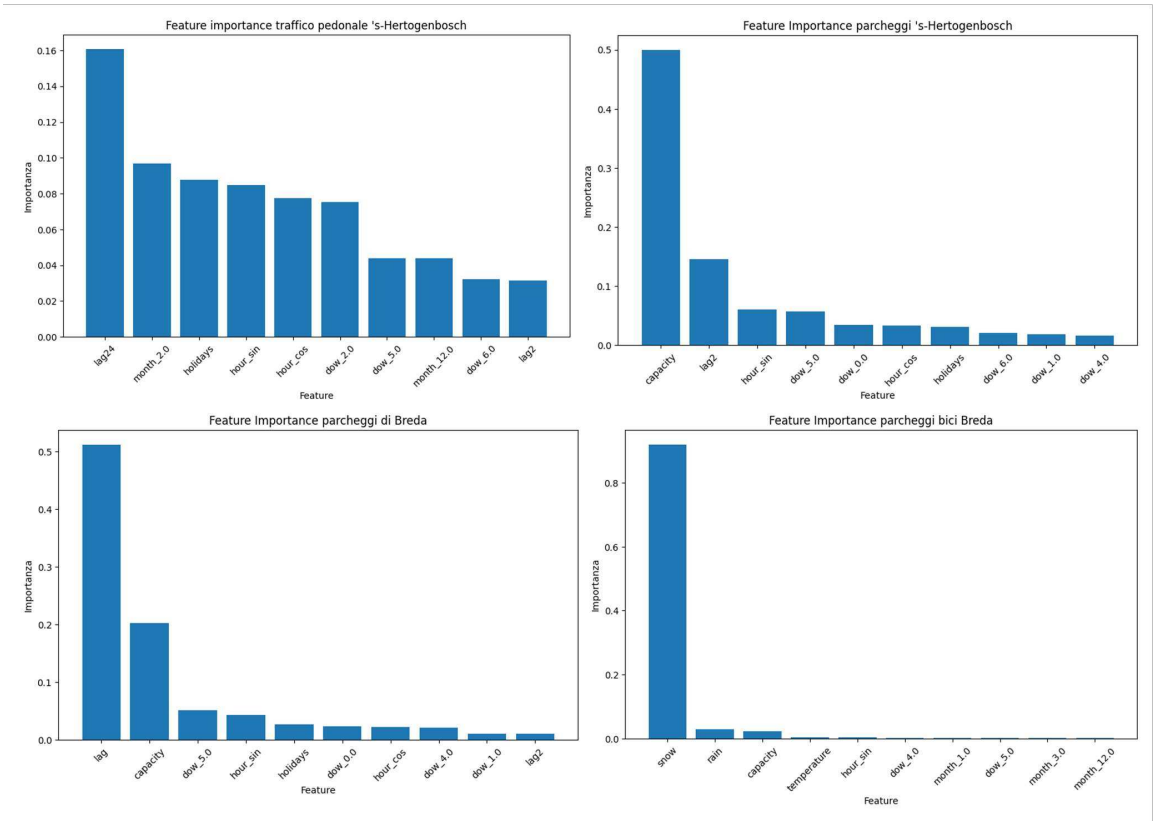

Figura 7.1: Importanza delle feature nel modellare i problemi.

Analizzando l'importanza che le varie features hanno nel modellare i problemi, Figura 7.1, si può notare come l'uso di dati meteorologici, si è rivelata una strategia altrettanto funzionale per modellare i vari problemi e ottenere prestazioni migliori, soprattutto nel caso dei parcheggi di biciclette per Breda.

Le informazioni riguardanti i giorni settimanali e gli orari delle osservazioni risultano anche essere utili. È lecito supporre che tutti i fenomeni analizzati, siano frutto di abitudini umane, che si ripetono in maniera ciclica nel tempo. Salvo straordinarie eccezioni, legate principalmente a fenomeni turistici o giorni di festa, i suddetti pattern tendono a ripetersi, e queste stesse ripetizioni sono la chiave per modellare il fenomeno.

Variabili, invece, in alcuni casi meno utili sono quelle legate al mese della data di osservazione. Infatti considerando che per l'addestramento di tutti i modelli sono stati utilizzati dati degli ultimi 200 o 90 giorni, non è stato possibile catturare relazioni mensili o annuali nei dati.

RQ2: Quali sono le strategie e i modelli di Machine Learning più adatti per affrontare e modellare un problema di smart city forecasting?

La risposta a questa domanda è che difatti non è esistono delle singole soluzioni e pratiche universalmente migliori di altre. Gestire un progetto di Machine Learning richiede di valutare e testare molteplici strategie al fine di identificare quella migliore. Tuttavia ci sono delle pratiche che sicuramente hanno facilitano la risoluzione del problema. Un primo approccio utile è stato quello di comprendere e analizzare i dati prima di manipolarli. La valutazione di eventuali autocorrelazioni, l'identificazione di stagionalità, l'esistenza di valori anomali o nulli e la necessità di ricampionare le osservazioni sono tutte operazioni fondamentali per la riuscita della sperimentazione, o almeno per la semplificazione di essa.
Una buona fase di pre-processing e feature engineering sono altrettanto importanti. Ricercare ed estrarre variabili, trasformarle in rappresentazioni efficienti, normalizzarle e/o standardizzarle sono tutte fasi essenziali per la buona riuscita del piano e per la costruzioni di pipeline scalabili e generalizzabili.
Anche valutare l'ampio ventaglio di modelli esistenti può essere impegnativo.
Fortunatamente la letteratura offre numerose soluzioni testate e approvate in merito alle serie temporali, che permettono di restringere il campo e ridurre le possibili strade da percorrere. Specificatamente, i risultati di questo studio hanno permesso di affermare come l'uso di un modello statistico possa rivelarsi, in alcuni casi sufficientemente efficiente. Nella fattispecie di questo lavoro, il modello che generalmente offre risultati più stabili, affidabili e accurati, sui diversi fenomeni trattati, è un regressore di tipo Extreme Gradient Boosting con approccio gerarchico.

Fin da sempre è stato noto come la creazione di un progetto software, e a maggior ragione un progetto con una componente di ML, richieda la valutazione e l'accettazione di una serie di compromessi. Bisogna valutare il bilancio tra accuratezza e complessità dei modelli. Modelli complessi come le reti neurali potrebbero essere estremamente efficaci per modellare un problema di predizione di serie temporali, ma potrebbero richiedere tempi di addestramento e risorse computazionali nettamente superiori a quelle disponibili.

Del resto modelli più semplici, come dimostrato da questo lavoro, potrebbero comunque offrire risultati accettabili utilizzando un minor numero di risorse.
Un altro aspetto concerne l'interpretabilità dei vari modelli testati, che permettono di garantire maggiore comprensione delle predizioni e offrire una visione trasparente del processo decisionale.
Un altro compromesso che bisogna affrontare è quello tra la dimensione del dataset utilizzato e la frequenza di aggiornamento e addestramento. Se da un lato aumentare la quantità di dati storici necessari ad addestrare il modello può sicuramente incrementare la capacità di quest'ultimo di catturare relazioni tra i dati, e di conseguenza offrire risultati migliori, utilizzare dati con una frequenza intensa, potrebbe, ancora una volta, richiedere costi computazionali non disponibili.

Le risposte alle domande di ricerca hanno rivestito un ruolo fondamentale nel processo di design delle API finalizzate all'implementazione di modelli di machine learning in MLOps per la pianificazione urbana.
La ricerca dettagliata sulle features più pertinenti e sui modelli di machine learning più efficaci ha diretto la definizione di parametri chiave per la creazione delle API. Ad esempio, l'identificazione delle features più informative ha guidato la selezione delle variabili da includere nei dati di input delle API, assicurando che siano rappresentative delle dinamiche temporali cruciali per la previsione nella pianificazione urbana. Inoltre, il processo iterativo di sperimentazione e confronto tra vari modelli e tecniche di ri-campionamento ha influenzato direttamente la scelta della migliore pipeline da mettere in produzione.
È stato possibile definire il rapporto ottimale tra tempi di addestramento (e ri-addestramento) dei modelli e l'allocazione efficiente delle risorse di memoria, aspetti critici nell'implementazione pratica dei modelli in ambienti operativi. Ad esempio il modello ARIMAX ha mostrato ottime performance in molte connessioni stradali di Den Bosch, al costo però di tempi di addestramento molto lunghi ed elevati rischi di fallimento causa overflow di memoria, rispetto il modello XGB.
Questa ottimizzazione è stata essenziale per garantire che le API potessero essere in grado di adattarsi dinamicamente a cambiamenti nelle condizioni del contesto urbano e di mantenere le stesse prestazioni nel lungo termine .

Infine è importate evidenziare alcune possibili limitazioni che potrebbero influenzare i risultati:

- **Assunzioni statistiche:** L'uso di modelli di Machine Learning basati su analisi statistiche delle serie temporali potrebbe fornire approcci avanzati per la previsione dei dati temporali. Tuttavia, la validità e l'affidabilità delle predizioni dipendono strettamente dal rispetto delle fondamentali assunzioni statistiche. Elementi critici come la stazionarietà, l'indipendenza temporale, la normalità delle distribuzioni e la variabilità delle osservazioni, svolgono un ruolo chiave nell'assicurare la solidità delle conclusioni tratte da tali modelli. La violazione di queste assunzioni potrebbe compromettere la precisione delle stime e la validità delle previsioni future.

- **Futuro incerto:** La storia ci insegna che le abitudini umane sono soggette a continue evoluzioni legate a fattori controllati e non. Affinché netti cambiamenti delle abitudini possano essere percepiti da un modello di Machine Learning e riflesse nelle previsioni, potrebbe essere necessario valutare nuove e più complesse risorse.

CAPITOLO 8

Conclusioni e sviluppi futuri

In conclusione, questa tesi rappresenta un contributo significativo nell'ambito della gestione urbana data-driven nei Paesi Bassi, caratterizzati da una notevole crescita demografica e urbanistica. Attraverso il progetto Smart City Monitor, sono stati affrontati i crescenti problemi legati alla gestione delle città, con un focus particolare sulle amministrazioni di Breda e 's-Hertogenbosch nella regione del Brabante Settentrionale. Il lavoro di ricerca si è concentrato sull'analisi dei dati e sulla progettazione di pipeline di addestramento per modelli di Machine Learning applicati a serie temporali. L'obiettivo è stato fornire API utilizzabili da terzi, in grado di generare predizioni attendibili e in tempo reale su fenomeni chiave di una Smart City, come il traffico pedonale e la disponibilità di parcheggi auto e per biciclette.
I risultati ottenuti sono stati molto incoraggianti, dimostrando l'efficacia delle strategie di Machine Learning implementate. Le pipeline di addestramento, integrate in un contesto operativo, hanno fornito predizioni sufficientemente accurate, aprendo prospettive interessanti per l'ottimizzazione della vita cittadina. Il modello Extreme Gradient Boosting combinato ad un approccio di riconciliazione gerarchica si è dimostrato particolarmente efficiente nel predire i fenomeni in esame, in particolar modo per la previsione di traffico pedonale, così come l'uso combinato di variabili esogene e temporali.

I risultati ottenuti attraverso questo lavoro non sono sicuramente definitivi e sono diversi i possibili spunti di sviluppi futuri.
Questa ricerca potrebbe estendersi all'inclusione di altri fenomeni rilevanti per una gestione urbana completa e avanzata. Oltre alle predizioni dei fenomeni precedentemente descritti, si potrebbero considerare nuovi ambiti quali la previsione del traffico automobilistico nelle zone urbane, il monitoraggio dei motoveicoli e l'analisi del traffico ciclistico.
Si potrebbero sperimentare modelli più complessi, come le reti neurali ricorrenti, particolarmente adatte per fenomeni sequenziali e temporali.
Infine si potrebbero ottenere spunti interessanti analizzando la correlazione tra diversi fenomeni, e studiare come questi finiscono per influenzarsi a vicenda. Ad esempio valutare come un elevata affluenza pedonale potrebbe essere associata ad un certo tasso di occupazione dei parcheggi pubblici.

Bibliografia

[1] Macrotrends, "Netherlands urban population 1960-2023," https://www.macrotrends.net/countries/NLD/netherlands/urban-population, n.d., accesso al sito: 12 giugno 2023. (Citato a pagina 1)

[2] S. Musa, "Smart city roadmap," https://www.academia.edu/21181336/Smart_City_Roadmap, gennaio 2016. (Citato a pagina 2)

[3] Government of the Netherlands, "Mobility, public transport and road safety," https://www.government.nl/topics/mobility-public-transport-and-road-safety, accesso al sito: 12 giugno 2023. (Citato a pagina 2)

[4] "Smart city monitor - den bosch," https://www.denbosch.nl/nl/datastad/smart-city-monitor. (Citato alle pagine 3 e 23)

[5] "Smart city monitor - opzuid," https://www.stimulus.nl/opzuid/smart-city-monitor-houdt-datastad-s-hertogenbosch-gezond-en-vitaal. (Citato alle pagine 3 e 23)

[6] Wikipedia contributors, "Data science — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Data_science&oldid=1151954922, 2023, [Online; accessed 28-April-2023]. (Citato a pagina 6)

[7] L. Sigwadi, "Data science and the fourth industrial revolution (4ir)," *University of the Western Cape*, 2020. (Citato a pagina 7)

[8] R. Ridi, "La piramide dell'informazione: una introduzione," *AIB studi*, vol. 59, no. 1-2, 2019. (Citato a pagina 7)

[9] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. Hernández-Orallo, M. Kull, N. Lachiche, M. J. Ramirez-Quintana, and P. Flach, "Crisp-dm twenty years later: From data mining processes to data science trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3048–3061, 2019. (Citato a pagina 8)

[10] A. Vogelsang and M. Borg, "Requirements engineering for machine learning: Perspectives from data scientists," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2019, pp. 245–251. (Citato a pagina 9)

[11] A. Burkov, *Machine learning engineering*. True Positive Incorporated Montreal, QC, Canada, 2020, vol. 1. (Citato a pagina 10)

[12] K. Salama, J. Kazmierczak, and D. Schut, "Practitioners guide to mlops: A framework for continuous delivery and automation of machine learning," *Google Could White paper*, 2021. (Citato a pagina 11)

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794. (Citato a pagina 17)

[14] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022. (Citato a pagina 17)

[15] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018. (Citato a pagina 21)

[16] K. R. Kunzmann, "Smart cities: A new paradigm of urban development," *Crios*, vol. 4, no. 1, pp. 9–20, 2014. (Citato a pagina 21)

[17] M. Krzyzanowski, B. Kuna-Dibbert, and J. Schneider, *Health effects of transport-related air pollution*. WHO Regional Office Europe, 2005. (Citato a pagina 21)

[18] G. Cascavilla, D. A. Tamburri, F. Leotta, M. Mecella, and W. Van Den Heuvel, "Counter-terrorism in cyber-physical spaces: Best practices and technologies from the state of the art," *Information and Software Technology*, p. 107260, 2023. (Citato a pagina 22)

[19] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th design automation conference*, 2010, pp. 731–736. (Citato a pagina 22)

[20] J. Ullrich and E. Weippl, "Cyphysec: Defending cyber-physical systems," *ERCIM NEWS*, no. 102, pp. 18–18, 2015. (Citato a pagina 22)

[21] C. Berger, A. Hees, S. Braunreuther, and G. Reinhart, "Characterization of cyber-physical sensor systems," *Procedia Cirp*, vol. 41, pp. 638–643, 2016. (Citato a pagina 22)

[22] "Smart city monitor - jads," https://www.jads.nl/news. (Citato a pagina 24)

[23] J. Mihelj, A. Kos, and U. Sedlar, "Source reputation assessment in an iot-based vehicular traffic monitoring system," *Procedia computer science*, vol. 147, pp. 295–299, 2019. (Citato a pagina 26)

[24] D. Heo, J. Chung, B. Kim, H. Yong, G. Shin, J.-W. Cho, D. Kim, and S. Lee, "Triboelectric speed bump as a self-powered automobile warning and velocity sensor," *Nano Energy*, vol. 72, p. 104719, 2020. (Citato a pagina 26)

[25] D. Mavrokapnidis, N. Mohammadi, and J. Taylor, "Community dynamics in smart city digital twins: A computer vision-based approach for monitoring and forecasting collective urban hazard exposure," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2021. (Citato a pagina 26)

[26] S. C. v. d.-A. Spek, "Optimizing routing and safety for pedestrians." (Citato a pagina 27)

[27] A. Cohen and S. Dalyot, "Machine-learning prediction models for pedestrian traffic flow levels: Towards optimizing walking routes for blind pedestrians," *Transactions in GIS*, vol. 24, no. 5, pp. 1264–1279, 2020. (Citato a pagina 27)

[28] J. Bakermans, "Pedestrian route planning in a hybrid data environment." (Citato a pagina 28)

[29] H. O. Jacobs, O. K. Hughes, M. Johnson-Roberson, and R. Vasudevan, "Real-time certified probabilistic pedestrian forecasting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2064–2071, 2017. (Citato a pagina 28)

[30] X. Wang, J. Liono, W. Mcintosh, and F. D. Salim, "Predicting the city foot traffic with pedestrian sensor data," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2017, pp. 1–10. (Citato a pagina 28)

[31] F. Van den Bossche, G. Wets, and T. Brijs, "A regression model with arima errors to investigate the frequency and severity of road traffic accidents," LUC, Tech. Rep., 2004. (Citato a pagina 29)

[32] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, V. A. Kamaev *et al.*, "A survey of forecast error measures," *World applied sciences journal*, vol. 24, no. 24, pp. 171–176, 2013. (Citato a pagina 32)

[33] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe *et al.*, "Accelerating the machine learning lifecycle with mlflow." *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018. (Citato a pagina 40)

[34] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90. (Citato a pagina 42)

[35] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, "sktime: A unified interface for machine learning with time series," *arXiv preprint arXiv:1909.07872*, 2019. (Citato a pagina 47)

[36] R. F. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007. (Citato a pagina 61)

Ringraziamenti

Vorrei dedicare questo spazio a chi, con dedizione e pazienza, ha contribuito in qualche modo al raggiungimento di questo traguardo.

Desidero esprimere la mia gratitudine a Fabiano che mi ha accompagnato lungo il percorso di tesi al JADS e al Professore Fabio Palomba per avermi dato la possibilità di vivere una delle esperienze formative più importanti della mia carriera accademica.

Ringrazio la mia famiglia per avermi permesso di concludere questo traguardo e di appoggiarmi in ogni mia scelta di vita riempiendomi di amore.

Ringrazio Alice, per avermi dato la gioia di essere lo zio di una così meravigliosa creatura.

Ringrazio Marina e Valerio, per aver sempre creduto in me e nelle mie capacità.

Ringrazio i miei nonni, anche chi non è qui, ma sono sicuro stia facendo il tifo per me, per essere sempre stati orgogliosi di me e dei miei traguardi.

Ringrazio tutti i miei fidati amici, su cui so sempre di poter contare.