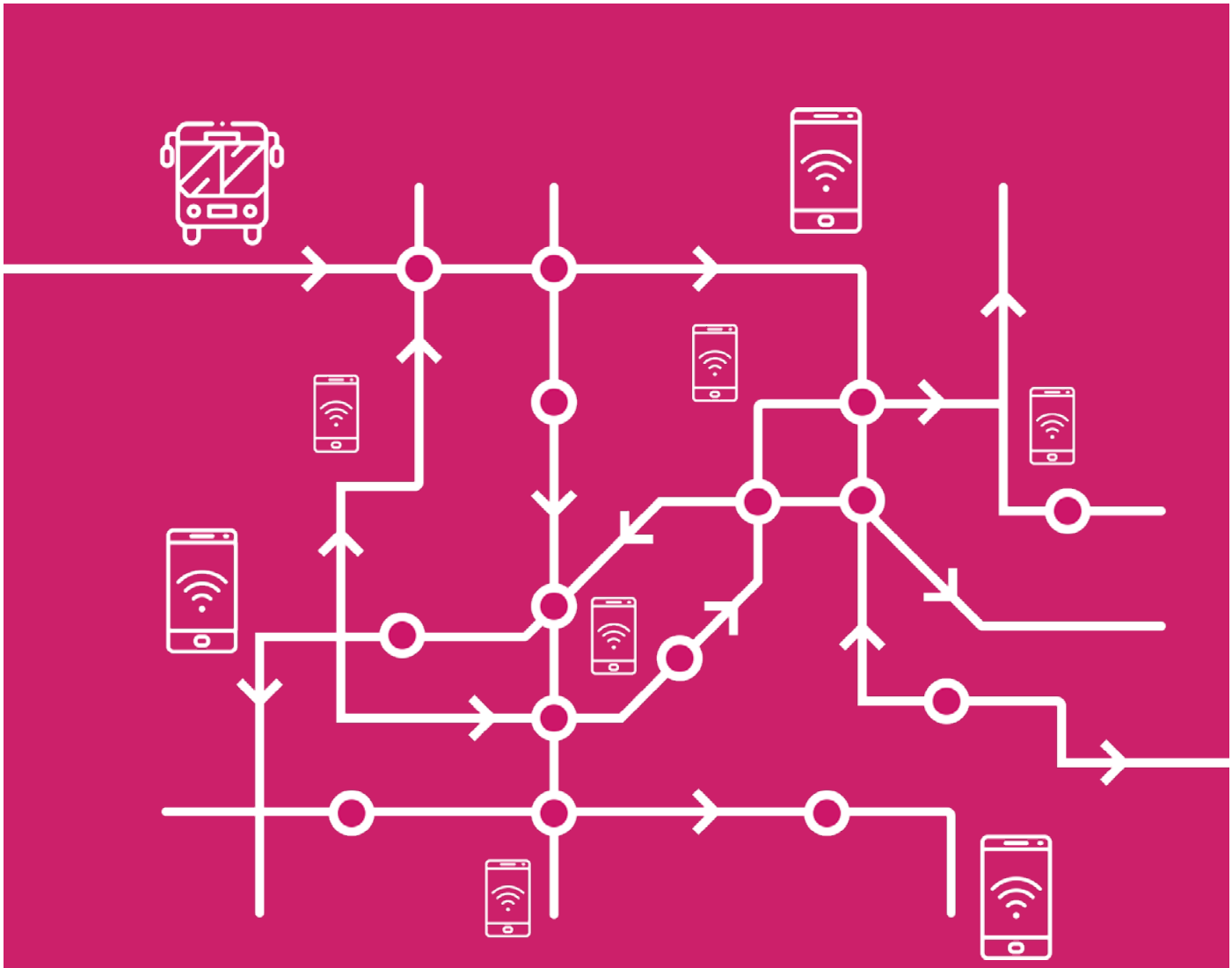


Machine Learning per il monitoraggio tramite WiFi delle presenze su mezzi pubblici

L'esempio di Ferrara nel progetto SMASH

U-Hopper in partnership con Dedagroup Public Services

U·HOPPER

Contenuti

3	**Introduzione**
7	**Raccolta dei dati**
9	**Pulizia dei dati raccolti**
10	**Analisi dei dati ripuliti**
11	Primo tentativo di approccio
12	Secondo tentativo di approccio
13	**Considerazioni**
14	**Conclusioni**

Introduzione

"Possiamo stimare il numero di passeggeri a bordo di un mezzo di trasporto pubblico monitorando i dispositivi con WiFi acceso?"

Il numero di passeggeri a bordo di un mezzo pubblico in ogni tratto del suo percorso e in un qualunque orario di un dato giorno è un'**informazione estremamente preziosa per chi si occupa di trasporto pubblico** (agenzie di mobilità, operatori di trasporto). La conoscenza di questa informazione, infatti, **consentirebbe di meglio programmare la frequenza delle corse e di scegliere la tipologia dei veicoli da impiegare** in maniera più consona alla particolare situazione in essere.

Oltre ad avere un **impatto positivo sui costi** sostenuti dalla società che opera il trasporto pubblico (e, di riflesso, su quelli della collettività che ne usufruisce), una miglior gestione dei veicoli farebbe sì che il numero di mezzi pubblici in circolazione sia sempre ottimale, riducendo il numero di mezzi che girano vuoti, con **indubbi benefici dal punto di vista ambientale e della viabilità**.

Per i mezzi pubblici dove è presente un controllo capillare all'entrata e all'uscita (pensiamo ad una metropolitana con tornelli all'ingresso), tale problema viene affrontato a partire dall'analisi dei biglietti validati.

Per mezzi tipo autobus, invece, il solo dato di bigliettazione risulta di limitata utilità: non vengono tracciate le discese ma solo le salite e in molti casi i titoli di viaggio tipo abbonamento non vengono validati.

A tal fine, diversi **metodi sono stati proposti**:

- **Fotocellule** (sensori passive infrared, PIR), tipicamente posizionati sulle porte di salita/discesa. Poco efficaci in condizioni di affollamento, in presenza di adulti con bambini, passeggini, sedie a rotelle etc.

- **Sensori di peso**, posizionati sugli ammortizzatori. Stimano la massa complessiva del veicolo e, sottraendo la massa a vuoto e il peso del carburante, permettono una stima approssimata del numero di persone trasportate. Richiedono la misurazione continua della quantità di carburante, e si basano su una approssimazione del "peso medio" per passeggero.

- **Videocamere** (stereo/3d/termocamere etc.), generalmente posizionate sul soffitto del mezzo. Sensibili a variazioni di illuminazione nel contesto ambientale (passaggi luce/ombra, lame di luce etc.); poco precise in condizioni di alto affollamento. Presentano inoltre problematiche non banali dal punto di vista della privacy (operano in una zona grigia del GDPR).

In generale, le soluzioni tradizionali presentano **livelli di precisione medio-bassi in condizioni operative realistiche** (si va dal 50-60% delle fotocellule e dei sensori di peso al 70-75% delle videocamere).

Nell'ambito del progetto SMASH, Dedagroup Public Services (DPS) si è trovata ad affrontare questo problema (*Quanti passeggeri ci sono sull'autobus?*) nel contesto della città di Ferrara.

Il problema, infatti, era stato segnalato dalla locale Agenzia Mobilità e Impianti (AMI) e dall'azienda di trasporto pubblico (TPER), interessate a trovare una soluzione alternativa ai metodi tradizionali usati negli ultimi anni (rilievi manuali a bordo mezzo).

Progetto SMASH

Il progetto SMASH (Sustainable Mobility Analysis as Service Hub) è dedicato alla costruzione di soluzioni per la mobilità sostenibile attraverso l'uso intelligente ed innovativo delle tante fonti dati già disponibili presso soggetti pubblici e privati che si occupano di gestire la tematica sui territori di competenza.

Obiettivo principale di SMASH è la creazione di servizi web di analisi geografica e temporale dei dati per rispondere ai bisogni di enti locali, di agenzie pubbliche e aziende private che necessitano di analizzare i dati di mobilità per:

- integrare i dati di mobilità con altri dati (es. qualità dell'aria);
- ottimizzare la pianificazione dei servizi e delle infrastrutture per la mobilità;
- monitorare e misurare gli impatti dei progetti di mobilità sostenibile;
- ottimizzare e rendere più sostenibili i percorsi del trasporto pubblico;
- migliorare la fruibilità dei servizi di mobilità;
- coinvolgere i cittadini in iniziative di mobilità sostenibile.

Co-finanziato dal network EIT Climate KIC, SMASH ha visto Dedagroup Public Services come azienda capofila del progetto che sviluppa le attività in partnership con l'Agenzia per l'Energia e lo Sviluppo Sostenibile di Modena, Forum Virium Helsinki, Fondazione Bruno Kessler di Trento e con l'azienda britannica BetterPoints.

Grazie al coinvolgimento di U-Hopper, Dedagroup Public Services è riuscita a progettare e testare una soluzione in grado di **determinare il numero di passeggeri a bordo di un mezzo pubblico a partire dall'analisi dei pacchetti emessi da device portatili equipaggiati con un'interfaccia WiFi** (Smartphone, per esempio).

Il progetto si è sviluppato attraverso:

a) l'**elaborazione dei dati** sugli spostamenti reali dei mezzi pubblici, svolta da Dedagroup utilizzando gli **algoritmi** sviluppati nel progetto SMASH, e

b) l'**analisi**, svolta da U-Hopper tramite **tecniche di Intelligenza Artificiale e più precisamente di Machine Learning**, del numero di dispositivi rilevati via via a bordo del mezzo pubblico, in modo da ricavare da tale grandezza una stima il più possibile accurata del numero effettivo di passeggeri a bordo del veicolo in ogni dato istante.

Nelle sezioni qui di seguito vengono illustrate le diverse fasi di cui si è composto il progetto: la raccolta dei dati, la loro pulizia e, infine, la validazione sperimentale condotta.

Le fasi del progetto

1. Raccolta

il progetto inizia con la raccolta dei dati, che costituiranno la verità di base su cui costruire l'algoritmo di ML: tra i dati utilizzati ci sono le letture rilevate dal sensore, i conteggi effettuati a bordo mezzo e i tracciati degli spostamenti reali degli autobus.

2. Pulizia

si procede con la preparazione e pulizia dei dati, per renderli omogenei, strutturati e per correggere eventuali inconsistenze ed errori.

3. Training

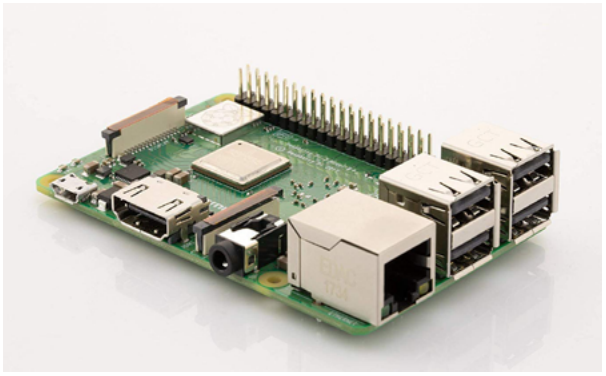
vengono effettuati dei training test per individuare la forma funzionale e i relativi parametri e variabili.

4. Testing

la funzione identificata viene testata per verificare la bontà del modello. Se necessario, vengono apportate modifiche e ripetuti i test, fino a trovare la funzione e modello ottimale per poter fare previsioni basate su nuovi dati.

Raccolta dei dati

Per procedere alla rilevazione dei pacchetti emessi da device portatili in modo da poter determinare il numero di questi ultimi a bordo del mezzo pubblico, U-Hopper ha provveduto alla **programmazione di appositi sensori e alla loro successiva installazione a bordo di alcuni autobus circolanti** nella città di Ferrara su determinate linee urbane (più precisamente, le linee 6, 9 e 11).



Tali sensori erano costituiti da dei *Raspberry Pi 3* (ovvero da una singola scheda elettronica implementante un intero computer) dotati del sistema operativo nativo di Raspberry, basato su Unix, ai quali è stata aggiunta una scheda WiFi 802.11 esterna per poter rilevare la trasmissione di dati wireless tramite tale protocollo di comunicazione.

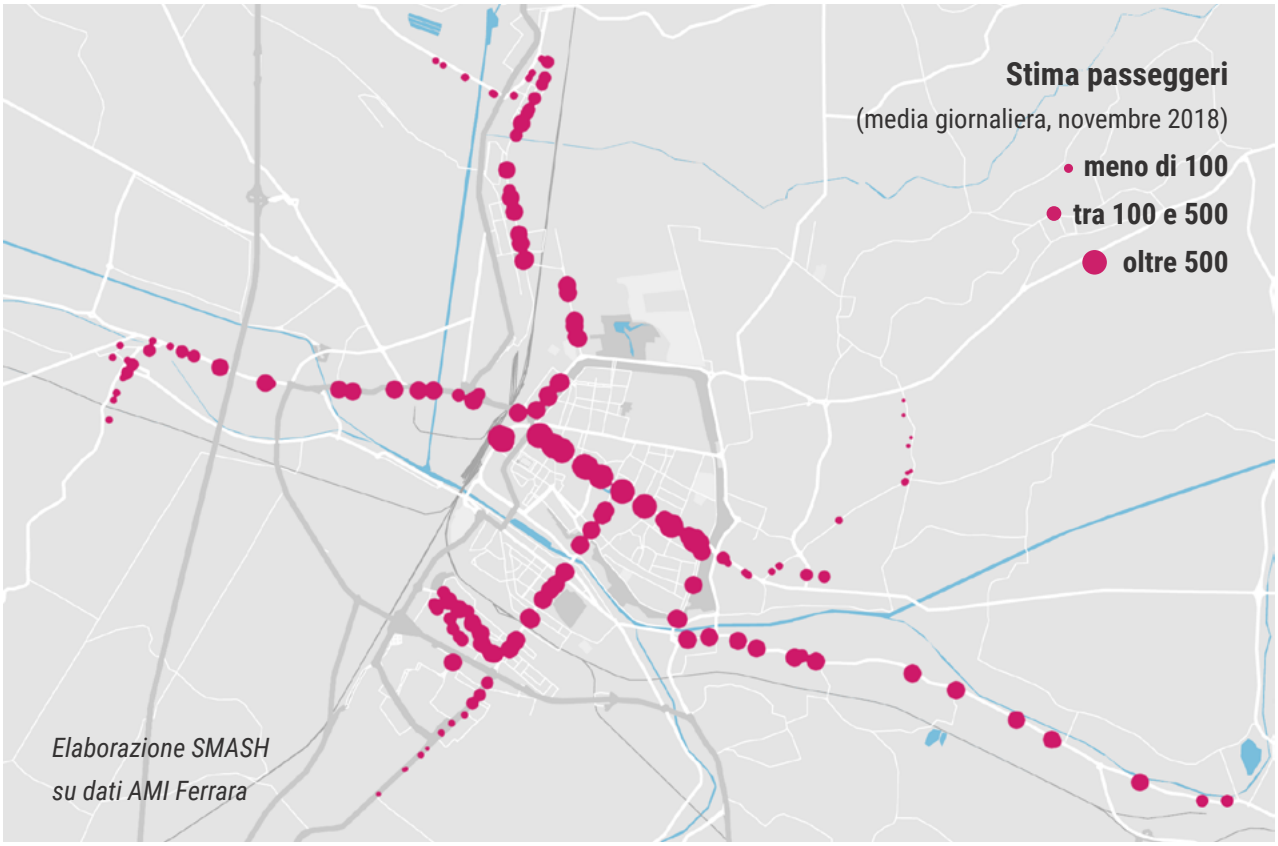
I dati raccolti erano costituiti dai seguenti campi, per ogni pacchetto ricevuto:

- il **timestamp del pacchetto**, ovvero l'istante in cui il pacchetto WiFi è stato rilevato;
- il **MAC address del dispositivo emittente**, un codice lungo 48 bit che identifica univocamente il dispositivo emittente;
- l'**RSSI**, che indica la potenza con la quale il segnale radio è stato ricevuto (e quindi permette di stimare, almeno in modo grossolano, la distanza tra emettitore e ricevitore);
- altre informazioni non utili ai fini dell'analisi svolta.

Contestualmente, nei giorni 22, 25 e 26 novembre 2018, **alcuni operatori di DPS hanno provveduto alla registrazione manuale del numero di passeggeri a bordo dei medesimi autobus**, tenendo traccia del numero di passeggeri saliti a bordo e discesi ad ogni fermata.

I **dati registrati manualmente hanno costituito ciò che, per usare il gergo del ML, è la cosiddetta "*ground truth*" (o "*verità di base*")**, utile a calibrare l'algoritmo in modo che questo possa restituire una stima accurata della quantità desiderata - nel caso in esame, del numero di passeggeri a bordo di un mezzo pubblico. Una volta effettuata la calibrazione, sarà sufficiente conoscere il numero di pacchetti rilevati dai sensori per ottenere il numero di passeggeri, senza più bisogno di alcuna misurazione da parte di operatori.

In aggiunta a ciò, l'agenzia di trasporto ha fornito sia le tabelle con gli orari programmati sia i dati degli spostamenti reali degli autobus (derivati dai sistemi di rilevazione GPS già presenti a bordo mezzo) sui quali è stata effettuata la rilevazione, per conoscere i momenti di passaggio nelle varie fermate.



3
linee urbane
coinvolte

3
giornate
di raccolta dati

185k
indirizzi MAC
rilevati

Pulizia dei dati raccolti

Conclusa la fase di raccolta dei dati, si è resa indispensabile la pulizia dei dati stessi, sia quelli relativi alle tabelle orarie degli autobus, sia alle rilevazioni dei sensori.

Per quanto riguarda i primi, i dati erano costituiti dall'orario programmato di passaggio, ovvero dalla tabella oraria "teorica" distribuita ai clienti e nelle fermate, e dall'orario effettivo di passaggio. Quest'ultimo dato rappresentava l'informazione più utile, in quanto permetteva di verificare il momento esatto in cui i passeggeri salivano o scendevano dai mezzi. Tuttavia, tali dati erano incompleti ed e stato pertanto necessario approssimare il momento di passaggio nella maggior parte delle fermate, **unendo gli orari "teorici"** con i dati reali effettivamente disponibili.

Per mettere in relazione gli orari programmati con quelli reali è stato necessario ricostruire come la stessa corsa viene rappresentata nei due gruppi di dati, dal momento che le informazioni di linea, corsa e mezzo vengono descritte in modo diverso.

Per quanto riguarda i dati raccolti dai sensori, diverse fasi di pulizia sono state attuate.

Innanzitutto, è stato applicato un filtro in base al valore dell'RSSI, andando ad **eliminare quei pacchetti ricevuti con scarsa potenza, e quindi con grande probabilità non provenienti dall'interno dell'autobus**.

In secondo luogo, sono stati eliminati quei dispositivi che apparivano per un periodo di tempo continuativo troppo lungo (più di 45 minuti, ad esempio l'autista, o gli operatori stessi che effettuavano la rilevazione manuale) o troppo corto (non più di 30 secondi, ad esempio il dispositivo all'interno di un'automobile affiancata al semaforo).

Infine, sono stati eliminati quei dispositivi apparsi troppe volte all'interno della stessa giornata: quest'ultimo caso è servito per rimuovere dati provenienti, ad esempio, da attività commerciali lungo il percorso dotate di punti d'accesso WiFi, o da reti domestiche vicine alla strada.

Analisi dei dati ripuliti

Da un punto di vista squisitamente matematico, l'obiettivo dell'analisi dati è l'individuazione di una funzione che, ricevuti in input il numero di device rilevati dai sensori in un dato istante su un determinato mezzo pubblico e una serie di informazioni - o variabili - corollarie (si veda al riguardo il seguito di questa sezione), restituisca come risultato il numero di passeggeri a bordo del veicolo nell'istante in questione.

Tale funzione può assumere vesti matematiche diverse, ciascuna delle quali conterrà - in generale - una serie di parametri (o coefficienti) che andranno opportunamente calibrati. Per determinare la forma funzionale - con relativi coefficienti - in grado di stimare nella maniera più accurata il numero di passeggeri, vengono impiegate le misurazioni effettuate manualmente dagli operatori.

Più precisamente, **seguendo una prassi consolidata del ML, le misurazioni manuali sono state divise in due gruppi tenuti rigidamente separati, i cosiddetti *training set e test set***: il primo è stato impiegato per individuare forma funzionale e relativi parametri (fase di training), il secondo per testare il comportamento della funzione individuata su dati "nuovi", dati - cioè - che non abbiano avuto alcuna influenza nella determinazione della funzione (fase di test).

In questo modo, si simulano quelle che saranno le condizioni operative dell'algoritmo il quale, a lungo termine, dovrà infatti elaborare rilevazioni dei sensori del tutto originali e, soprattutto, non corroborate da contestuali misurazioni manuali (come già puntualizzato nella sezione dedicata alla raccolta dei dati).

Prima di procedere allo studio della funzione di cui sopra, **in via preliminare si è determinato se l'ipotesi di base** - impiegare il numero di device rilevati a bordo di un mezzo come *proxy* per il numero di passeggeri a bordo del medesimo - avesse fondamento o meno.

Per questo motivo si è misurata la correlazione ρ tra queste due quantità, ottenendo un valore di ρ pari a circa 0,43 che denota un'effettiva relazione tra le due grandezze[1].

Per valutare la bontà della funzione individuata, si è ricorso al cosiddetto *coefficiente R*[2]: senza entrare troppo nel dettaglio, sarà sufficiente ricordare come tale coefficiente misuri la differenza media tra il numero di passeggeri stimato a partire dalle rilevazioni dei sensori e quello misurato manualmente dagli operatori di DGS - **più il valore di R^2 è prossimo a 1**, minore è la differenza tra il numero di passeggeri stimato e quello effettivo, **migliore dunque è l'accuratezza dell'algoritmo**.

Primo tentativo di approccio

In un primo tempo, nella fase di training sono state utilizzate come variabili in input il numero di device rilevati e le seguenti **variabili corollarie: giorno della settimana, ora del giorno in cui è stata effettuata la rilevazione e linea urbana del mezzo pubblico in questione.**

Per quanto riguarda le rilevazioni dei sensori, invece di fornire in input il numero "crudo" di device rilevati ottenuto dopo la fase di pulizia dei dati, **si è proceduto preliminarmente alla decomposizione della corrispondente serie temporale**[2]. Tale decomposizione ha consentito, per ogni misura del numero di device, di ricavarne il *trend* (la misura di quanto, in ogni dato istante, il numero di device stia aumentando/diminuendo) e la *stagionalità* (la componente che si ripete identica a intervalli regolari). **Proprio trend e stagionalità - assieme alle variabili corollarie di cui è si è discusso in precedenza - hanno rappresentato l'effettivo input dell'algoritmo.**

Con questo approccio, il coefficiente R^2 è risultato essere pari a circa 0,62: seppur non scarso, questo valore - comunque migliorabile - ha indotto U-Hopper a tentare di individuare un algoritmo possibilmente più efficace.

[1] Giova qui ricordare come la correlazione tra due variabili sia una grandezza matematica che può variare tra -1 e 1: valori molto prossimi a zero denotano una sostanziale indipendenza delle variabili in esame, diversamente le variabili sono caratterizzate da andamenti similari - il che giustifica l'impiego dell'una per studiare il comportamento dell'altra.
[2] Una serie temporale non è altro che una sequenza di misure di una data quantità effettuate in istanti di tempo successivi l'uno all'altro: è questo il caso del numero di device rilevati dai sensori.

Le variabili corollarie

I motivi per cui le variabili *giorno della settimana (d), ora del giorno (h) e linea urbana (l)* sono stati ritenuti rilevanti e utilizzati come input alla funzione sono facilmente comprensibili: la frequentazione media di un mezzo pubblico varia infatti a seconda del giorno della settimana (per esempio, in un giorno festivo i veicoli dovrebbero essere meno affollati rispetto a un giorno lavorativo), dell'ora (alla sera tardi il numero di passeggeri dovrebbe essere inferiore rispetto alle fasce orarie corrispondenti all'inizio e alla fine della giornata lavorativa) e della linea urbana (una linea urbana che serva un polo scolastico avrà un picco di frequentazione nel primissimo pomeriggio al termine delle lezioni, una invece che serva un'area industriale avrà un picco al termine dei vari turni di lavoro).

Secondo tentativo di approccio

Una seconda strategia che è stata sperimentata è la seguente: invece di impiegare il giorno della settimana *d*, l'ora del giorno *h* e la linea urbana *l* come input alla funzione, queste quantità sono state utilizzate per dividere i dati presenti nel *training set* in vari gruppi sulla base delle diverse combinazioni (*d, h, l*) e poi, per ciascun sottogruppo, si è individuata - in maniera indipendente dagli altri sottogruppi - **una forma funzionale più specifica, avente ora come input i soli trend e stagionalità.**

Questa "personalizzazione" della forma funzionale - non più un'unica funzione da applicare a ogni rilevazione, bensì diverse funzioni, una per ogni combinazione (*d, h, l*) - introduce maggior flessibilità nell'intero approccio, consentendo all'algoritmo di meglio adattarsi alle varie situazioni. **Questa duttilità è risultata in un coefficiente R^2 pari a 0,79, sensibilmente migliore rispetto all'approccio precedente.**

5
variabili
di input

655
parametri
di input

2,9%
errore relativo
su corse a pieno carico

0,79
R^2

Considerazioni

Nel contesto di questo progetto, la rilevazione manuale iniziale - indispensabile, come messo in evidenza precedentemente, per una buona calibrazione dell'algoritmo - è stata effettuata nell'arco di soli tre giorni unicamente su tre linee urbane. Questo fatto, legato a restrizioni temporali e finanziarie (l'impiego continuo di operatori comporta senza dubbio un investimento iniziale relativamente oneroso), ha reso nel complesso piuttosto esiguo il set di dati utili all'elaborazione - circa 3.500 rilevazioni complessivamente.

Per rendere ancora più accurato l'algoritmo, sarebbe opportuno dislocare operatori per le rilevazioni manuali su più linee urbane e per un numero maggiore di giorni, idealmente distribuiti nell'intero arco dell'anno solare in modo da poter meglio cogliere le inevitabili fluttuazioni stagionali.

Un ulteriore accorgimento per rendere più efficace l'analisi potrebbe consistere nell'integrare l'attuale dataset con dati provenienti da altre fonti che dovrebbero però influire sulla frequentazione dei mezzi pubblici (per esempio le condizioni meteorologiche nella città di Ferrara nei giorni monitorati nel presente studio).

Conclusioni

I risultati ottenuti dall'elaborazione dei dati raccolti dai sensori hanno confermato la bontà dell'idea alla base dell'intero progetto, che consiste nello **stimare il numero di passeggeri a bordo di un mezzo pubblico a partire dal numero di dispositivi WiFi rilevati sul mezzo stesso, utilizzando tecniche di Machine Learning**. Questa intuizione si rivela particolarmente efficace nel momento in cui non si fa uso di un'unica forma funzionale da applicare a ogni rilevazione ma quando si impiegano più forme funzionali, ciascuna specifica per una determinata combinazione *giorno della settimana - ora del giorno - linea urbana*.

In ultimo luogo, un **aspetto importante da sottolineare a favore della soluzione è sicuramente la facilità di implementazione dei sensori per il rilevamento dei device portatili e il loro costo contenuto**. Essi possono essere infatti attivati in qualunque momento e per un intervallo di tempo sufficientemente ampio (in questo senso, l'unica limitazione viene dalla durata delle batterie) senza particolari accorgimenti o costi eccessivi. Ciò, in linea di massima, rende il numero di queste rilevazioni piuttosto elevato, il livello di precisione più alto e quindi la soluzione più competitiva rispetto a quelle tradizionali (fotocellule, sensori di peso e fotocamere).

Nonostante un investimento *iniziale* temporale ed economico rilevante (ma tuttavia contenuto, si pensi all'impiego degli operatori per le rilevazioni manuali iniziali e al tempo necessario per le fasi di pulizia dati, test e training dell'algoritmo), **la soluzione proposta assicura risultati puntuali e facilmente traducibili in azioni data-driven volte all'ottimizzazione dei costi e della viabilità** per le aziende coinvolte nella gestione del trasporto pubblico.

L'utilizzo di tecniche di Machine Learning è la chiave dell'intero progetto. Il Machine Learning ha permesso di stimare con precisione il numero di passeggeri sui mezzi pubblici nella città di Ferrara, sulla base dei pacchetti emessi da device portatili tramite WiFi.

L'**impiego di queste tecniche**, tuttavia, non si limita a questo singolo caso e settore. Il Machine Learning **trova infatti applicazione in molteplici ambiti, trasversalmente in ogni settore**!

Contattaci per **valutare insieme le opportunità** per la tua azienda e farti consigliare **quali progetti sviluppare sfruttando tutto il suo potenziale!**

www.u-hopper.com

info@u-hopper.com

SEDE OPERATIVA
Via R. da Sanseverino, 95
38122 Trento (TN) - Italy c/o Impact Hub

U·HOPPER