Document

Pathetum

Riferimento	2024_Modulo_Pathetum_C09
Versione	1.0
Data	21/03/2025
Destinatario	Top management
Presentato da	G. Ruocco, P. D'Antuono, F. C. Ponticelli, G. Pastena
Approvato da	Top Management

GitHub repository: https://github.com/PDant17/GreenBottlePathetum

Team Members

Nome	Ruolo	Acronimo	Informazioni di contatto
Giovanni Ruocco	Team Member	GR	g.ruocco45@studenti.unisa.it
Pietro D'Antuono	Team Member	PD	p.dantuono2@studenti.unisa.it
Fabio Catello Ponticelli	Team Member	FP	f.ponticelli2@studenti.unisa.it
Giuseppe Pastena	Team Member	GP	g.pastena1@studenti.unisa.it

Revision History

Data	Versione	Descrizione	Autori
06/03/2025	0.1	Aggiunta la sezione "Introduzione"	TUTTI
06/03/2025	0.2	Aggiunta la sezione "Descrizione dell'agente"	TUTTI
06/03/2025	0.3	Creazione logo	GP
07/03/2025	0.4	Aggiunta la sezione "Raccolta, analisi e preprocessing dei dati"	TUTTI
10/O3/2025	0.5	Aggiunta la sezione "Algoritmi di clustering"	TUTTI

10/O3/2025	0.5.1	Aggiunta la sezione “K-Means”	PD
10/O3/2025	0.5.2	Aggiunta la sezione “Clustering Gerarchico Agglomerativo”	GP
10/O3/2025	0.5.3	Aggiunta la sezione “DBSCAN”	GR
10/O3/2025	0.5.4	Aggiunta la sezione “Conclusioni”	FP
13/03/2025	0.6	Aggiunta la sezione “Integrazione con il sistema”	TUTTI
13/03/2025	0.7	Aggiunta la sezione “Migliorie future”	TUTTI
21/03/2025	1.0	Revisione finale	TUTTI

Sommario

1 Introduzione

La piattaforma GreenBottle dell'azienda AquaPure si prefigge, come scopo principale, quello di effettuare consegne a domicilio di bevande in bottiglie di vetro. Per fare ciò si serve di un certo numero di corrieri, i quali, per essere ancor più rispettosi delle tematiche green, usano veicoli elettrici. Diventa quindi necessario stabilire la strategia migliore per effettuare le loro numerose consegne, garantendo un servizio rapido, affidabile e che non carichi di lavoro eccessivo i corrieri a disposizione. Questo problema può essere risolto grazie all'aiuto dell'Intelligenza Artificiale, che fornirà agli Admin ed ai Corrieri di GreenBottle le soluzioni ottimali ai loro problemi.

2 Descrizione dell'agente

2.1 Obiettivo

Lo scopo dell'agente è quello di calcolare:

- Il miglior raggruppamento di ordini possibile per i corrieri a disposizione.
- Il percorso ottimale che ciascun corriere deve seguire per consegnare gli ordini assegnati.
- Il numero ottimale di corrieri che andrebbero usati per gli ordini ricevuti, nel caso questo sia maggiore del numero di corrieri effettivamente a disposizione dell'azienda.

2.2 Specifica PEAS

Performance	La misura di prestazione per valutare l'operato dell'agente. In questo caso indica: • La minimizzazione dei costi operativi per l'azienda, che tiene conto del numero di corrieri usati e dei loro percorsi. • La distribuzione uniforme del carico di lavoro tra i corrieri per evitare sovraccarichi o inefficienze.
Environment	L'ambiente in cui opera l'agente è un'astrazione di un'area geografica ed ha le seguenti caratteristiche: • Completamente osservabile: l'agente ha accesso a tutte le informazioni relative agli ordini. • Deterministico: ogni azione produce un risultato prevedibile in base alle informazioni fornite. Episodico: ogni ciclo di consegna è trattato come un episodio indipendente. • Statico: le informazioni di base, cioè ordini ed indirizzi di consegna, non cambiano durante la pianificazione. • Discreto: l'agente opera su un insieme finito di input e output, cioè ordini, indirizzi di consegna e percorsi. • A singolo agente.
Actuators	Sono gli attuatori a disposizione dell'agente per eseguire le azioni. In questo caso sono: • L'assegnazione degli ordini ai singoli corrieri. • La definizione dei percorsi ottimali che ogni corriere deve seguire.
Sensors	Sono i sensori attraverso cui l'agente riceve gli input. In questo caso sono: • Dati relativi agli ordini ricevuti.

2.2 Analisi del problema

Il problema che affrontiamo, per quanto presenti diversi aspetti di cui tener conto, può essere visto come un problema di clustering. Ci troviamo infatti a dover suddividere in gruppi un insieme di ordini da assegnare a dei corrieri, per cui il clustering, che si occupa della categorizzazione simbolica o numerica, si presta particolarmente bene a questo scopo.

- Tramite il clustering potremo trovare sia il miglior numero di corrieri da utilizzare che il miglior raggruppamento possibile per quelli che abbiamo a disposizione.
- Successivamente, per ciascun cluster generato, potremo cercare il percorso ottimale che il corriere dovrà seguire.

3 Raccolta, analisi e preprocessing dei dati

I dati utilizzati per questo progetto sono interamente sintetici, generati al fine di simulare scenari realistici per l'ottimizzazione delle consegne.

L'approccio al problema parte dal presupposto che i dati aziendali siano disponibili. Tuttavia, non avendo accesso a un dataset reale dell'azienda, si è scelto di simulare un ambiente che riproduca fedelmente una realtà aziendale esistente. Per fare ciò, abbiamo ipotizzato l'esistenza di un'azienda virtuale che opera nel nostro territorio, ovvero la Campania. Abbiamo quindi selezionato un'area rappresentativa del territorio campano, suddividendola in zone strategiche, tra cui città metropolitane, aree rurali e paesini di campagna. In queste zone sono stati posizionati ordini sintetici in percentuali differenti, seguendo un criterio di randomicità controllata. In particolare, si è data maggiore densità di ordini nelle aree urbane, mentre si è mantenuta una distribuzione più sparsa nelle aree rurali e nei piccoli centri.

Questa simulazione è stata progettata per essere quanto più realistica possibile, al fine di generare diverse iterazioni del problema nel tempo. L'obiettivo è stato ricreare un caso d'uso rappresentativo di uno scenario operativo reale, considerando che la strategia di distribuzione e ottimizzazione dipende in larga misura dalla posizione e dalla distribuzione degli ordini stessi.

3.1 Scelta del dataset

Per sviluppare un efficace sistema di ottimizzazione delle consegne, è fondamentale utilizzare un dataset contenente informazioni dettagliate e rilevanti. Nello specifico, il dataset selezionato dovrà includere:

- **Ordini:** informazioni relative ai destinatari, dettagli sul contenuto e indirizzi di consegna.
- **Corrieri:** informazioni riguardanti il numero di corrieri disponibili per effettuare le consegne.

La presenza di queste variabili nel dataset permetterà di implementare modelli di clustering e ottimizzazione più accurati e realistici, contribuendo a migliorare significativamente la qualità del servizio offerto e a contenere i costi operativi aziendali.

3.2 Analisi del dataset

L'analisi preliminare del dataset è una fase fondamentale che consente di valutare la qualità dei dati generati, identificare eventuali incongruenze e scoprire pattern utili per l'ottimizzazione del servizio. Durante questa fase, si procederà con la rimozione delle informazioni non pertinenti ai calcoli richiesti, come flag relativi ai ritiri, al supporto aggiuntivo e descrizioni non essenziali. I dati considerati essenziali per il modello includono:

- **Indirizzo di consegna**, che sarà oggetto di pre-processing per garantire uniformità e precisione.
- **ID dell'ordine**, che verrà utilizzato nella fase finale per identificare chiaramente ciascuna consegna.

Questa analisi faciliterà una gestione ottimale dei dati nella successiva fase di clustering e nell'ottimizzazione dei percorsi di consegna.

4 Algoritmi di clustering

4.1 K-Means

L'algoritmo **K-Means** è uno degli algoritmi di clustering più popolari per dati numerici ed ha lo scopo di suddividere il dataset in k cluster per minimizzare la varianza interna. Si tratta di un algoritmo efficiente, facile da implementare e che si comporta molto bene con dataset anche molto grandi.

Vantaggi

- Ottima efficienza computazionale: la complessità temporale è di O(n * k * t), con n numero di punti, k numero di cluster e t numero di iterazioni. Questo lo rende, generalmente, più veloce e scalabile delle altre opzioni.

Svantaggi

- Può non funzionare correttamente nel caso i dati siano distribuiti in modo irregolare, come nel caso di zone ad alta densità, in quanto non gestirebbe in modo ottimale la varianza.

4.2 Clustering Gerarchico Agglomerativo

Il clustering **gerarchico agglomerativo** è un metodo che costruisce una struttura ad albero in modo bottom-up detta dendrogramma per rappresentare una gerarchia di cluster. La costruzione avviene in modo bottom-up, ciascun nodo rappresenta un cluster ed i livelli successivi indicano, in questo caso, delle unioni.

Vantaggi

- Non è richiesto specificare un numero iniziale di cluster, il che è utile quando non si ha conoscenza preliminare di quanti ne possano esistere.
- La rappresentazione tramite dendrogramma consente di comprendere visivamente le relazioni tra i dati e facilita il monitoraggio dell'evoluzione del processo di clustering.
- Può rilevare cluster di forme arbitrariamente complesse e non fa assunzioni sulla distribuzione dei dati.

Svantaggi

- La complessità computazionale del clustering gerarchico è generalmente $O(n^2)$, dove n è il numero di punti dati. Questo lo rende poco pratico su dataset molto grandi.
- Il clustering gerarchico può essere sensibile al rumore e agli outlier, che possono influire significativamente sulla qualità dei cluster risultanti.

4.3 DBSCAN

L'algoritmo **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) è uno degli algoritmi di clustering basati su densità più popolari per dati numerici e si basa sulla regolazione di due parametri: la distanza massima epsilon, che definisce un intorno circolare, e minPts, che stabilisce il numero minimo di punti per considerare denso un intorno di un punto. È particolarmente efficace per identificare cluster di forma arbitraria e per rilevare outlier, dunque è utile per analisi esplorative su dataset con cluster di forma irregolare e distribuzioni non uniformi.

Vantaggi

- È in grado di rilevare strutture complesse, come cluster allungati o di densità variabile.
- Classifica automaticamente come rumore i punti isolati, permettendo una rilevazione automatica degli outlier.
- Non è richiesto specificare un numero iniziale di cluster, il che è utile quando non si ha conoscenza preliminare di quanti ne possano esistere.

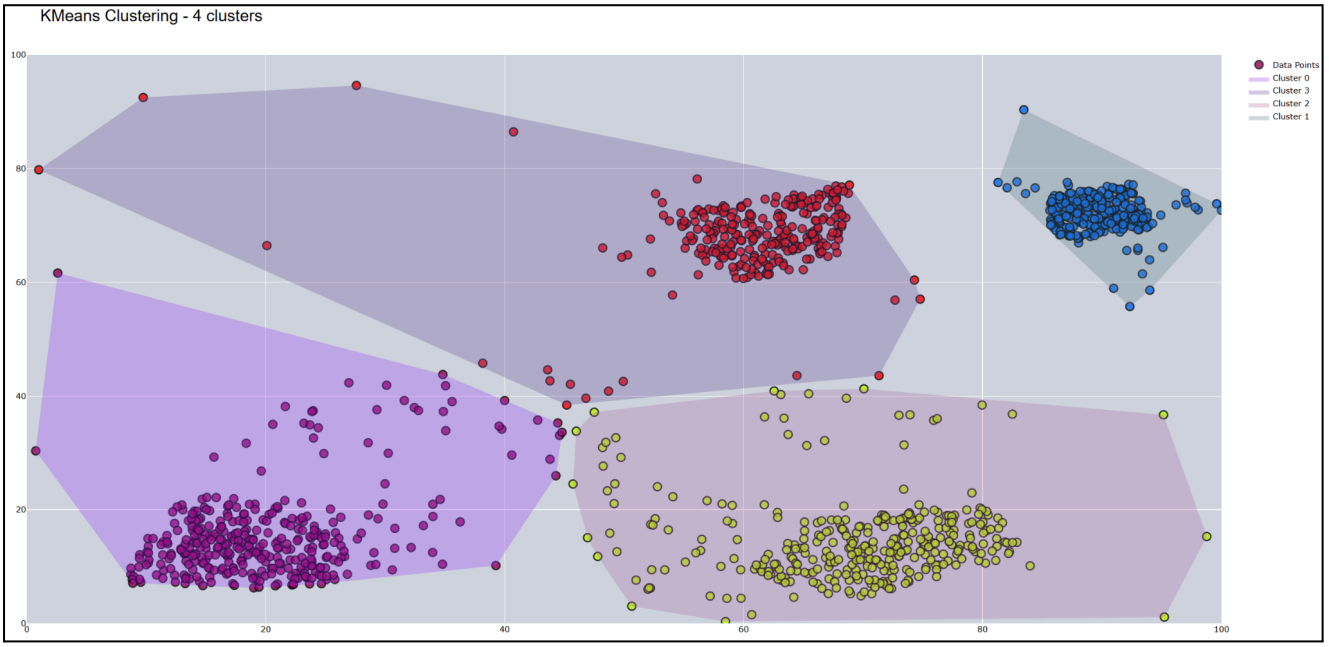
Svantaggi

- La qualità del clustering dipende fortemente dalla scelta dei valori di epsilon e minPts, che possono portare ad ottenere troppi cluster oppure cluster troppo grandi.
- Nei casi peggiori ha complessità O(n^2), rendendolo poco adatto a dataset molto grandi, soprattutto quando la metrica di distanza è complessa.
- Poiché la soglia di densità è fissa, potrebbe incorrere a problemi se i cluster hanno densità molto diverse tra loro.

4.4 Conclusioni

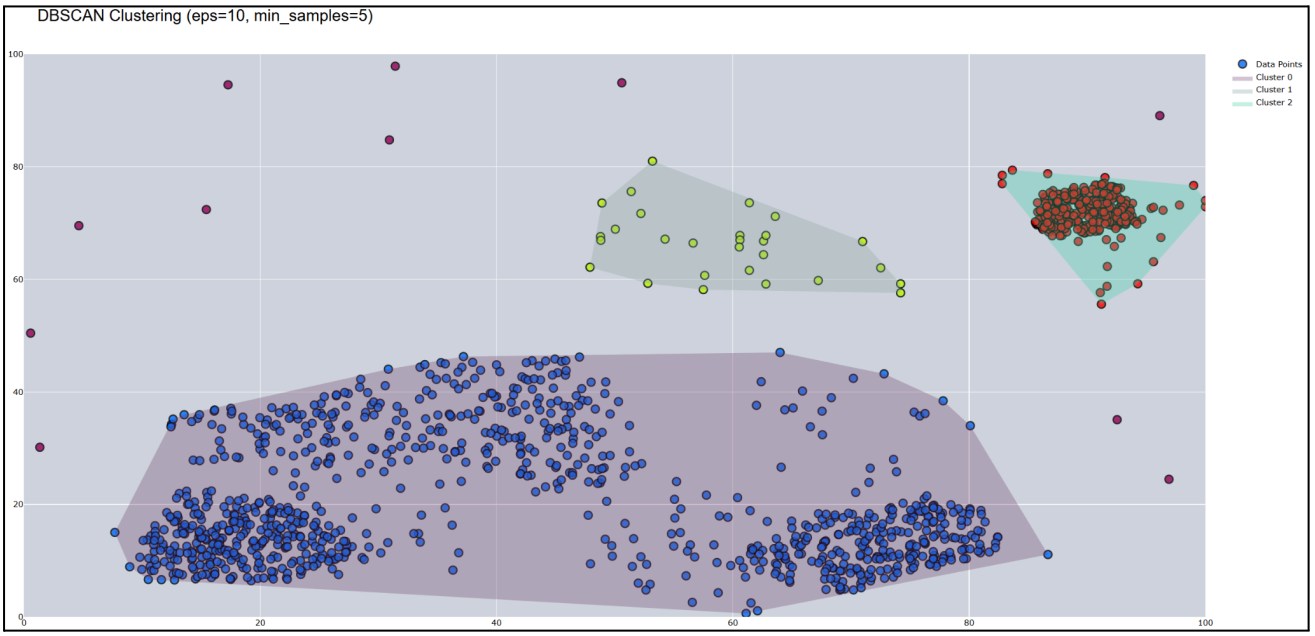
Nell'ambito del problema di GreenBottle, dove è necessario ottimizzare la distribuzione degli ordini su un territorio definito e con un numero di corrieri predefinito, K-Means emerge come la scelta ideale. La sua efficienza computazionale, unita alla capacità di gestire in modo rapido e preciso un numero fissato di cluster, lo rende perfetto per ottimizzare i percorsi e ridurre i costi operativi. A differenza del clustering gerarchico, che risulterebbe più lento e complesso da gestire, K-Means offre una soluzione scalabile e semplice, rispondendo in modo efficace alle esigenze di un sistema di consegne dinamico e a tempo.

4.4.1 K-Means



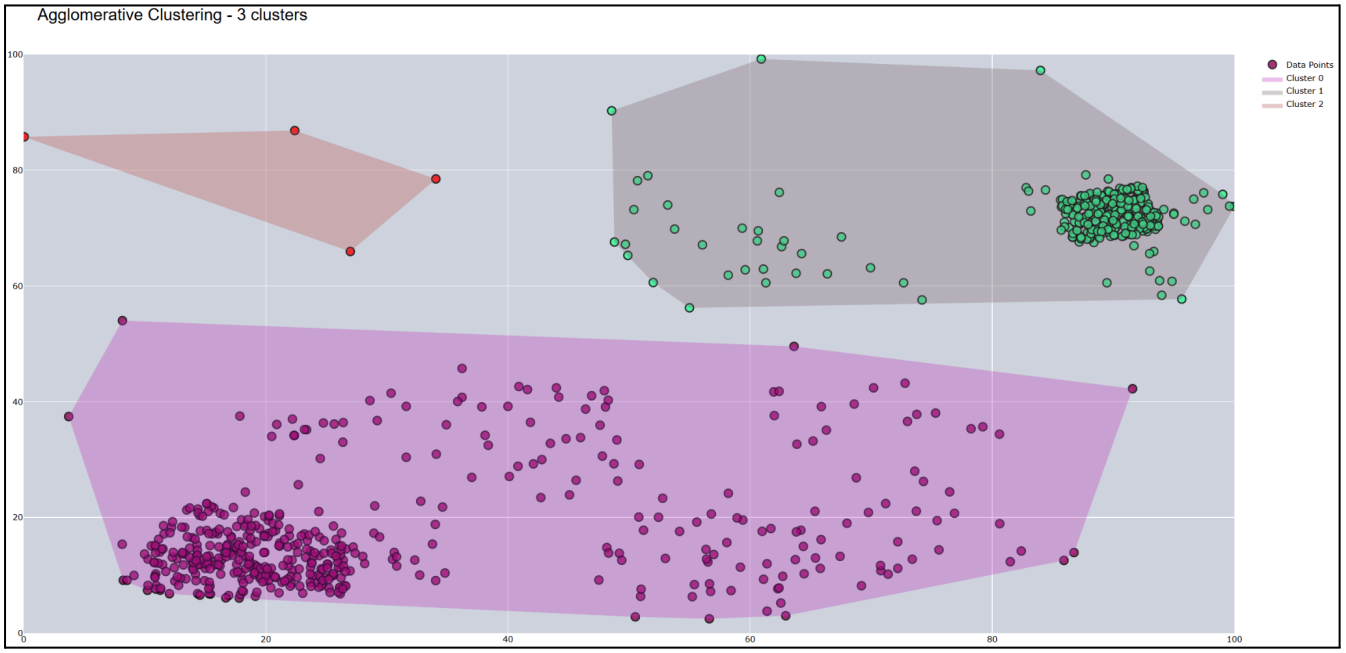
L'analisi del clustering con K-Means (k=4) mostra una segmentazione efficace, con un **silhouette score di 0.75**, indice di buona coesione interna e separabilità tra i gruppi. I cluster risultano distinti e ben definiti, con una distribuzione coerente rispetto alla struttura dei dati. Tuttavia, la forma di alcuni cluster potrebbe non essere perfettamente sferica, una limitazione intrinseca dell'algoritmo. Complessivamente, il risultato è solido e adeguato all'analisi.

4.4.2 DBSCAN



L'analisi del clustering con **DBSCAN** (eps=10, min_samples=5) evidenzia una segmentazione efficace dei dati, con un **silhouette score di 0.69**, inferiore rispetto a K-Means ma comunque accettabile. Il metodo ha individuato cluster di forma irregolare e ha identificato diversi **outlier** (punti non assegnati a nessun cluster), aspetto utile in presenza di dati rumorosi. Tuttavia, l'ampia dispersione di un cluster potrebbe indicare la necessità di ottimizzare i parametri. Complessivamente, DBSCAN fornisce un'alternativa valida ma non accettabile.

4.4.3 Agglomerative Clustering



L'analisi del clustering con **Agglomerative Clustering** (k=3) evidenzia una segmentazione **sbilanciata** e non ottimale. Un cluster è significativamente più denso rispetto agli altri, suggerendo una distribuzione non equilibrata degli ordini tra i corrieri. Questo potrebbe portare a inefficienze operative, come sovraccarico su alcuni corrieri e sottoutilizzo di altri. Inoltre, la scelta di k=3 sembra semplificare eccessivamente la struttura dei dati, riducendo la granularità necessaria per un'allocazione equa delle consegne. **Questa soluzione non è accettabile** per una gestione bilanciata degli ordini.

5 Integrazione con il sistema

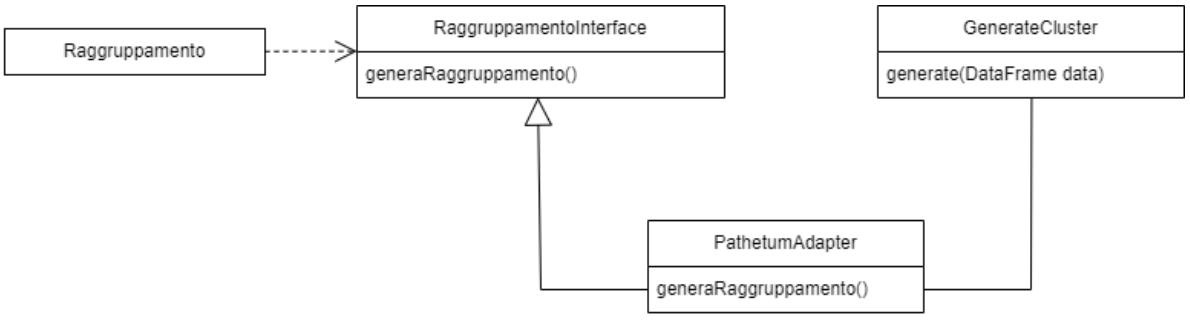
L'Adapter è un design pattern strutturale, cioè un pattern che si occupa del modo in cui le classi e gli oggetti formano strutture complesse e che riduce l'accoppiamento tra classi tramite l'incapsulamento e la creazione di classi astratte che facilitino estensioni future.

Per tali caratteristiche, l'Adapter è incredibilmente utile per quanto riguarda l'integrazione delle componenti off the shelf, che possono essere incapsulate affinché restino separate dal sistema. Il progetto godrà dunque di un elevato livello di disaccoppiamento e sarà poco impattato dalle componenti esterne.

Per implementarlo c'è bisogno di due componenti: un'interfaccia i cui metodi sono implementati in termini di richieste alla componente esterna ed una classe "adapter" che la implementa e che si occupa di delegare le richieste.

GreenBottle pone l'accento sulla sostenibilità ed uno degli aspetti caratteristici della piattaforma è la consegna delle bottiglie tramite veicoli elettrici. Per garantire che vengano effettuate nel modo più rispettoso possibile dell'ambiente è necessario, inoltre, che i corrieri seguano percorsi ottimali quando effettuano le loro consegne. A tal proposito, l'Adapter è utilizzato per il calcolo del percorso ottimale per i corrieri, elaborato dal modulo Pathetum esterno e visualizzato nella piattaforma web GreenBottle.

L'integrazione con il modulo di Intelligenza Artificiale viene realizzata tramite un'API che restituisce dati in formato JSON. L'Adapter converte questi dati grezzi in oggetti utilizzabili all'interno del sistema GreenBottle.



6 Migliorie future

Alcune idee utili da implementare nel futuro potrebbero essere le seguenti:

6.1 Aggiunta del Peso agli Ordini

Attualmente, il clustering si basa esclusivamente sulla distanza e la quantità. Introdurre il peso degli ordini come ulteriore parametro consentirebbe una distribuzione più equilibrata e realistica dei carichi di lavoro tra i vari cluster. Questo miglioramento porterebbe benefici concreti in termini di ottimizzazione logistica, riduzione dei costi operativi e gestione più efficace delle risorse disponibili.

6.2 Vincolo sul Numero Massimo di Elementi per Cluster

Al fine di garantire che ciascun cluster mantenga dimensioni gestibili e operative, si potrebbe introdurre un vincolo che limiti il numero massimo di ordini inclusi in ciascun gruppo. Questo permetterebbe una gestione più efficace dei cluster dal punto di vista operativo, facilitando il processo decisionale, migliorando la rapidità nella fase di smistamento e garantendo maggiore flessibilità nella gestione degli imprevisti.

6.3 Selezione Manuale degli Ordini da parte dell'Admin

Consentire all'Admin di selezionare manualmente quali ordini passare al modulo di clustering, invece di trasferirli tutti indistintamente, garantirebbe maggiore flessibilità operativa e un controllo più fine del processo. Questa possibilità aiuterebbe a ottimizzare ulteriormente le performance del clustering, gestendo eccezioni e specifiche situazioni operative in modo più efficace.

6.4 Clustering Ricorsivo

In contesti più complessi, potrebbe essere utile applicare un secondo livello di clustering all'interno dei cluster principali. Questo approccio permetterebbe una segmentazione più dettagliata, ad esempio per ottimizzare ulteriormente i percorsi all'interno di zone molto dense o per suddividere compiti tra più corrieri all'interno di un'area. Potrebbe risultare particolarmente vantaggioso per migliorare la scalabilità del sistema in scenari con grandi volumi di ordini.