POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

Clustering dei dati spazio-temporali

Relatore:

Prof.ssa Tania Cerquitelli

Correlatore:

Paolo Bethaz

Candidato:

Assunta Valentina Aprile

Anno Accademico

2020-2021

Sommario

Introduzione

L'obiettivo principale del presente lavoro di tesi è la presentazione di diverse metodologie di *clustering* per l'analisi dei dati spazio-temporali. I dati spazio-temporali sono georeferenziati e caratterizzati dall'informazione temporale circa l'istante di rilevazione o validità del dato. Nello specifico, gli attributi di un dataset spazio-temporale sono:

- dimensione spaziale, che esprime la posizione dell'oggetto sulla Terra e corrisponde tipicamente alle coordinate di latitudine e longitudine;
- dimensione temporale, che descrive l'istante di tempo di validità dell'istanza, corrispondente ad esempio all'istante in cui il dato è stato rilevato o salvato nel database;
- una o più dimensioni collegate a misure non spaziali, che caratterizzano l'istanza (ad esempio, nome, popolazione, tasso di disoccupazione di una città in un certo intervallo di tempo, ecc..)

Le proprietà geografiche e temporali sono un aspetto chiave di molti problemi di *data analysis* nel dominio del *business* in generale, della politica e dell'economia, ma anche della scienza.

Esempi di dati spazio-temporali sono: temperatura media registrata da diverse stazioni meteorologiche in un certo istante di tempo; numero di casi di malattie registrati in diverse città in un periodo di tempo, ad esempio mensile; metriche dell'inquinamento dell'aria di diverse città rilevate in un certo intervallo, ad esempio con frequenza oraria.

Gli algoritmi di *clustering* dei dati spazio-temporali sono delle tecniche non supervisionate di *data analysis* con l'obiettivo di raggruppare i dati sulla base della relativa similarità temporale e spaziale, ma non solo. È un campo di interesse relativamente nuovo del *data mining* che ha attirato sempre più attenzione soprattutto a causa della pervasività di quei dispositivi basati sulla posizione e che dunque permettono di tener traccia non solo di quest'ultima, ma anche dell'ora e di diverse proprietà dell'ambiente. Infatti, grazie ad una gamma sempre più vasta di sensori e ad un prezzo sempre più ridotto è stata rilevata una crescita esponenziale dei dati *geo-tagged* con frequenze di campionamento sempre più basse. [1]

Di seguito si descrive il dettaglio dei capitoli del presente lavoro di tesi.

Nel Capitolo 1 vengono presentati i principali algoritmi non supervisionati di *clustering*, in particolare si distinguono gli algoritmi di *clustering* partitivo da quelli di tipo gerarchico, per poi passare ad una seconda classificazione degli algoritmi di *clustering,* distinti in *model based, density based* e *distance-based.*

Nel Capitolo 2 viene fatto un focus sugli algoritmi di *clustering* utilizzati nello specifico per l'analisi di dati spazio-temporali.

Poiché le analisi successive si basano principalmente sull'applicazione di algoritmi di *clustering distance-based,* nel Capitolo 3 vengono descritte le metriche di distanza utilizzate e adattate ai dati spazio-temporali.

Nel Capitolo 4 vengono presentati i principali indici utilizzati per la valutazione della qualità dei *cluster* ottenuti, in particolare l'indice di *Silhouette* e il *Dunn Index.*

Al fine di testare la nuova metrica di distanza introdotta per i dati spazio-temporali, sono stati simulati una serie di dataset con attributi spaziali e temporali; tali dataset sono stati *clusterizzati* con gli algoritmi *K-Medoids, Agglomerative* e *ST-DBSCAN.* I principali risultati ottenuti sono stati descritti nel Capitolo 5.

Nel Capitolo 6 vengono presentati i principali risultati ottenuti invece *clusterizzando* dataset reali caratterizzati da attributi spaziali e temporali. In particolare, nel paragrafo 6.1 viene *clusterizzato* un primo dataset contenente i valori orari della temperatura atmosferica, misurati nell'arco di 5 anni e relativi a 50 città degli Stati Uniti e del Canada e 6 città di Israele; inoltre, per ogni città sono disponibili le coordinate geografiche, in termini di latitudine e longitudine.

Nel paragrafo 6.2 viene *clusterizzato* un secondo dataset che corrisponde ad una porzione di dati raccolti da *Microsoft Research Asia* durante il progetto *Geolife* [2] *,* durante il quale sono stati registrati tramite *smartphone* i dati relativi agli spostamenti di 182 utenti volontari in un periodo di 5 anni.

Nel paragrafo 6.3 viene *clusterizzato* un dataset relativo al progetto *SF-Park* di San Francisco che riporta la disponibilità di alcuni parcheggi di San Francisco misurata grazie a sensori posti sotto l'asfalto. Inoltre, il

modello di *clustering spazio-temporale* utilizzato viene proposto come tecnica di *data analysis* per la determinazione *near-real time* del prezzo dinamico dei parcheggi. Tale prezzo, infatti, può essere determinato per ogni parcheggio sulla base del *cluster* di appartenenza identificato. Inoltre, per l'identificazione di tali *cluster* si può modificare il peso delle componenti spaziali e temporali, a seconda dell'obiettivo e dei criteri che si vogliono adottare.

Infine, nel paragrafo 6.4 viene *clusterizzato* un dataset contenete le informazioni relative al servizio di *bike sharing* di alcune città della California (tra cui San Francisco, Los Angeles, Santa Clara).

1. *Metodi di* clustering

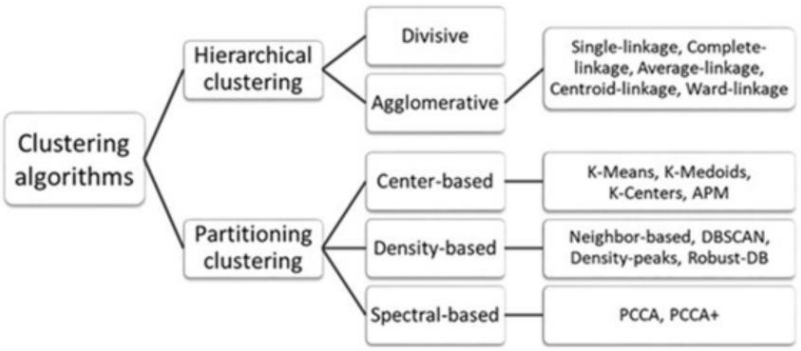
Lo *spatio-temporal clustering* è un argomento di interesse del campo del *spatio-temporal data mining* e *knowledge discovery*, utilizzato per individuare e/o validare *trends* di fenomeni geografici che si evolvono nel tempo [3]. Più in generale, il *clustering* è uno degli approcci utilizzati per la modellazione descrittiva dei *big data*; in particolare i metodi di *clustering* analizzano ed esplorano un set di dati per raggruppare gli oggetti in modo tale che vengono inseriti in gruppi (detti appunti *cluster*) che hanno caratteristiche comuni: ciò significa che un punto che appartiene ad un *cluster* ha una distanza da tutti gli altri punti dello stesso *cluster* minore rispetto alla distanza dai punti di altri *cluster* [1]. Gli aspetti minimi da definire in quanto essenziali ad ogni processo di *clustering* sono due, ovvero: la metrica di similarità (o dissimilarità) e l'algoritmo di *cluster* da utilizzare. È importate distinguere gli algoritmi di *cluster* dalle metriche di distanza; infatti ad esempio diversi algoritmi di *cluster* possono utilizzare una stessa metrica di similarità, e viceversa. [1]

Per quanto riguarda gli algoritmi di clustering di dati spazio-temporali, Agrawal et al. [4] sottolineano che questi devono soddisfare alcuni requisiti, come:

- adattabilità ad attributi spaziali, temporali e non-spaziali (questi ultimi detti in generale "metriche"); ciò implica che ci sia la possibilità di aggiungere, levare o pesare in maniera diversa i vari attributi
- scalabilità, in termini di potenza computazionale e adattabilità ad *high dimensional data* (i dati spazio-temporali sono spesso raccolti tramite sensori, il che corrisponde ad una frequenza di generazione dei dati molto elevata).

In generale, gli algoritmi di *clustering* possono essere classificati in: *model-based, distance-based, density-based* [1]. Un'ulteriore classificazione che può essere fatta a monte è quella che distingue i *partitional clustering* dai *hierarchical clustering* (Figura 1).

Figura 1 Classificazione del clustering



1.1. Partitional clustering

I *partitional clustering* permettono di ottenere gruppi di dati organizzati in partizioni, dunque in *cluster* non sovrapposti; di conseguenza ogni punto del *dataset* verrà assegnato ad un solo *cluster.* I *cluster* sono formati in modo da massimizzare la similarità *intra-cluster* e minimizzare la similarità *inter-cluster.* Formalmente, dato un *dataset* D di n istanze e k è il numero di *cluster* da formare, un algoritmo di partizionamento organizza le istanze in k partizioni k ≤ n, dove ogni partizione rappresenta un *cluster* (Figura 2).

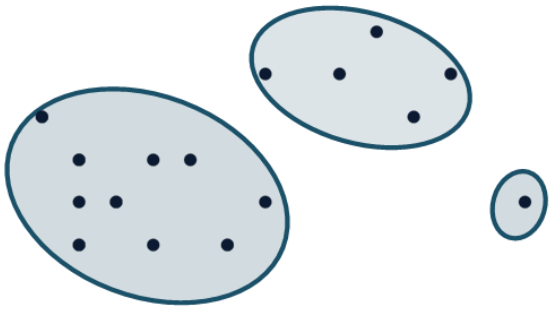


Figura 2 Esempio di partitional clustering

1.2. Hierarchical clustering

A differenza dei *partitional clustering*, gli algoritmi gerarchici creano un insieme di *cluster* nidificati: tale risultato può essere rappresentato graficamente attraverso un diagramma ad albero, chiamato dendrogramma, in cui ogni *cluster* è l'unione dei suoi sotto *cluster* (Figura 3). È un tipo di *clustering* adatto ai dati gerarchici, come le tassonomie. Il *clustering* gerarchico può essere di due tipi: *divisive clustering,* noto anche come approccio *top-down,* e *agglomerative clustering*, noto anche come approccio *bottom-up.* Nel primo caso (approccio *top-down*), tutti i punti appartengono inizialmente ad un unico *cluster*, per poi essere suddivise in *cluster* sempre più piccoli fino alla situazione diametralmente opposta in cui ogni unità appartiene ad un *cluster*. Nel secondo caso (approccio *bottom-up*) le unità sono raggruppate successivamente in *cluster* partendo dall'insieme iniziale dei dati, per cui ogni unità corrisponde ad un *cluster* distinto, fino ad arrivare ad un unico *cluster* in cui confluiscono tutti i punti.

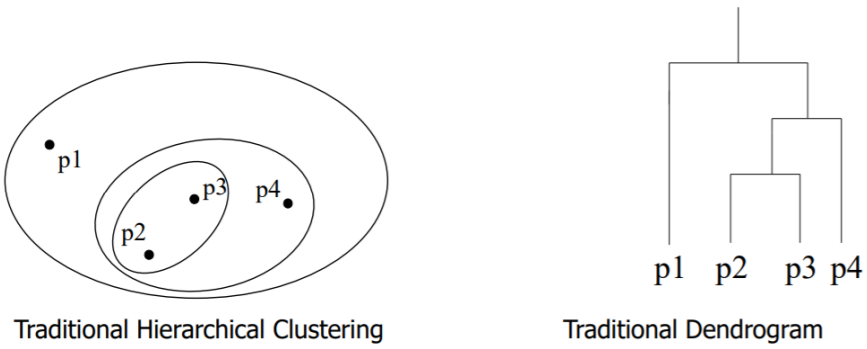


Figura 3 Esempio di Hierarchical clustering e dendogramma corrispondente

Ponendo l'attenzione *sull'agglomerative clustering,* possono essere fissati diversi criteri per decretare la fusione dei *cluster*; tra quelli più adottati in letteratura si cita:

- il *Single linkage Clustering*
- il *Complete linkage Clustering*
- l'*Average linkage Clustering.*

Di seguito sono descritte le peculiarità di ciascuno degli approcci.

Complete Linkage

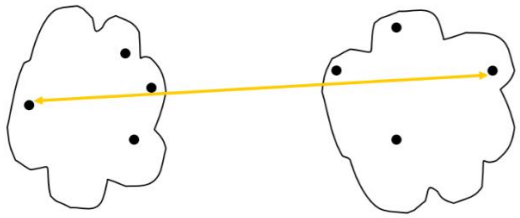


Figura 4 Complete linkage (max)

Con questo tipo di approccio la distanza (dissimilarità) tra due *cluster* C_i e C_j è definita come la distanza più grande tra tutte quelle calcolabili rispettivamente tra ciascun punto del gruppo *i* e ciascun punto del gruppo *j* (Figura 4).

Quindi, la distanza tra due gruppi diversi è calcolata:

$$D(C_i, C_j) = \max[\, d(x,y)\,]$$

$$x \in C_i, y \in C_j$$

dove x e y sono due punti appartenenti rispettivamente al *cluster* C_i e al *cluster* C_j.

Ad ogni iterazione vengono uniti i due cluster con la distanza massima più piccola. Tale criterio permette di gestire cluster di forma non sferoidale, ma è sensibile alla presenza di *outliers*.

Single linkage

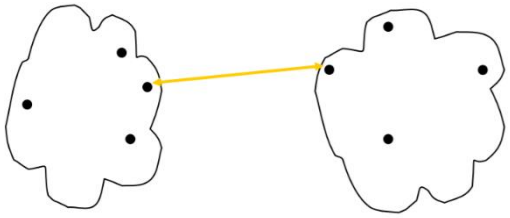


Figura 5 Single linkage (min)

Secondo questo approccio la distanza (dissimilarità) tra due *cluster* C_i e C_j è definita come la minima distanza calcolata tra ogni coppia di punti appartenenti ai due diversi *cluster* (Figura 5).

Formalmente è possibile esprimerla come:

$$D(C_i, C_j) = min[d(x,y)]$$

$$x \in C_i, y \in C_j$$

dove x e y sono due punti appartenenti rispettivamente al *cluster* C_i e al *cluster* C_j.

Ad ogni iterazione vengono uniti i due *cluster* con la distanza minima più piccola. Tale criterio è meno sensibile alla presenza di *outliers*, rispetto al criterio del complete linkage; tuttavia, tende a separare *cluster* di grandi dimensioni ed è maggiormente performante nel caso di *cluster* globulari.

Average linkage

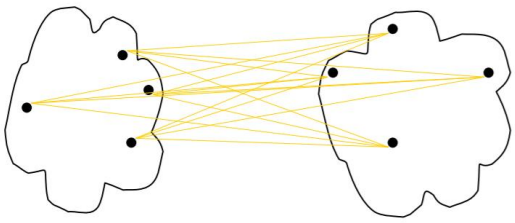


Figura 6 Average linkage

Questo approccio rappresenta un compromesso tra i due algoritmi precedenti (Figura 6). In tal caso, infatti, la distanza (dissimilarità) tra due *cluster* C_i e C_j è definita come media delle distanze tra le coppie di punti appartenenti ai due diversi cluster:

$$D(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i, y \in C_j} d(x,y)$$

$$x \in C_i, y \in C_j$$

dove x e y sono due punti appartenenti rispettivamente al *cluster* C_i e al *cluster* C_j.

Ad ogni iterazione vengono uniti i due cluster con la distanza media più piccola. Questo criterio rappresenta un compromesso tra la single linkage e il complete linkage e risulta meno sensibile alla presenza di *ouliers*; tuttavia, è indicato nel caso di cluster sferici.

La tipologia di *linkage* deve essere scelta a seconda del tipo di problema in analisi.

1.3. Model-based clustering

L'obiettivo di questi algoritmi è derivare un modello globale in grado di descrivere l'intero dataset; alcuni di questi metodi si basano sulla definizione di una densità di probabilità multivariata e permettono di assegnare ad ogni punto del dataset una probabilità di appartenenza ai diversi *cluster.* Tali algoritmi si distinguono da quelli, ad esempio, *distance-based* e *density based*, che assegnano ogni elemento del *dataset* ad un *cluster.* I parametri delle diverse distribuzioni, come media e varianza, permettono di descrivere i *cluster*; ad esempio, la stima della media della distribuzione di ogni variabile è analoga alla determinazione del centroide con l'algoritmo *K-means* [5]. Tali parametri devono essere determinati in modo tale che venga massimizzata la probabilità che il modello individuato generi il *dataset* in analisi; questo può viene fatto ad esempio con l'algoritmo EM (*Expectation-Maximization algorithm*). Nel caso specifico di *spatio-temporal clustering,* tale algoritmo è stato utilizzato da Gaffney e Smyth [6], i quali propongono un metodo di *clustering* basato su un *mixture model* per traiettorie continue, che vengono modellate con una funzione del tempo e ipotizzando che seguano una traiettoria centrale più un rumore gaussiano.

1.4. Density-based clustering – DBSCAN family

I metodi di clustering *density-based* utilizzano una soglia di densità attorno a ciascun punto per distinguere i *cluster* di dati densi da quello che può essere definito rumore. Il DBSCAN (***D**ensity-**B**ased **S**patial **C**lustering*) è uno dei primi esempi di *clustering* basato sulla densità; la soglia di densità è espressa per mezzo di due parametri:

- un raggio ε massimo attorno a ciascun punto
- un numero minimo di punti, detto ***MinPts***, entro questo intervallo.

L'algoritmo DBSCAN (Figura 7) scansiona l'intero *dataset* e classifica ogni punto ***p*** appartenente al *dataset* come:

- ***core point***, cioè un punto che appartiene ad un *cluster,* se il suo intorno di raggio ε contiene almeno un numero di punti pari a ***MinPts***
- ***border point***, cioè un punto al bordo di un *cluster,* se è raggiungibile da un *core point* ma non soddisfa la condizione di quest'ultimo, ovvero nel suo intorno di raggio ε sono presenti un numero di punti inferiori a ***MinPts***
- ***noise***, ovvero punto definitivamente al di fuori di uno qualsiasi *cluster,* considerato dunque rumore, in quanto non soddisfa né la condizione di *core point* né di *border point.*

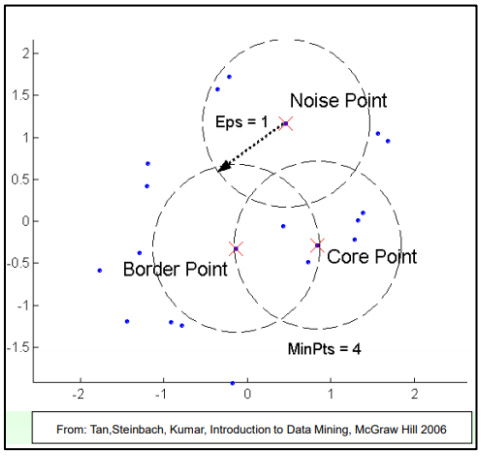


Figura 7 Algoritmo DBSCSN

Birant e Kut [7] proposero un algoritmo detto ***ST-DBSCAN*** (***S**patio **T**emporal **DBSCAN***) [8], che non è altro che un'estensione dell'algoritmo DBSCAN, ma adattato a dati spazio-temporali (tale algoritmo verrà descritto più nello specifico successivamente nel paragrafo 2.1).

Un secondo algoritmo *density-based* è detto ***OPTICS*** (***O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure*), simile al DBSCAN, in quanto procede esplorando il *dataset* ed enumerando tutti i punti ***p***, per ognuno dei quali

viene verificato se la condizione del *core point* è soddisfatta; in caso positivo, inizia ad allargare il potenziale *cluster* controllando la condizione di *core point* per tutti i vicini di p. Se l'oggetto p non è un *core point*, il processo di scansione continua con il prossimo punto non visitato. I risultati sono riepilogati in un diagramma di raggiungibilità (un esempio è riportato in Figura 8): gli oggetti sono rappresentati lungo l'asse orizzontale nell'ordine di visita e la dimensione verticale rappresenta le loro distanze di raggiungibilità. Intuitivamente, la distanza di raggiungibilità di un oggetto $\boldsymbol{p_i}$corrisponde alla distanza minima dall'insieme dei suoi predecessori $\boldsymbol{p_j}$con $0 < j < i$. Di conseguenza, un valore elevato della distanza raggiungibile approssimativamente indica una distanza elevata tra i punti: ciò indica che l'oggetto si trova in un'area poco densa. I *cluster* effettivi possono essere determinati definendo una soglia di distanza raggiungibile: gli elementi esplorati in modo consecutivo che si trovano al di sotto della soglia scelta formano un unico *cluster*. Il risultato dell'algoritmo OPTICS non è sensibile all'ordine dei punti del *dataset*. Infatti, questi vengono visitati rispettando l'ordine solo fino a quando non viene trovato un *core point*, dopodiché l'intorno del *core point* viene espanso; per questo motivo, l'ordine di visita dipende dalle distanze tra i punti. Analogamente al *ST-DBSCAN,* Agrawal et al. [4] propongono l'***ST-OPTICS,*** un'estensione dell'algoritmo *OPTICS,* adattato ai dati spazio-temporali.

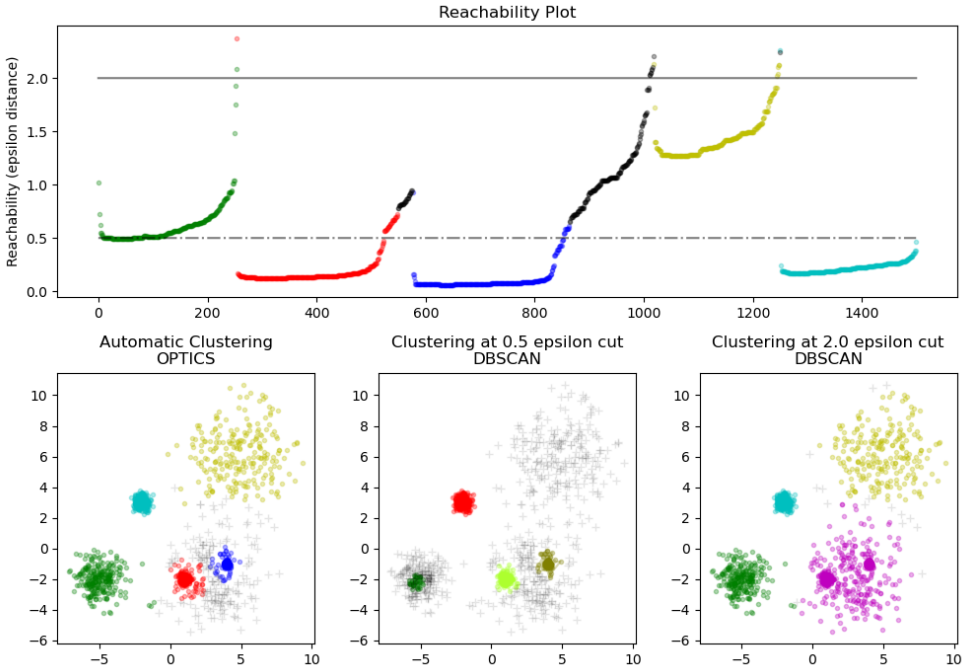


Figura 8 Confronto tra il clustering ottenuto con l'algoritmo OPTICS (sinistra) e DBSCAN con epsilon pari a 0.5 (al centro) e esilon pari a 2.0 (a destra).

1.5. Distance-based clustering (o Representative-Based Algorithms)

Gli algoritmi appartenenti a tale categoria sono basati sul concetto di distanza (o similarità) per individuare i diversi *cluster*. I punti rappresentativi di un *cluster* possono essere funzione dei *data points,* come ad esempio la media nel caso dell'algoritmo *K-Means,* oppure possono coincidere proprio con dei punti appartenenti al *dataset,* come ad esempio nel caso dell'algoritmo *K-Medoids.* Dopo aver determinato i punti rappresentativi, ogni *data point* viene assegnato al punto rappresentativo più vicino. Tipicamente, il numero *k* di *cluster* da individuare è fissato dall'utente. Considerando un dataset ***D*** contenente *n data points* $\boldsymbol{X_1, X_2, \dots, X_n}$, lo scopo è determinare ***k*** *rapresentatives* $\boldsymbol{Y_1, Y_2, \dots, Y_k}$ tali da minimizzare la funzione obiettivo seguente:

$$\boldsymbol{O} = \sum_{i=1}^{n} [\min_{j} DIST(X_i, \boldsymbol{Y_j})$$

Ciò significa che i punti rappresentativi devono essere scelti in modo da minimizzare la somma delle distanze tra tutti i punti del *dataset* e i relativi punti rappresentativi più vicini [5].

Utilizzando questo approccio, il problema del *clustering* può essere ricondotto ad un problema di identificazione di:

- tipo di algoritmo di *clustering* più adatto
- funzione di distanza, che determina quali punti sono candidati a far parte dello stesso *cluster*.

Il concetto di similarità dei punti appartenenti ad un dataset spazio-temporali può variare a seconda dello scenario applicativo considerato. In generale, in questo caso, due oggetti possono essere considerati simili se hanno seguito la stessa traiettoria spazio-temporale entro un determinato intervallo, cioè sono stati negli stessi posti allo stesso tempo. Tuttavia, la granularità dei movimenti osservati (ovvero il numero di punti spazio-temporali campionati per ciascuna traiettoria), l'incertezza sui punti misurati e, in generale, altre variazioni della disponibilità delle posizioni dei due oggetti confrontati hanno comportato la necessità di definire diverse misure di somiglianza per i *dataset* spazio-temporali. La definizione di queste misure varia spesso in base al dominio di analisi specifico. [1]

Affinché una funzione possa essere considerata una metrica di distanza tra due oggetti del dataset x_i e x_j, descritti ognuno dal proprio vettore $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ e $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$, deve soddisfare tre condizioni, ovvero:

1. non negatività: $d(x_i, x_j) \geq 0 \ \forall x_i, x_j \epsilon R^n$ e $d(x_i, x_j) = 0 \ se \ e \ solo \ se \ x_i = x_j$
2. simmetria $d(x_i, x_j) = d(x_j, x_i) \quad \forall x_i, x_j \epsilon R^n$
3. disuguaglianza triangolare: $d(x_i, x_j) \leq d(x_i, x_k) + d(x_j, x_k) \quad \forall x_i, x_j, x_k \epsilon R^n$

La funzioni di distanza utilizzata nel *clustering* può essere divisa in tre categorie generali [9]:

- distanze ***L^p-norm*** o **distanza di *Minkowski*,** definite dalla seguente equazione:

Equazione 1 Distanza di Minkowski

$$d(x_i, x_j) = \|x\|_p = \left[\sum_{r=1}^{n} (x_{ir} - x_{jr})^p\right]^{\frac{1}{p}}$$

dove: p è un parametro per cui $p \geq 1$ e $p \epsilon R$
n è il numero di dimensioni (attributi)
x_{ir} e x_{jr} sono le *k-esimi* componenti rispettivamente di x_i e x_j

In particolare, la ***2-norm*** o ***L²,*** detta '**distanza euclidea',** è la metrica che appartiene a tale famiglia più utilizzata, definita dalla seguente equazione:

Equazione 2 Distanza euclidea

$$d(x_i, x_j) = \|x\|_2 = \sqrt[2]{\sum_{r=1}^{n} (x_{ir} - x_{jr})^2}$$

La distanza euclidea tra due punti x_i e x_j rappresenta la lunghezza del segmento tra i due punti, ovvero la distanza lineare. In alcuni contesti, la distanza euclidea non è in grado di catturare le distanze effettive in un dato spazio. Un esempio noto è quello della distanza percorsa dai tassisti in un piano stradale a griglia che dovrebbe misurare la distanza non in termini di lunghezza lineare dal punto di partenza a quello di arrivo, ma in termini di distanza rettilinea (detta *taxicab metric, rectilinear distanc* o ***Manhattan distance***), che tiene conto che le strade sono ortogonali o parallele tra loro (Figura 9) [10].
Essa corrisponde alla ***1-norm*** o L^1 ed è pari alla somma dei valori assoluti delle differenze delle diverse componenti, formalmente così definita:

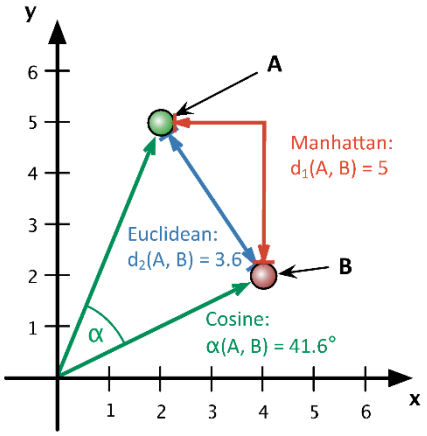


Figura 9 Confronto tra distanza euclidea (in blu) e distanza Manhattan (in rosso) tra il punto A e B

Equazione 3 Distanza di Manhattan

$$d(x_i, x_j) = \|x\|_1 = \sum_{r=1}^{n} |x_{ir} - x_{jr}|$$

- misure **statistiche**; ad esempio, il **coefficiente di Pearson** è un metodo basato sulle statistiche, così definito:

Equazione 4 Coefficiente di Pearson

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

dove:

- X e Y sono due variabili casuali
- σ_{XY} è la covarianza tra X e Y
- σ_X e σ_Y sono le due deviazioni standard

- misure **elastiche**: (es ***Dynamic time warping distance - DTW:*** utilizzata per determinare la corrispondenza ottimale tra due serie storiche aumentando o comprimendo i loro segmenti).

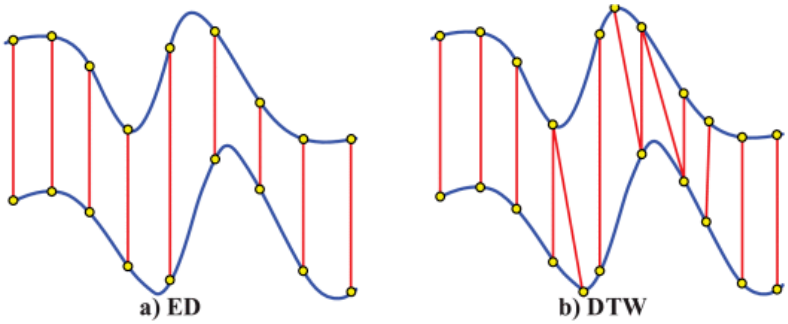


Figura 10 (a) Euclidean Distance (b) Dynamic Time Warping

Poichè la trattazione seguente riguarda soprattutto l'analisi dei dati spazio-temporali, si ritiene necessario descrivere nello specifico la differenza tra la distanza euclidea e la *dynamic time warping (DTW*), illustrate in Figura 10, in quanto sono le metriche di distanza più utilizzate per le *time series*. Per questa tipologia di dato, la distanza euclidea considera come dimensioni i valori degli attributi nei diversi istanti di tempo¸ quindi tale metrica può essere applicata quando le *timeseries* hanno stessa dimensione e c'è una corrispondenza uno a uno tra i valori di due *timeseries* in uno stesso istante di tempo. Inoltre, non è la metrica più adatta per *timeseries* non sincrone. [5] Il *Dynamic Time Warping* (DTW), invece, è una tecnica che consente di allineare due sequenze, anche con numero di osservazioni diverse, calcolandone la distanza. Berndt e Clifford [11] hanno analizzato l'utilizzo del DTW in ambito del riconoscimento vocale: infatti, il problema dell'individuazione

delle parole all'interno di un discorso presenta aspetti simili all'individuazione di pattern di *timeseries*. In generale, il DTW consente di trovare un allineamento ottimale tra due segnali attraverso la loro distorsione (*warp*) non lineare rispetto alla variabile tempo. L'algoritmo DTW risulta essere dunque un buon approccio per risolvere il problema del disallineamento delle *timeseries.* In particolare, date due serie storiche $X = (x_1, x_2, \dots, x_n)$ e $Y = (y_1, y_2, \dots y_m)$, è possibile definire una matrice di dimensioni $n\ x\ m$ in cui ciascun elemento (i, j) corrisponde all'allineamento tra il generico elemento x_i e il generico elemento y_j.

		x[t] n=6					
		1	1	2	3	2	0
y[t] m=7	1	0	0	1	2	1	-1
	2	-1	-1	0	1	0	-2
	3	-2	-2	-1	0	-1	-3
	2	-1	-1	0	1	0	-2
	1	0	0	1	2	1	-1
	1	0	0	1	2	1	-1
	0	1	1	2	3	2	0

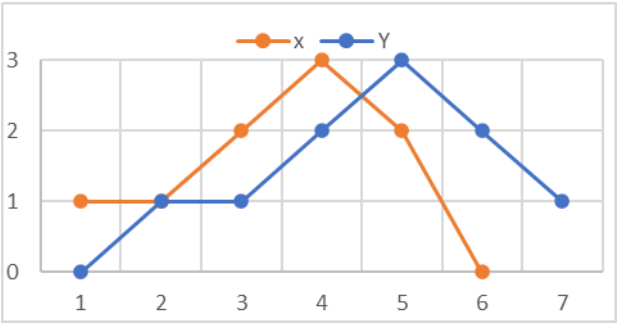


Figura 11 Esempi di due timeseries con matrice di allineamento con le relative distanze d(xi,yj)=| xi - yj |

In Figura 11 si può osservare un esempio di due *timeseries* con numero di osservazioni diverse, $x[t] = (1,1,2,3,2,0)$ e $y[t] = (0,1,1,2,3,2,1)$ e la corrispondente matrice: ogni riga della griglia corrisponde agli istanti di tempo relativi alla serie $x[t]$ e ogni colonna corrisponda agli istanti di tempo relativi alla serie $y[t]$. Ciascun elemento della matrice contiene la distanza $d(x_i, y_j)$ tra due generici valori x_i e y_j, con $i = 1,2, \dots, n$ e $j = 1,2, \dots m$ (in tal caso *n=6, m=7)* [9]. Per calcolare questi valori è necessario definire una misura di distanza tra gli elementi delle serie, le misure proposte da Berndt e Clifford [11] sono:

- valore assoluto della differenza: $d(x_i, y_j) = |x_i - y_j|$
- quadrato della differenza: $d(x_i, y_j) = (x_i - y_j)^2$

Una volta stabilita l'unità di misura, si può definire il problema del *dynamic time warping* come un problema di minimizzazione della distanza cumulata di ogni percorso individuato, detto *warping path* $W = w_1, w_2, \dots, w_k, \dots w_h$.

Equazione 5 Dynamic Time Warping

$$DTW(X,Y) = \min_{w}[\sum_{k=1}^{p} d(w_k)]$$

dove:

d è la metrica di distanza scelta tra gli elementi di due *timeseries*

w_k è un punto (x_i, y_j) della matrice di distanza e rappresenta il *k-esimo* elemento del *warping path W*

Nel caso in cui le *timeseries* abbiano le stesse dimensioni il percorso ottimo *W* coincide con la diagonale. Poiché l'algoritmo confronta ogni punto di una serie con tutti i punti dell'altra, il numero di possibili *W* è molto alto; si osserva però che la sequenza di *W* con valori minimi tipicamente è intorno alla diagonale. Per ridurre il campo di ricerca dei *warping path* vengono imposte le seguenti restrizioni:

1) *monotonia rispetto al tempo:* i punti devono essere ordinati in modo monotono rispetto al tempo
$$i_{k-1} \leq i_k \quad \text{e} \quad j_{k-1} \leq j_k$$
2) *continuità:*
$$i_{k-1} - i_k \leq 1 \quad \text{e} \quad j_{k-1} - j_k \leq 1$$
3) *warping window:* gli elementi della matrice che possono identificare il *warping path* rientrano in una 'finestra' con ampiezza ω, garantendo che tale percorso non si allontana eccessivamente dalla diagonale:
$$|i_k - j_k| \leq \omega, \text{ con } \omega > 0$$

4) *slope constraint:* il campo di ricerca dei percorsi consentiti può essere limitato vincolando la pendenza, in modo da evitare movimenti eccessivamente ampi in una sola direzione.

5) *condizioni al contorno,* ovvero condizioni generali che limitano il campo di ricerca, garantendo che il *warping path* segua una direzione diagonale. Tipicamente si impone che il primo elemento di *W* sia il primo elemento della diagonale in basso a sinistra e l'ultimo quello in alto a destra:

$$i_1 = 1, j_1 = 1\,;\;\; i_n = n\,, j_m = m$$

L'algoritmo per il calcolo della distanza cumulata può essere riscritto secondo la seguente relazione iterativa:

Equazione 6 Distanza cumulata - Dynamic Time Warping

$$\gamma(i,j) = d(i,j) + \min[\gamma(i-1,j), \gamma(i-1,j-1), \gamma(i,j-1)\,]$$

L'Equazione 6 riporta la distanza cumulata $\gamma(i,j)$, ottenuta sommando alla distanza $d(i,j)$ tra gli elementi allineati la distanza minima degli elementi vicini, ovvero $i-1,j; i-1,j-1; i,j-1$. A ogni $w_k$del *warping path* ottimale W, corrisponde una coppia di osservazioni delle serie che consentono di ridefinire quest'ultime in un nuovo spazio temporale in cui risultano essere allineate e con lo stesso numero di osservazioni. In questo esempio le due sequenze risultanti sono $x'[t] = (1,1,2,3,2,0)$ e $y'[t] = (0,1,1,2,3,2,1)$, come mostrato dalla Figura 12. La distanza tra le due serie temporali è pari al valore assoluto della differenza, che coincide con l'area compresa tra le due curve e in questo caso è pari a 2 [13].

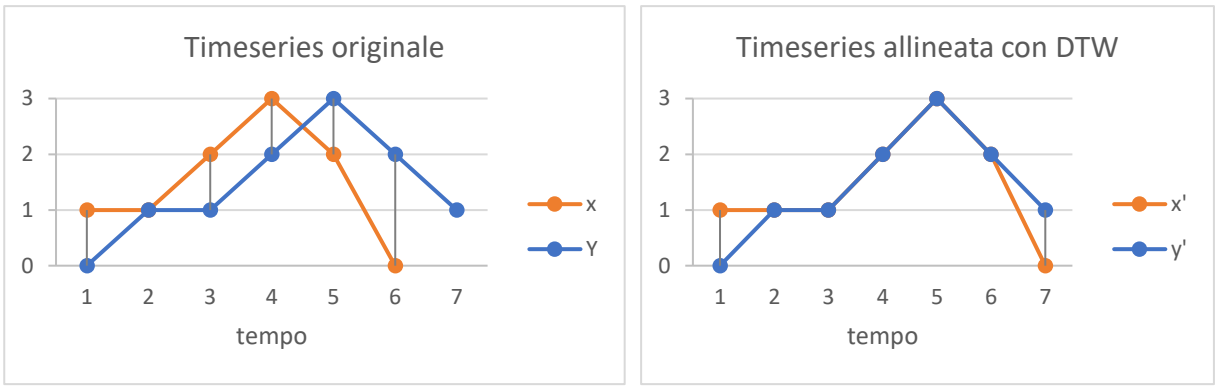


Figura 12 Timeseries originale (sinistra) e timeseries allineata mediante algoritmo DTW (destra)

2. Clustering dei dati spazio-temporali: focus sulla letteratura

Lamb et al. [14] individuano almeno due approcci che permettono di identificare gruppi di dati, adatti anche per quelli spazio-temporali, classificabili come:

- esplorativi, ovvero gli algoritmi di *machine learning* non supervisionati (tra cui il *clustering*) che hanno l'obiettivo di identificare gruppi di dati sulla base di metriche di similarità/dissimilarità.
- confermativo, ovvero le tecniche statistiche tradizionali che confermano ipotesi sviluppate a priori.

In particolare, metodi esplorativi che hanno l'obiettivo di identificare cluster di dati spazio-temporali rientrano nel dominio di *geographic knowledge discovery (GKD).*

Jeremy Mennis e Diansheng Guo [15] individuano almeno tre problemi che emergono quando vengono applicate tecniche di analisi tradizionali ai dati spaziali, ovvero: (i) se il modello ipotizzato (es: regressione lineare) non è appropriato per il fenomeno in esame, l'analisi può al massimo indicare che i dati non mostrano relazioni interessanti, ma non vengono suggerite alternative migliori; (ii) incapacità di elaborare grandi volumi di dati; (iii) non sono adatti a gestire "nuovi tipi di dati", come traiettorie di oggetti in movimento. In particolare, se le traiettorie tracciano gli spostamenti umani, questi ultimi sono spesso vincolati alla rete di trasporto esistente (es. sistema stradario); la distanza tra due punti non è quella lineare, individuata dalla distanza euclidea, ma corrisponde alla lunghezza fisica, ad esempio, delle strade da percorrere. In un contesto del genere, è stato dimostrato che il *clustering* può essere falsato, se si considera come metrica di distanza quella euclidea, la quale sottostima la distanza di punti che in realtà sono vincolati ad una rete. Per rispondere a tale problema, David S. Lamb, Joni Downs e Steven Reader [14] suggeriscono di utilizzare la *network distance.*

In generale, il processo di *clustering* dipende fortemente dalle caratteristiche specifiche dei dati considerati. In particolare, tali differenze si enfatizzano nel contesto dei dati spazio-temporali, i quali possono essere raccolti e caratterizzati in diversi modi. Dunque, è utile definire una tassonomia [1] dei dati spazio-temporali, al fine di proporre per ogni categoria degli esempi di *clustering* utilizzati in letteratura. Tale classificazione è riassunta graficamente in Figura 13.

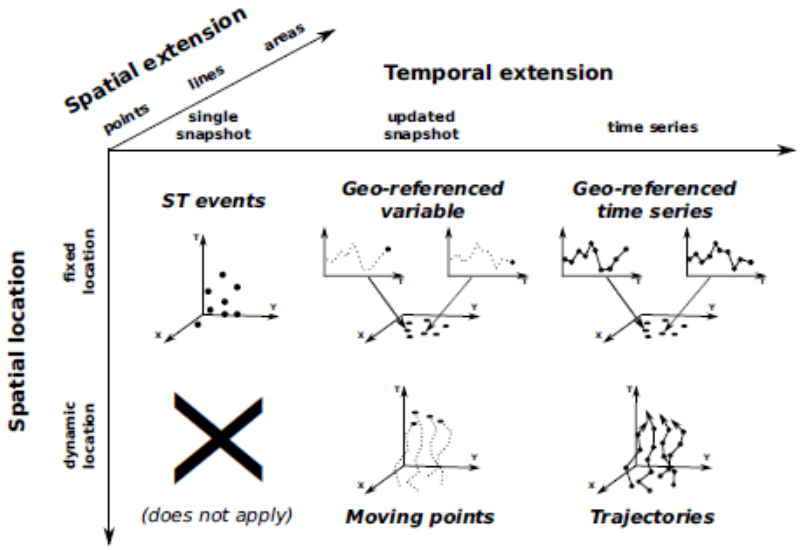


Figura 13 Tassonomia dei dati spazio-temporali

I dati spazio-temporali sono caratterizzati da:

- una dimensione temporale, che indica la data e/o l'ora in cui l'evoluzione dell'"oggetto" è catturata; in tal caso si distinguono:
  - *snapshot* statico: l'evento è caratterizzato da un singolo dato relativo ad un istante di tempo
  - *updated snapshot*: l'evento è caratterizzato da cambiamenti di stato, quindi sono disponibili *snapshot* aggiornati, ma non l'intero andamento storico
  - *time*series: l'evento è caratterizzato dall'andamento storico dello stato che evolve nel tempo

- una dimensione spaziale, che indica la posizione associata all'oggetto, distinguendo:
  - *posizione fissa (fixed location),* se l'oggetto non può cambiare posizione (ad esempio le informazioni sono raccolte da sensori in una data posizione)
  - *posizione dinamica (dynamic location), se* l'oggetto si può muovere nel tempo

Oltre a queste due dimensioni, i dati spazio-temporali possono essere caratterizzati da una terza, che è legata ad esempio a proprietà dell'oggetto valide in un certo tempo e in un certo spazio (ad esempio la temperatura misurata da un sensore) oppure può essere legata all'estensione spaziale dell'oggetto (ad esempio lunghezza della traiettoria o estensione dell'area).

2.1. ST-Events

Si definisce *ST-Event* un vettore che descrive un evento associato ad un *timestamp* e ad una posizione fissa, dunque non è previsto nessun movimento o evoluzione nel tempo. Lo scopo del *cluster* è raggruppare eventi che mostrano similarità di tempo e spazio; nel caso specifico di *ST-event,* Kulldorff [16] propone di individuare tali *cluster* considerando cilindri spazio-temporali (Figura 14), cioè regioni circolari (che individua la base del cilindro) considerate entro un lasso di tempo (che individua l'altezza del cilindro), in cui la densità di eventi dello stesso tipo è maggiore di quella esterna, rappresentando essenzialmente le aree in cui gli eventi si sono verificati in modo coerente in un intervallo di tempo significativo. In alcune applicazioni, come l'epidemiologia, è prevista una modifica delle dimensioni e della posizione di tali aree, pertanto sono state proposte estensioni delle di tali algoritmi, considerando anche forme diverse dai semplici cilindri.

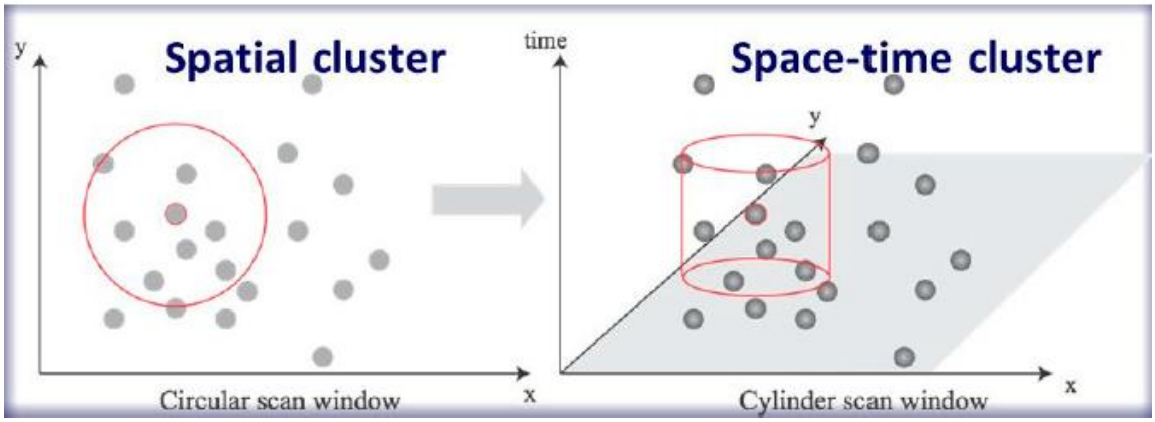
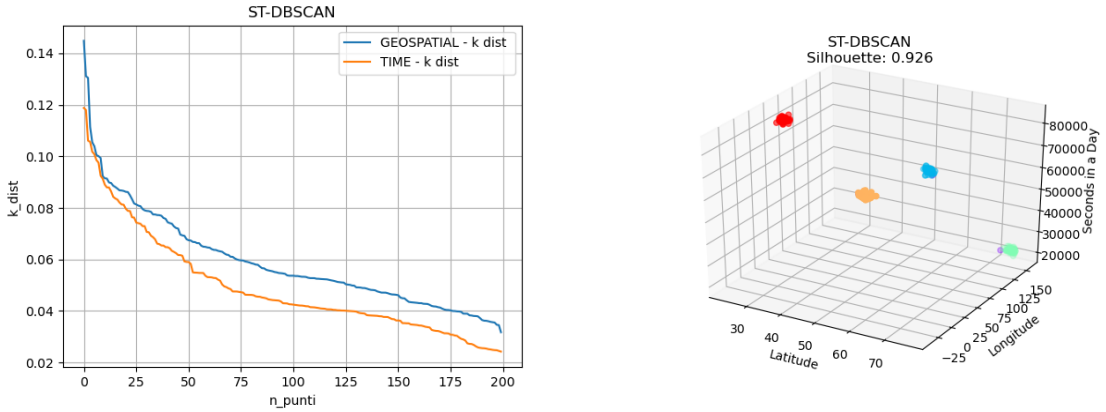


Figura 14 Rappresentazione grafica del principio alla base del cylinder scan window per il clustering spatio-temporale

Ad esempio, Iyengar [17], accanto alle forme cilindriche, introduce quelle piramidali, che rappresentano una piccola regione, ovvero il vertice della piramide, corrispondente ad esempio all'origine di un'epidemia; tale regione circoscritta cresce nel tempo, che corrisponde alla sezione trasversale della piramide, ad esempio il progressivo scoppio, fino a raggiungere la sua massima espansione, corrispondente alla base della piramide. Accanto a tali algoritmi, definibili *scan window,* Birant e Kut [8] proposero un algoritmo detto **ST-DBSCAN** (***Spatio Temporal DBSCAN***), un'estensione dell'algoritmo DBSCAN, i quale viene adattato a dati spazio-temporali; infatti, separa le informazioni "spaziali" da quelle "non-spaziali", individuando due metriche di distanza separate. ST-DBSCAN rispetto al DBSCAN, introduce un secondo parametro del raggio di prossimità, ovvero quello temporale ε_2, oltre a quello spaziale ε_1. Pertanto, un punto generico *p* appartenente al dataset è considerato *core* quando il numero di punti nell'intorno è maggiore o uguale alla soglia *MinPts* di entrambe le soglie, cioè spaziali e temporali. Il grafico k-dist, proposto da Ester [18] come metodo euristico per la determinazione dei parametri di input può essere utilizzato per determinare sia il raggio minimo temporale che spaziale. In particolare, il primo step è tracciare il grafico *k-dist* usando rispettivamente la dimensione spaziale e temporale (o in generale, le dimensioni spaziali e quelle non spaziali). Tramite il grafico, l'analista potrebbe dedurre le soglie adeguate per le lunghezze spaziali e temporali dei cluster. Nel secondo passaggio, la lunghezza viene fissata come parametro dell'algoritmo e vengono estratti i cluster. La Figura 15 mostra un esempio del *k-dist plot,* separatamente per i due tipi di attributi e il relativo *cluster* ottenuto utilizzando la libreria *st-dbscan* di *Python* [19].

Figura 15 Esempio di k-dist plot per ST-DBSCAN (sinistra) e relativo cluster ottenuto (sinistra)



Accanto all'applicazione dell'algoritmo ST-DBSCAN, si citano alcune metodologie utilizzate in letteratura per analizzare dati spazio-temporali. Ad esempio, Pöelitz et al [20] usano l'algoritmo DBSCAN tenendo in considerazione solo gli attributi che indicano le coordinate spaziali; successivamente, ad ogni cluster spaziale ottenuto applicano ulteriormente l'algoritmo DBSCAN, considerando l'attributo temporale; secondo questo metodo, i risultati del secondo cluster sono influenzati dal primo. Tale approccio viene anche seguito da Pölitz e Andrienko [21] per effettuare un cluster delle traiettorie.

2.2. *Variabili geo-referenziate*

Una variabile geo-referenziata fa riferimento all'evoluzione nel tempo di alcuni fenomeni in una posizione fissata; in particolare, secondo la classificazione di Kisilevich et at [1], introdotta precedentemente, per le variabili geo-referenziate sono disponibili solo valori più recenti e non l'intera *timeseris*. In questo caso, l'attività di *clustering* può essere considerata molto simile al caso di *ST-Events* discusso precedentemente, con l'eccezione che le loro caratteristiche non spaziali non sono costanti. Un problema tipico in questo contesto è calcolare in modo efficiente un cluster che (i) tiene conto sia dello spazio e caratteristiche non spaziali e (ii) sfrutta i cluster trovati nel *timestamp* precedente, quindi cercando di rilevare le modifiche rilevanti nei dati e aggiornare in modo incrementale i *cluster*, anziché elaborarli da zero. Tale tipologia di *dataset* non è stata trattata nel presente lavoro di tesi.

2.3. *Moving object*

I *moving objects* sono caratterizzati da una posizione spaziale e temporale dell'oggetto che cambiano nel tempo. Nel caso più semplice, le informazioni disponibili su tali oggetti consistono nella loro posizione più recente mentre non viene conservata alcuna traccia delle posizioni passate, come ad esempio accada nel contesto del monitoraggio in *real time* dei veicoli per applicazioni di sicurezza nella guida autonoma. Come nel caso di variabili georeferenziate, in questo contesto il tipico problema di *clustering* è mantenere un insieme aggiornato di *cluster* attraverso l'aggiornamento incrementale dai risultati precedenti, cercando di rilevare le recenti modifiche ai dati (in particolare, i loro recenti movimenti) che erano significativi o che probabilmente saranno seguiti da grandi cambiamenti nel prossimo futuro (ad esempio a causa di un cambio di direzione dell'oggetto). Gli algoritmi di *clustering* adottati per la tipologia di dato *moving object* sono più complessi e considerano ad esempio altre informazioni, come la velocità o l'accelerazione dello spostamento; non sono stati trattati nel presente lavoro di tesi.

2.4. *Geo-referenced time series*

Nel caso più generale, è possibile archiviare l'intera storia dell'oggetto in evoluzione (*timeseries*), individuando quindi una *timeseries* georeferenziata. In questo caso, raggruppare un insieme di oggetti richiede di confrontare il modo in cui si evolvono le loro serie storiche e di metterle in relazione con la posizione nello spazio corrispondente. Quindi il problema principale consiste nel rilevare oggetti che sono vicini nello spazio e mostrano una similarità temporale.

Ad esempio, Izakian et al [9] analizzano *georeferenced timeseries,* poiché prendono in considerazione un dataset che contiene la serie storica di temperatura, umidità, precipitazioni misurati nelle località agricole dell'Alberta, in Canada, misurate da sensori (con posizioni fisse), durante un periodo di tempo. In questo caso è stato applicato l'algoritmo **Fuzzy C-means** per *clusterizzare* time series delle condizioni climatiche, considerando come metrica di distanza una distanza Euclidea "modificata", ovvero che considerasse in maniera distinta sia la variabile temporale che quella spaziale.

Un altro esempio di *georeferenced time series* è costituito dalle misure sulla qualità dell'aria delle città della Cina, fornite da *China National Environmental Monitoring Center* (CNEMC) è rese pubbliche ogni ora. In particolare, Ziyue Chen et al [22] per la loro analisi utilizzano un *dataset* contenente misure di PM2.5, relative a 363 città della Cina dal 1 Marzo 2015 al 28 Febbraio 2018. Per l'analisi di tali dati geo-referenziati è stato usato il *repeated-bisection clustering* considerando solo le variabili temporali, per cui la metrica di distanza - similarità selezionata è la *cosine,* definita nel modo seguente:

Equazione 7 Cosine similarity

$$\cos(d_i, d_j) = \frac{d_i^T * d_j}{|d_i * d_j|}$$

L'indice di qualità del clustering selezionato è così definito:

$$I_2 = \sum_{i=1}^{k} \sqrt{\sum_{d_i, d_j \in S_i} \cos(d_i, d_j)}$$

dove:

- k=numero di cluster
- S_i è l'i-esimo cluster
- d_i e d_j sono due oggetti appartenenti al cluster S_i .

Dal momento che l'informazione geografica non è stata utilizzata per il clustering è stato successivamente utilizzato l'indice $\boldsymbol{q}$ per misurare la distribuzione delle città tra i vari cluster temporali, definito come segue:

$$q_t = 1 - \frac{1}{N\delta_t^2} \sum_{h=1}^{L} N_h \delta_{ht}^2$$

dove:

- q_t è l'indice del giorno t
- N indica il numero di città totali
- L il numero di cluster
- h è l'ID di ogni cluster
- N_h è il numero di città del cluster h
- δ_t^2 è la varianza totale di $PM_{2.5}$ del giorno t
- δ_{ht}^2 è la varianza totale di $PM_{2.5}$ del girono t considerando solo le città nel cluster h.

L'indice q_t appartiene all'intervallo $[0,1]$ ed è uguale a 1 se la concentrazione di PM2.5 di città nello stesso *cluster* nel giorno *t* è la stessa mentre è completamente diversa tra città appartenenti a diversi *cluster,* e viceversa. Dunque, un q_t elevato è indice di buona qualità del *cluster* individuato.

2.5. Traiettorie

Quando l'intera cronologia di un oggetto in movimento viene archiviata ed è disponibile per l'analisi, la sequenza delle posizioni spaziali in cui è stato l'oggetto, insieme ai relativi *timestamp*, formano quella che viene chiamata una traiettoria. Le traiettorie descrivono il movimento di oggetti, e quindi il *clustering* può essere utilizzato per rilevare gruppi di oggetti che si sono comportati in modo simile, ad esempio seguendo percorsi simili (anche in momenti diversi), o muovendosi coerentemente insieme (ovvero, avendo posizioni prossime per lunghi intervalli di tempo) o condividendo altre proprietà del movimento (come la velocità delle traiettorie).

In letteratura, per il *clustering* delle traiettorie sono state spesso utilizzate combinazioni di algoritmi o metodologie sviluppate *ad hoc* per sfruttare e mettere in risalto le informazioni derivanti dalle dipendenze spazio-temporali. Sono ad esempio state sviluppate tecniche di *microclustering,* come vengono definite da Kisilevich et al [1]. Hwang et [23] propongono un approccio dove le traiettorie sono rappresentate come segmenti con intervalli di tempo mancanti. Il metodo proposto ha l'obiettivo di determinare il '*close time interval'*, cioè l'intervallo di tempo massimo in cui tutte le coppie di traiettorie sono l'una vicino all'altra. La similarità delle traiettorie si basa su la quantità di tempo in cui queste sono vicine.

Oppure, Nanni e Pedreschi [24] propongono un nuovo approccio chiamato "***temporal focusing"*** - "focalizzazione temporale" per sfruttare meglio l'aspetto temporale e migliorare la qualità del raggruppamento delle traiettorie; infatti, due traiettorie possono essere molto diverse se si considera l'intero intervallo di tempo, tuttavia, se si considera solo un piccolo sotto intervallo queste traiettorie possono mostrare delle similarità. Quindi è fondamentale per l'algoritmo poter lavorare su diverse granularità spaziali e temporali. L'idea generale dell'approccio *temporal focusing* è di raggruppare le traiettorie usando tutti i possibili intervalli temporali (*time windows*), valutare i risultati e trovare il *clustering* migliore.

Come menzionato dagli autori, di solito alcune parti delle traiettorie sono più importanti di altre. Ad esempio, nelle ore di punta ci si può aspettare che molte persone si trasferiscano da casa al luogo di lavoro e viceversa, formando schemi di movimento che possono essere raggruppati insieme.

L'analisi delle traiettorie può essere finalizzata anche a identificare *important places* [1]. Nell'analisi di Kang et al [25], viene proposto un *clustering* incrementale per l'identificazione di luoghi importanti in un'unica traiettoria. L'algoritmo si basa sulla ricerca di "luoghi importanti", ovvero luoghi per cui sono state identificate molte misure. Due parametri determinano la definizione del *cluster*: la distanza tra posizioni e il tempo trascorso in un *cluster*. L'idea di base è la seguente: ogni nuova misurazione della posizione fornita da un dispositivo a cui è associato una posizione viene confrontato con la posizione precedente. Se la distanza tra la posizione attuale e quella precedente è inferiore a una certa soglia fissata, la nuova posizione viene aggiunta al cluster creato in precedenza. Altrimenti, si identifica un nuovo cluster candidato con la nuova posizione. Tale cluster diventa un cluster di luoghi importanti se la differenza di tempo tra il primo punto (da un punto di vista temporale) appartenente al cluster e l'ultimo punto è maggiore di una certa soglia.

Nella trattazione successiva verranno presentati metodi per il *clustering* degli *ST-Events* e delle *georeferenced timeseries*. Inoltre, verrà anche considerato un tipo di *ST-Events* caratterizzato da due posizioni, una di partenza e una di arrivo: è il caso, ad esempio, dei dati relativi ai percorsi dei servizi di *bike sharing* oppure a quelli relativi al sistema di *parking* in cui è nota la disponibilità di posti per il parcheggio lungo una strada, la quale è individuata da un punto di partenza e un finale.

2.6. *Applicazioni del clustering spazio-temporale*

Al fine di analizzare la letteratura relativa al *clustering* spazio-temporale, tali dati sono divisi in tre categorie principali rispetto alle modalità di raccolta [20], ovvero:

- **dati di movimento**: sono spesso ottenuti da dispositivi basati sulla posizione come il GPS e contengono generalmente ID di un oggetto, le sue coordinate e il *timestamp*
- **dati di reti cellulari**: sono ottenuti da operatori telefonici
- **dati ambientali*:*** generalmente ottenuti da reti di sensori.

La specificità delle proprietà di questi dati richiede approcci diversi per l'analisi e finalità differenti. Ad esempio, i dati di movimento potrebbero essere utilizzati per analizzare i movimenti degli animali e i loro comportamenti nel tempo, oppure per analizzare la mobilità delle persone e la localizzazione di gruppi di persone in luoghi considerabili di interesse. Le telefonate che le persone fanno in una città possono essere utilizzate nell'analisi dell'attività urbana, utili per le autorità locali, i fornitori di servizi, ecc.

I dati ambientali vengono analizzati utilizzando informazioni su luoghi e orari specifici, di grande importanza ad esempio per ecologi e geografi.

La Tabella 1 riassume le categorie di dati spazio-temporali con il relativo problema di analisi, esempi di applicazioni, i principali metodi di base utilizzati in letteratura e la relativa letteratura di riferimento.

Tabella 1 Applicazioni del clustering spatio-temporale. FONTE: Spatio-temporal clustering. Slava Kisilevich, Florian Mansmann, Mirco Nanni, Salvatore Rinzivillo

Category	Problem	Application	Method based on	Selected literature
Movement	Trajectory clustering	Cars, evacuation traces, landings and interdictions of migrant boats	OPTICS	(Nanni and Pedreschi(2006)) (Rinzivillo et al(2008)Rinzivillo, Pedreschi, Nanni, Giannotti, Andrienko, and Andrienko) (Andrienko and Andrienko(2008)) (Andrienko and Andrienko(2009)) (Andrienko et al(2009)Andrienko, Andrienko, Rinzivillo, Nanni, Pedreschi, and Giannotti)
	Trajectory aggregation			
	Trajectory generalization			
	Moving clusters	Migrating animals, flocks, convoys of vehicles	DBSCAN	(Kalnis et al(2005)Kalnis, Mamoulis, and Bakiras)
			Gridding, DBSCAN	(Jeung et al(2008)Jeung, Yiu, Zhou, Jensen, and Shen)
			DBSCAN	(Vieira et al(2009)Vieira, Bakalov, and Tsotras)
	Extracting important places from trajectories	People's trajectories	DBSCAN,	(Palma et al(2008)Palma, Bogorny, Kuijpers, and Alvares)
			Incremental clustering	(Kang et al(2004)Kang, Welbourne, Stewart, and Borriello)
	Trajectory patterns	Fleet of trucks	Density of spatial regions	(Giannotti et al(2007)Giannotti, Nanni, Pinelli, and Pedreschi)
		Synthetic data	BIRCH	(Kang and Yong(2009))
Cellular networks	Urban activity	Phone calls	k-means	(Reades et al(2007)Reades, Calabrese, Sevtsuk, and Ratti)
Environmental	Oceanography	Seawater distribution	DBSCAN	(Birant and Kut(2006), Birant and Kut(2007))
	Seismology	Seismic activity	Gridding, DBSCAN	(Wang et al(2006)Wang, Wang, and Li)

3. Metriche di distanza adattate ai dati spazio-temporali

Al fine di testare la metrica introdotta per il *clustering* spazio-temporale, di seguito descritta, è stata seguita l'analisi condotta da Jeremy Mennis e Diansheng Guo [15], i quali propongono un metodo generale che parte dalla simulazione di *dataset* spazio-temporali, i quali vengono prima clusterizzati al fine di verificare l'influenza dei vari parametri della metrica di distanza; una volta fissati tali parametri, il modello di *clustering* viene applicato al *traking* degli animali, con lo scopo di identificare *hotspots*, ovvero luoghi particolarmente importanti per le attività di questi ultimi.

Riprendendo la tassonomia dei dati spazio-temporali, Jeremy Mennis e Diansheng Guo [15] considerano *spatio-temporal events,* in quanto i dataset simulati sono costituiti da record caratterizzati da tre variabili, di cui due spaziali (latitudine e longitudine) e una temporale, simulata in termini di secondi dalla mezzanotte (quindi la variabile temporale t è compresa in un intervallo tra 0 s (corrispondenti alle ore 00:00) e 86 400 s (corrispondenti alle ore 00:00 del giorno successivo).

In particolare, riprendendo l'analisi di Jeremy Mennis e Diansheng Guo [15] , è stato adottato un algoritmo gerarchico che utilizza come metrica di distanza una combinazione lineare delle distanze spaziali, temporali e degli altri attributi, opportunamente pesate.

È stato selezionato l'algoritmo gerarchico per due motivi, ovvero:

- non è necessario selezionare apriori il numero di *cluster* (come nel caso del *K-Means*)
- permette di individuare e visualizzare con il dendogramma i diversi cluster gerarchici, corrispondenti a scale temporali e spaziali a diversa granularità

Per quanto riguarda la metrica di distanza, si riporta l'Equazione 8Equazione 9 generale, introdotta per il *clustering* dei dati spazio-temporali.

$$\boldsymbol{D(x_i, x_j)} = \frac{\mathbf{1}}{\sum_{i=1}^{n} \boldsymbol{w_i}} \left(\boldsymbol{w_1} * \frac{\boldsymbol{D_S(s_i, s_j)}}{\boldsymbol{S_S}} + \boldsymbol{w_2} * \frac{\boldsymbol{D_T(t_i, t_j)}}{\boldsymbol{S_T}} + \sum_{a=3}^{q} \boldsymbol{w_a} * \frac{\boldsymbol{D_a(a_i, a_j)}}{\boldsymbol{S_a}} \right)$$

Equazione 8 Funzione generale della distanza spazio-temporale

Di seguito si descrivono le diverse metriche $\boldsymbol{D_S}, \boldsymbol{D_T}$ e $\boldsymbol{D_a}$ che possono essere adottate.

Il problema di applicare algoritmi di cluster ai dati spazio-temporali è una gestione efficace dei dati in modo da preservare e sfruttare le dipendenze spazio-temporali. Per questo motivo è necessario adattare il concetto di distanza, qualora venissero applicati i classici algoritmi di clustering *distance-based* (es. *K-Means*).

Hesam Izakian et ali [26] applicano l'algoritmo di clustering *fuzzy C-Means,* modificando la funzione di distanza Euclidea, in modo da adattarla ai dati spazio-temporali e definendola come somma di due componenti, una temporale e l'altra spaziale. Gli *n* dati spazio-temporali sono rappresentati da *n* vettori $\bar{\boldsymbol{x}}_{\boldsymbol{\iota}} = [\boldsymbol{x_1}, \boldsymbol{x_2}, .., \boldsymbol{x_n}]$ e ogni oggetto $\bar{\boldsymbol{x}}_{\boldsymbol{\iota}}$ può essere espresso come una concatenazione di una parte spaziale $\bar{\boldsymbol{x}}_{\boldsymbol{i}}(s)$ e una temporale $\bar{\boldsymbol{x}}_{\boldsymbol{\iota}}(t)$, e quindi:

$$\bar{\boldsymbol{x}}_{\boldsymbol{\iota}} = [\boldsymbol{x}_i(s) | \boldsymbol{x}_i(t)]^T = \left[x_{i1}(s), x_{i2}(s), .., x_{ir}(s) \middle| x_{i1}(t), x_{i2}(t), \dots, x_{1q}(t)\right]^T.$$

Utilizzare la distanza Euclidea, che esprime la distanza planare tra due punti come un segmento lineare, può portare a false conclusioni a cui seguono distorsioni nel *clustering* quando si analizzano dati spaziali, soprattutto se sono collegati agli spostamenti urbani (siano essi traiettorie o *moving object*) i quali sono vincolati alla rete di trasporto. Per dimostrare i potenziali problemi che potrebbero sorgere nell'analizzare i dati collegati ad un *network* utilizzando metodi non specifici, Atsuyuki Okabe e Koki chi Sugihara [27] propongono l'esempio riportato in Figura 16:

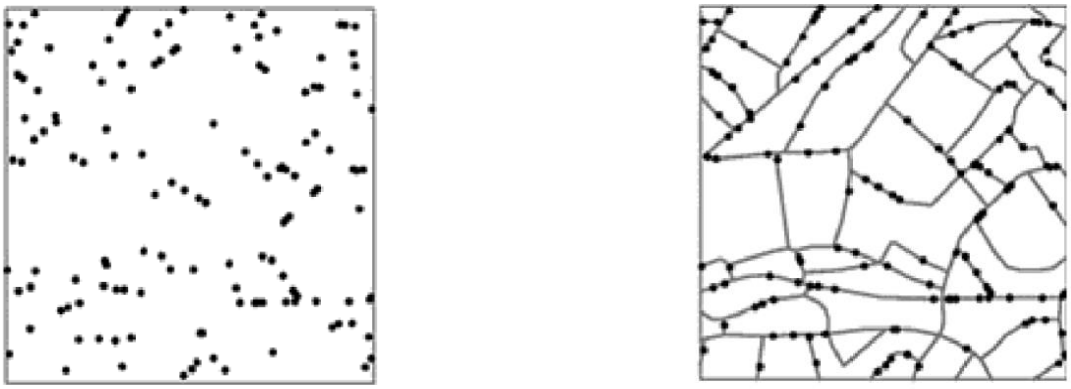


Figura 16 (a) distribuzione di punti non casuale in un piano (b) distribuzione di punti casuali in un network (si nota che la distribuzione dei punti è la stessa)

(a) *(b)*

Data la distribuzione di punti in Figura 16 (a), risulta difficile asserire che questi sono distribuiti in maniere casuale; tuttavia, tale affermazione è vera se si considera la Figura 16 (b): infatti, in tal caso i punti sono stati generati secondo una distribuzione uniforme a partire dal *network* tracciato.

Nel caso di *network data,* la distanza tra gli oggetti caratterizzati da una dimensione spaziale si basa sulla lunghezza dei segmenti che collegano il punto di partenza e quello di arrivo considerando un trasporto fisico reale.

Di altrettanta importanza è garantire una certa flessibilità sulla granularità e sul peso degli attributi temporali e spaziali considerati per il *clustering*. Per questo motivo Hesam Izakian et ali [26] applicano l'algoritmo di clustering *fuzzy C-Means,* utilizzano due indici di performance: *reconstruction error* e *prediction error;* il primo permette di definire la granularità degli attributi temporali e spaziali; il secondo è utilizzato quando si intende fare una previsione della componente temporale, data quella spaziale.

Lamb et al [14] propongono una funzione di distanza generica $\boldsymbol{D}(\boldsymbol{x_i}, \boldsymbol{x_j})$ date due osservazioni, $\boldsymbol{x_i}$ e $\boldsymbol{x_j}$, la loro similarità viene misurata dalla combinazione della loro posizione nel tempo e nello spazio e qualsiasi altra variabile esterna, detta '*metrica*'. Ogni componente è gestita con un diverso calcolo della distanza, quindi normalizzata e ponderata prima di sommare.

La funzione di distanza $\boldsymbol{D}(\boldsymbol{x_i}, \boldsymbol{x_j})$ calcola la distanza tra due oggetti rappresentati come vettori $\boldsymbol{x_i}$ e $\boldsymbol{x_j}$. La distanza tra le posizioni $\boldsymbol{D_S}$ così come quella tra il tempo $\boldsymbol{D_T}$ viene ridimensionata, o normalizzata, dividendo rispetto a fattori di scala $\boldsymbol{S_S}$ e $\boldsymbol{S_T}$ e dato un peso, rispettivamente $\boldsymbol{w_1}$ e $\boldsymbol{w_2}$. L'ultima componente della funzione distanza $\boldsymbol{D}(\boldsymbol{a_i}, \boldsymbol{a_j})$ è data dalla sommatoria delle distanze di ciascuna dimensione né temporale né spaziale, ognuna delle quali è calcolata separatamente. Anche in questo caso, la distanza di ogni attributo viene ridimensionata con un fattore di scala $\boldsymbol{S_a}$, a seconda del tipo di variabile e viene assegnato un peso di $\boldsymbol{w_a}$. La somma complessiva viene divisa per la somma dei pesi. Riassumendo, la funzione di distanza proposta da Lamb et al [14] è espressa dalla seguente equazione:

Equazione 9 Funzione generale della distanza spazio-temporale

$$\boldsymbol{D(x_i, x_j) = \frac{1}{\sum_{i=1}^{n} w_i}\left(w_1 * \frac{D_S(s_i, s_j)}{S_S} + w_2 * \frac{D_T(t_i, t_j)}{S_T} + \sum_{a=3}^{q} w_a * \frac{D_a(a_i, a_j)}{S_a} \right)}$$

L'Equazione 9 è volutamente generale: in questo modo infatti diversi tipi di distanza $\boldsymbol{D_S(s_i, s_j)}, \boldsymbol{D_T(t_i, t_j)}, \boldsymbol{D_a(a_i, a_j)}$ possono essere facilmente utilizzate, in modo da essere adattate ai diversi tipi di dati e a diversa granularità sia spaziale che temporale. Inoltre, modificando solamente la metrica di distanza è possibile adattare gli algoritmi di *clustering* già presenti in letteratura ai dati spazio-temporali.

Come metrica di distanza **spaziale** $\boldsymbol{D_s}$ può utilizzata la distanza euclidea (Equazione 2) o la *network distance*.

La distanza **temporale** $\boldsymbol{D_T}$ può essere considerata sia in modo lineare (Equazione 10) che ciclico (Equazione 11):

Equazione 10 Distanza temporale lineare (o 'absolute')

$$D_T(t_i, t_j) = abs(t_i, t_j)$$

Equazione 11 Distanza temporale ciclica (o 'cyclical')

$$D_T(t_i, t_j) = \sqrt{t_{iC}^2 + t_{jC}^2 - (2 * t_{iC} * t_{jC} * t_{ij})}$$

con:

- $t_{ij} = \cos\left(\pi * \frac{\Delta t}{T}\right)$
- $\Delta t = abs(t_{iC} - i_{jC})$
- $T = 86'400s$
- $t_{iC} = abs(t_i - 43'200s)$ e $t_{jC} = abs(t_j - 43'200s)$

La Figura 17 mostra l'andamento della distanza *cyclical* calcolata rispetto alle ore 01:00 (curva blu) e rispetto alle ore 10:00 (curva arancione). Ad esempio, si nota che la distanza dalle ore 01:00 (curva blu) è minima in prossimità della fascia oraria 22:00-03:00, mentre è massima nell'intervallo dalle 11:00 alle 15:00.

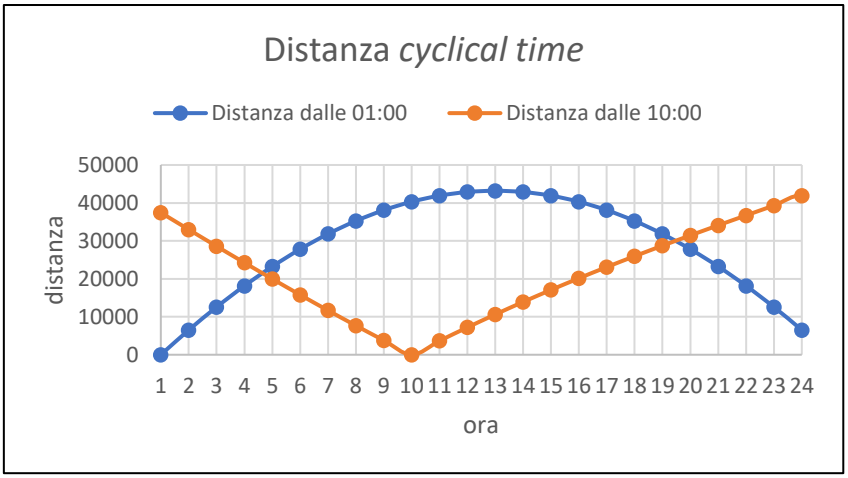


Figura 17 Esempio di distanza temporale 'cyclical'

La distanza degli attributi **non spaziali e temporali** $\boldsymbol{D_a}$ può essere semplicemente pari al valore assoluto della differenza tra i due valori, normalizzata rispetto ad un fattore di scala $\boldsymbol{S_a}$al fine di riportarla alla stessa scala di misura del tempo e dello spazio.

Anche Oliveira et al. [28] propongono una metrica di distanza simile a quella presentata, che chiamano ***4D,*** poiché permette di considerare (almeno) quattro dimensioni, di cui due spaziali, una temporale e una generica.

3.1. Distanza spaziale

Per ogni coppia di punti caratterizzati rispettivamente da due coordinate spaziali che ne indicano la posizione sulla Terra, la distanza spaziale $\boldsymbol{D_S}(s_i, s_j)$ può essere espressa in termini di:

- distanza Euclidea: nel caso di 2 dimensioni,

$$\boldsymbol{D_E}\,(s_i, s_j) = \sqrt{(s_{i1} - s_{j1})^2 + (s_{i2} - s_{j2})^2}$$

Equazione 12 Distanza euclidean per 2 dimensioni

(Oliveira et al. [28] considerano tale metrica per tutti gli attributi)

- *Haversine distance,* ovvero la distanza tra due punti sulla superficie terrestre, considerata di forma sferica.

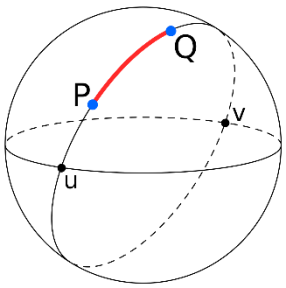
$$\boldsymbol{D_H}(s_i, s_j) = 2\,arcsin\left[\sqrt{sin^2\frac{s_{i1} - s_{j1}}{2} + cos(x_i)\,cos(s_{j1})\,sin^2\frac{s_{i2} - s_{j2}}{2}}\right]$$

Equazione 13 Distanza Haversine

con $s_i = s_i(x_i; y_i) = s_i(lat_i; lon_i)$ e $s_j = s_j(x_i; x_j) = s_j(lat_j; lon_j)$.

- *network distance* $\boldsymbol{D_G}$

Haversine distance



La distanza *Haversine* è considerata una metrica di distanza tra coordinate geografiche più appropriata ad esempio della distanza Euclidea, soprattutto quando la distanza tra due punti sulla Terra è molto grande e non può essere approssimata linearmente (Figura 18). Tale metrica è utilizzata ad esempio da Xiao et al. [29], che introducono una metrica di distanza, detta *TASTE* (*Text And Spatio-Temporal*), per il clustering dei tweets caratterizzati da un attributo spaziale, uno temporale e un testo.

Figura 18: Haversine distance

Network distance

Un modo per gestire la combinazione di dati vincolati ad un *network* e ad *event points* (o detti anche *free flowing data*), è definire un grafo non orientato a partire dalle posizioni in un insieme di *ST-Events.*

Idealmente, gli *ST-Events* riflettono il movimento tra le posizioni note. Ciò significa che le connessioni tra i punti riflettono il percorso che l'agente ha seguito per raggiungere quel punto. Per la costruzione del grafo Jeremy Mennis e Diansheng Guo [15] utilizzano prima la triangolazione di *Delaunay* (*Delaunay triangulation*), che crea un *network* di nodi adiacenti, i quali sono collegati da archi che non si intersecano.

La Figura 19 presenta un esempio di triangolazione di *Delaunay* (DT) per un set di dati che potenzialmente può essere ottenuto da un sistema di *traking* degli animali e la relativa triangolazione di *Delaunay* riflette il percorso che l'animale può seguire.

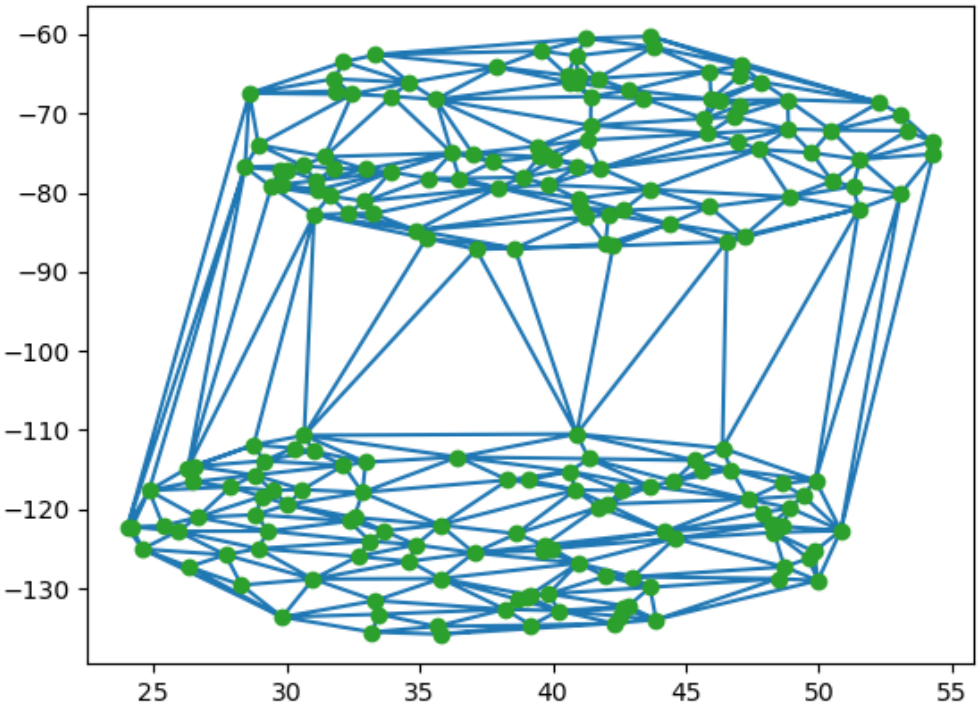


Figura 19 Esempio di una triangolazione di Delaunay

Al fine di calcolare la distanza tra i punti (*ST-Events*), la triangolazione di *Delaunay* è stata convertita in un grafo. Dato un grafo *G,* questo è caratterizzato da un insieme di nodi *V* e archi *E,* in modo che *G=(V, E).* Un arco appartenente all'insieme *E,* è caratterizzato da una coppia di nodi, appartenenti all'insieme *V.*

$$V = \{v_1, v_2, \dots, v_n\}$$

$$E = \left\{(v_i, v_j)_1, \dots, (v_i, v_j)_m\right\} = \{e_1, e_2, \dots., e_m\}; i \neq j\ ; i, j1,2, \dots n$$

Ad ogni coppia di nodi è assegnata uno un peso, che in questo caso rappresenta la distanza tra i nodi.

Dato un grafo G, si definisce percorso *(path)* dal nodo v_i a v_j la sequenza alternata di nodi e archi, tale che il primo nodo è v_i e l'ultimo è v_i a v_j*:*

$$Path = v_i, e_1, v_{i+1}, \dots., e_k, v_j$$

Dati due nodi v_i a v_j ci possono essere due o più percorsi; si definisce percorso di interesse (*path of interest*) il percorso che ha la distanza più breve, la quale può essere calcolata usando algoritmi come *Dijkstra* o A*. Di conseguenza, la distanza tra due nodi v_i a v_j coincide con la distanza minima del percorso, $\boldsymbol{D_G = (v_i, v_j)}$, detta *network distance.* Ha senso considerare tale metrica di distanza spaziale quando i punti (*ST-Events*) riflettono il movimento di oggetti vincolati ad un network.

3.2. Distanza temporale

La seconda componente dell' Equazione 9 è la metrica di distanza per la componente temporale $\boldsymbol{D_T(t_i, t_j)}$. Anche in questo caso ci sono diverse alternative, che variano in base alla concezione che si vuole dare del tempo nel contesto specifico.

Per esempio, se si considera il tempo in modo ciclico, si è interessati non tanto alla vicinanza temporale (ad esempio due eventi che si sono verificati nello stesso giorno), quanto alla similarità rispetto al periodo della

giornata (o del mese, anno, ecc..): quindi due eventi saranno considerati 'simili' se si sono manifestati nello stesso intervallo in una giornata (es. di mattina), seppur potenzialmente a mesi di distanza.

Se invece si ha una visione lineare del tempo, due eventi sono 'simili' se si sono verificati temporalmente in prossimità, quindi stesso anno, mese, giorno e ora. In questo caso, la distanza temporale è data dal valore assoluto della semplice differenza degli attributi temporali di due punti, espressi nell'unità di misura considerata, ad esempio in termini di secondi totali o giorni totali da una particolare data (per convenzione il 01/01/1970 ore 00:00), dunque le componenti temporali sono in termini di *timestamp*.

$$D_T(t_i, t_j) = abs(t_i - t_j)$$

Equazione 14 Metrica temporale 'absolute'

Nel caso in cui sia necessaria una visione del tempo ciclica, richiesta ad esempio quando nell'arco della giornata si ripetono degli specifici *patterns*, la dimensione temporale può essere espressa in termini di secondi dalla mezzanotte. Al fine di considerare "simili", quindi con minima distanza, instanti temporali come 0h e 23h 59min, la forma funzionale scelta per la metrica di distanza temporale è la seguente:

$$t_{ij} = \cos\left(\pi \frac{\Delta t}{T}\right); \qquad t_{ij} \in [0,1];$$

$$\Delta t = abs(t_{iC} - t_{jC})$$

$$T = 86\,400\ (seconds)$$

$$D_T(t_i, t_j) = \sqrt{\left(t_{iC}^2 + t_{jC}^2\right) - \left(2\ t_{iC} t_{jC} t_{ij}\right)}$$

Equazione 15 Metrica temporale: 'cyclical'

Ogni "evento temporale" t_i espresso in secondi dalla mezzanotte, è centrato rispetto a 43 200 s (t_{iC}).

3.3. Distanza degli attributi non spaziali e non temporali

Per gli attributi non spaziali e non temporali si può considerare come distanza la classica distanza Euclidea, o la distanza assoluta, opportunamente scalata per un fattore $S_a$al fine di riportare tali attributi sulla stessa scala di misura del tempo e dello spazio.

3.4. Fattori di scala

Una volta calcolata la distanza tra i due punti con la metrica di distanza scelta, quest'ultima viene scalata con un fattore corrispondente, $\boldsymbol{S_S}$ $\boldsymbol{S_T, S_a}$, necessario per eliminare l'unità di misura della distanza e portare la distanza sulla stessa scala degli altri componenti, e per creare vincoli spaziali legati a specifici fattori di scala che si vogliono utilizzare [30].
I fattori $\boldsymbol{S_S, S_T, S_a}$possono essere scelti in diversi modi, ad esempio si può considerare $\boldsymbol{S_S}$ pari alla massima distanza individuata tra i punti del dataset, oppure pari alla somma delle distanze del *network* $\boldsymbol{D_G}$ individuate.

Discorso analogo vale per la componente temporale. In questo caso, il fattore di scala $\boldsymbol{S_T}$ deve essere espresso nella stessa unità di misura dell'attributo temporale (ad esempio secondi). Ad esempio, considerando il fattore di scala di un'ora, si considera $\boldsymbol{S_T} = 3600\ s$, in modo tale che la distanza tra due istanti di tempo inferiori ad un'ora è inferiore a 1, mentre per distanze superiori ad un'ora la distanza è maggiore di uno.

Oliveira et al. [28] sottolineano che considerare come fattore di scala la distanza massima potrebbe penalizzare in modo eccessivo i punti che hanno tale distanza massima e inoltre è un metodo fortemente influenzato dalla presenza di *outliers.* Per superare tali problemi Oliveira et al. [28] suggeriscono di fissare i fattori di scala in base al dataset in analisi. Innanzitutto è necessario individuare la *bounding box* delle coordinate spaziali, ovvero il rettangolo di area minima che contiene tutti i punti del dataset in analisi, considerano le coordinate spaziali (es latitudine e longitudine); la *bounding box* (di solito abbreviata con *bbox*) di n-dimensionale richiede per la sua memorizzazione *2n* parole di memoria, una per ogni estremo degli n intervalli che la caratterizzano: nel caso di 2 dimensioni (es dataset in cui la posizione sulla Terra di ogni punto

è univocamente individuabile da una coppia di valori, quali latitudine e longitudine) la *bounding box* è individuabile da 4 dimensioni che corrispondono al vertice in basso a sinistra (*lower left*) e quello in alto a destra (*upper right*) della *bounding box,* codificate tipicamente nel modo seguente:

$$bbox = left, bottom, right, top$$

$$bbox = min\ Longitude\ , min\ Latitude\ , max\ Longitude\ , max\ Latitude$$

La Figura 20 mostra un esempio di *bounding box* di una mappa.

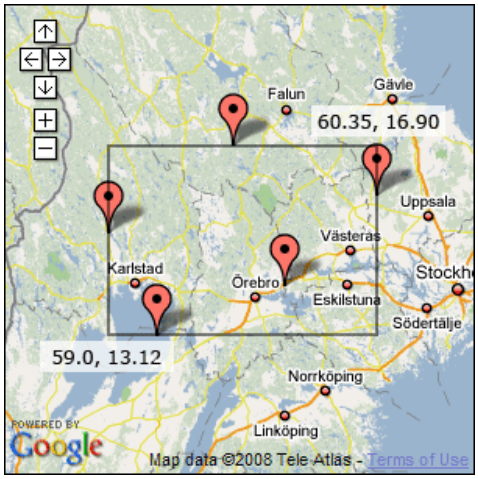


Figura 20 Esempio di bounding box di una mappa

Dopo aver individuate la *bounding box,* per ogni punto viene calcolata la distanza rispetto al vertice in basso a sinistra (*lower left*); tali distanze sono poi ordinate in modo ascendente. Dopodiché viene calcolata la differenza tra le distanze dei due punti consecutivi, le quali sono infine ordinate in modo ascendente. Tale metodo consente di individuare la distanza media tra punti vicini e la presenza di *outliers,* intesi come punti molto distanti dagli altri, considerando solo le coordinate spaziali. È un approccio simile a quello proposto dal *k-sort graphs.* La Figura 21 mostra un esempio del grafico ottenuto con questo approccio.

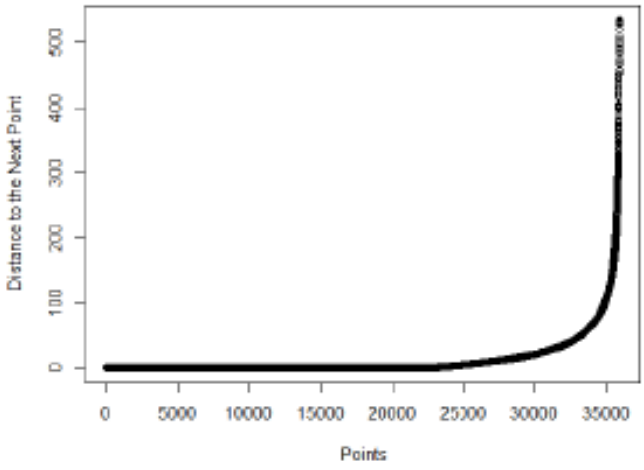


Figura 21 Esempio di sorted distances graph per la dimensione spaziale

Oliveira et al. [28], dopo aver analizzato diversi dataset, propongono come valore appropriato del fattore di scala S_S l'80 esimo percentile poiché tale valore divide le distanze che sono tipicamente presenti tra punti vicini da quelli che sono influenzati da *outliers.* Quando il *dataset* non presenta *outliers* e i dati sono spazialmente omogenei, la distanza tra i punti consecutivi è tipicamente prossima allo 0.

Per la dimensione temporale S_T è identificato mediante un processo analogo a quello appena descritto, considerando in questo caso la componente temporale rispetto alla quale viene calcolata la differenza con il

minimo *timestamp* del *dataset*. In modo analogo può essere ottenuto anche il fattore di scala S_a per tutti gli altri attributi.

Lambet al. [14] invece considerano come fattore di scala S_S per ogni cluster c_i la massima distanza dal *k-esimo vicino.* La scelta del *k-esimo* vicino è difficile; in letteratura sono presenti alcuni metodi euristici per la determinazione di tali parametri, inspirati ad esempio a quelli per determinare i parametri *Eps* e *MinPoint* per il *DBSCAN.*

Si specifica che nel presente lavoro di tesi, essendo presentata una metodologia generale per il *clustering* dei dati spazio-temporali non legata ad uno specifico ambito o *dataset,* i fattori di scala sono posti pari ad 1, in quanto non si ha una particolare esigenza di modificare la scala delle componenti spaziali e temporali (o comunque si tratta di una scelta legata all'ambito specifico della *data analysis*).

3.5. Metriche di distanza scelte

Le metriche di distanza sopradescritte per le componenti spaziali e temporali possono essere combinate al fine di ottenere un'unica metrica di distanza. A seconda delle metriche scelte si otterrà un risultato di *cluster* diverso. I criteri che orientano tali scelte non sono univoci e possono dipendere ad esempio dalla tipologia di dataset, dalla distribuzione dei dati o dall'obiettivo del *data analyst.*

Di seguito si riportano alcuni criteri che possono orientare la scelta sulla base della tipologia di dato e dell'unità di misura; si può optare per:

- metrica di distanza spaziale *Haversine,* se latitudine e longitudine sono espressi in gradi; al contrario si potrebbe utilizzare la metrica *Euclidea* se i dati spaziali utilizzano il sistema di coordinate UTM (*Universal Transverse Mercator).* In tal modo, non è necessario di convertire i dati da un sistema di coordinate all'altro, con conseguente perdita di informazioni;
- metrica di distanza spaziale *Haversine* quando la distanza tra due punti sulla Terra è molto grande e non può essere approssimata linearmente; in tal caso la distanza spaziale *Haversine* è più più appropriata ad esempio della distanza Euclidea
- metrica di distanza temporale *absolute,* quando il *data analyst* non è interessato ad identificare specifici *patterns* che si ripetono nell'arco di una giornata, ma l'analisi è finalizzata ad individuare cluster che variano sulla base dello specifico giorno.
  Al contrario, si può optare per una distanza temporale *cyclical,* nel caso in cui sia necessaria una visione del tempo ciclica e l'analisi è finalizzata ad individuare *patterns* che si ripetono all'interno di una giornata, indipendentemente dal giorno specifico

Analogamente, il **peso** delle componenti spaziali, temporali e in generali degli attributi non *spazio-temporali* possono essere determinati:

- sulla base dell'obiettivo dell'analisi, quindi scegliere di attribuire un peso maggiore ad una determinata componente riducendo l'influenza delle altre, in modo da orientare i risultati del *cluster*
- in modo da massimizzare un indice qualità del cluster, ad esempio l'indice di *Silhouette.*

Nel caso specifico, il paragrafo 6.2 contiene i risultati relativi al *clustering* del *GPS trajectory dataset*, ovvero una porzione di dataset raccolto da *Microsoft Research Asia* durante il progetto *Geolife* [2] *,* in cui sono stati raccolti tramite *smartphone* i dati relativi agli spostamenti di 182 utenti volontari in un periodo di 5 anni. In particolare, il paragrafo 6.2 riporta i risultati del *clustering* ottenuti:

- fissando in input i pesi delle componenti spaziali w_S, temporali w_T e metriche w_a
- determinando in modo automatico i pesi delle componenti spaziali w_S, temporali w_T e metriche w_a in modo da massimizzare l'indice di *Silhouette*

Tale dataset è stato clusterizzato utilizzando la seguente metrica di distanza:

Equazione 16: metrica di distanza ST scelta

$$D(x_i, x_j) = w_S * D_S(x_i, x_j) + w_T * D_T(t_i, t_j) + w_a * D_a(a_i, a_j)$$

Dove:

- distanza spaziale euclidea $\boldsymbol{D_S = D_E(x_i, x_j)} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$
- distanza temporale *absolute* $\boldsymbol{D_T = abs(t_i - t_j)}$
- distanza euclidea per la componente non ST $\boldsymbol{D_a = D_a(a_i, a_j)} = \sqrt{(a_{i1} - a_{j1})^2}$

Tutti gli altri dataset sono stati clusterizzati utilizzando la stessa metrica di distanza, ma fissando ad 1 il peso delle componenti non spaziali-temporali (quindi $\boldsymbol{w_a = 1}$), modificando solo $\boldsymbol{w_S}$ e $\boldsymbol{w_T}$, con il vincolo che la loro somma fosse unitaria ($\boldsymbol{w_S + w_T = 1}$).

Quindi è stato scelto di modificare in modo complementare le sole componenti spaziali e temporali, al fine di identificare dei cluster che tenessero conto in modo complementare dello spazio e del tempo, e allo stesso tempo mantenendo un peso indipendente per la metrica. In questo modo il *cluster* tiene conto innanzitutto della metrica/misura, quindi dell'attributo che caratterizza principalmente il punto, mentre viene modificata l'influenza della geolocalizzazione o dell'istante temporale di rilevazione, attributi che potrebbero essere più o meno interessanti per il *data analyst.*

Inoltre, i vincoli $\boldsymbol{w_a = 1}$ e $\boldsymbol{w_S + w_T = 1}$ semplificano l'analisi, riducendo i parametri indipendenti da fissare in input. Infatti, in tal modo è necessario determinare il peso di una sola componente (ad esempio in modo automatico, massimizzando l'indice di *Silhouette*).

I risultati sono riportati nel capitolo 5. per i dataset simulati, nel paragrafo 6.3 per il *dataset SF Park* e nel paragrafo 6.4 per il *dataset Bikesharing*.

4. Indici

La validazione dei risultati ottenuti dall'applicazione degli algoritmi di *clustering* è la parte più complessa, ma anche quella che apporta maggiore valore aggiunto, dal momento che senza un'adeguata interpretazione dei *cluster* diventa difficile sostenere la validità dei risultati ottenuti.

Un metodo oggettivo utilizzato per confrontare i risultati ottenuti da diversi algoritmi di *clustering* è l'utilizzo di indici, che consentono di valutare la qualità di un *cluster.* Di seguito si riportano i principali indici utilizzati in letteratura.

Quando è nota a priori la classe/etichetta di appartenenza del *cluster,* è possibile valutare l'***Homogeneity Score****,* ovvero una misura che serve per verificare che un *cluster* contenga elementi appartenenti solo ad una classe, quindi per misurarne la purezza. Tale indice è così definito:

Equazione 17 Homogeneity Score

$$\begin{cases} h = 1 \; se \; H(C) = 0 \\ h = 1 - \dfrac{H(C|K)}{H(C)} \end{cases}$$

In una soluzione ideale di *cluster* omogeneo il rapporto $\frac{H(C|K)}{H(C)}$ è uguale a 0; dunque un risultato desiderabile corrisponde ad un *Homogeneity Score h* massimo, pari a 1. Si nota quindi che nella definizione dell'*Homogeneity Score* compare anche quella dell'entropia condizionale $H(C|K)$ della suddivisione dei dati nelle classi *ground-truth* C rispetto alla suddivisione dei dati nei cluster *K* e l'entropia della classe $H(C)$. Questi indici sono così definiti:

Equazione 18 Entropia condizionale

$$H(C|K) = -\sum_{c=1}^{C}\sum_{k=1}^{K}\frac{n_{c,k}}{n}\log\frac{n_{c,k}}{n_k}$$

dove n è il numero totale di campioni, n_k è il numero di campioni che appartiene al cluster *K* e $n_{c,k}$ è il numero di campioni della classe C che sono stati assegnati al cluster K. Maggiore è l'entropia $H(C|K)$, maggiore è la suddivisione in modo disordinato delle classi C nei cluster K.

Si definisce, poi, l'entropia di una classe:

Equazione 19 Entropia di una classe

$$H(C) = -\sum_{c=1}^{C}\frac{n_c}{n}\log\frac{n_c}{n}$$

dove n è il numero totale di campioni e n_c è il numero di campioni che appartengono alla classe *c*. $H(C)$ assume valori elevati quando le varie classi *C* hanno un numero simile di elementi n_c [31].

Un secondo indice è l'***Adjusted Rand Index***, ovvero la misura della similarità tra due un'assegnazione di etichette fatta dal *clustering* con il *ground truth.* In particolare, l'*Adjusted Rand Index* è in grado di ignorare le permutazioni. Supponiamo di avere un *ground truth* come assegnato in Tabella 2: se calcoliamo l'ARI dell'assegnazione a) rispetto al *ground truth* otteniamo un certo punteggio; se decidessimo di sostituire le etichette dell'assegnazione a), in particolare gli 0 con gli 1, gli 1 con i 3 e i 2 con i 4, ottenendo così l'assegnazione b), l'ARI non cambierebbe affatto.

Tabella 2 Esempio di diverse assegnazioni di clustering paragonate al ground truth

Ground truth	**0**	**0**	**0**	**1**	**1**	**1**
Assegnazione a	0	0	1	1	2	2
Assegnazione b	1	1	3	3	4	4

In caso di un *clustering* perfetto in cui tutte le etichette assegnate ai punti coincidessero con le etichette di classe reali, il valore massimo raggiunto dall'ARI è di 1. Un'assegnazione casuale nei *cluster* produce un *adjusted rand index* di 0, mentre il punteggio peggiore è -1.

Nella definizione dell'*adjusted rand index*, è presente quella del *rand index.*

Dato un insieme di *n* elementi $S = \{O_1, \dots, O_n\}$, siano $U = \{u_1, \dots, u_R\}$ e $V = \{v_1, .., v_C\}$ due diverse partizioni di *S* tale che $\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j$ e $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ per $1 \le i \ne i' \le R$ e $1 \le j \ne j' \le C$. Nel caso specifico U è il *ground truth* mentre V è il risultato del *clustering.*

Sia:

- a è il numero di coppie di elementi che sono posizionati nella stessa classe in U e nello stesso cluster in V
- b è il numero di coppie di elementi nella stessa classe in V ma posti in cluster diversi in U
- c è il numero di coppie di elementi nello stesso cluster in V ma non della stessa classe in U
- d è il numero di coppie di elementi in differenti classi in U e diversi cluster in V
- n è il numero totale degli elementi

Il *Rand Index* è così definito:

Equazione 20 Rand Index

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

dove a e d possono essere considerati come *agreement,* mentre c e d come *disagreement*. Il *Rand Index* quindi può assumere valori compresi tra 0 e 1 ed è uguale a 1 quando si individua un *cluster* completamente compatibile con il *ground truth.*

Il *Rand Index* ha il difetto che non restituisce 0 per un'assegnazione di cluster casuale. Si può porre rimedio a questo effetto ricorrendo all' *Adjusted Rand Index*, assumendo come modello di casualità una generica distribuzione ipergeometrica; considerando il valore atteso dell'indice $E[I]$, la definzione dell'ARI è la seguente:

Equazione 21 Adjusted Rand Index (1/2)

$$ARI = \frac{index - expected\ index}{\text{maximum}\ index - expected\ index} = \frac{RI - E[I]}{\max(RI) - E[RI]}$$

Per una definizione più formale dell'*Adjusted Rand Index* si propone la seguente notazione [32]. La Tabella 3 è un esempio di tabella di contingenza:

Tabella 3 Tabella di contingenza per la definizione dell'Adjusted Rand Score

CLUSTER / CLASSE	V_1	V_2	…	V_S	*SUM*
U_1	n_{11}	n_{12}	⋮	n_{1s}	a_1
U_2	n_{21}	n_{22}	⋮	n_{2s}	a_2
⋮	⋮	⋮	⋮	⋮	⋮
U_R	n_{r1}	n_{r2}	…	n_{rs}	a_r
SUM	b_1	b_2	…	b_s	

Equazione 22 Adjusted Rand Index (2/2)

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right]}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right]}{\binom{n}{2}}}$$

L' *Adjusted Rand Index* ha valore massimo pari a 1: un elevato ARI indica un'elevata corrispondenza tra *ground truth* e assegnazione del *clustering.* A differenza del *Rand Index* che può assumere valori compresi tra 0 and 1, l'*Adjusted Rand Index* può assumere anche valori negativi, quando l'indice *RI* è più basso del valore atteso.

L'Homogeneity Score e *l'Adjusted Rand Index* sono indici che possono essere utilizzati quando è nota l'etichetta della classe di appartenenza, come nel caso di *dataset* simulati (Capitolo 5).

Nel caso di *dataset* reali, quando non è nota la conoscenza a priori della classe di appartenenza e al contrario il *clustering* è una tecnica utilizzata per una prima comprensione della distribuzione dei dati, l'indice più utilizzato è la *Silhouette.*

L'indice di ***Silhouette*** combina il concetto di coesione e quello di separazione (Figura 22). Per ogni oggetto, i si definisce:

- $a(i)$ è la distanza media di i con tutti gli altri oggetti del **proprio cluster**; è una metrica di ***coesione***
- $b(i)$ è la distanza più bassa tra le distanze medie tra i e tutti gli oggetti appartenenti agli **altri *cluster*** (tranne il cluster di cui i è membro; è una metrica di ***separazione***

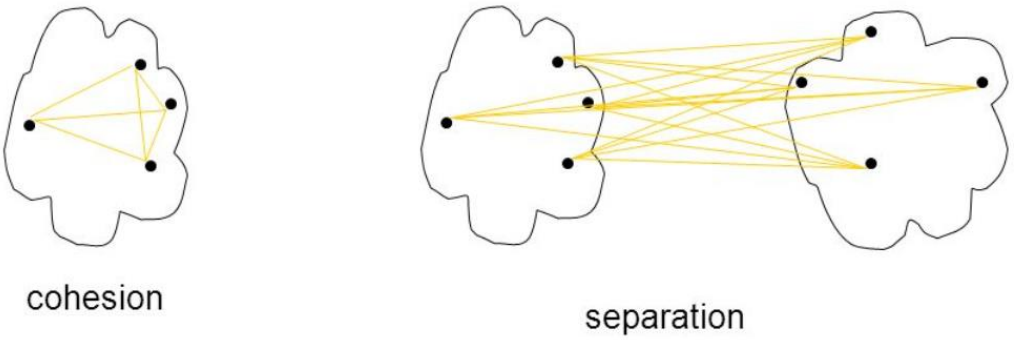


Figura 22 Concetto di coesione $a(i)$ e separazione di un cluster $b(i)$

L'indice di *Silhouette* viene quindi calcolato secondo la seguente equazione:

Equazione 23 Indice di silhouette

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

Il valore di tale coefficiente può oscillare tra -1 e 1. Valori negativi indicano scarsa qualità del *cluster*, poichè indica che $a(i)$ è più grande di $b(i)$, cioè la distanza media dell'oggetto i con gli elementi del suo *cluster* è più grande della minima tra le distanze medie con gli oggetti degli altri *cluster*.

Per calcolare il valore della *Silhouette* di un *cluster*, si calcola la media dei valori di *Silhouette* di tutti i membri appartenenti allo stesso *cluster*. Infine, per calcolare il valore di *Silhouette* di un intero *clustering*, si calcola la media dei valori di *Silhouette* di ogni *cluster*.

Un quarto indice utilizzabile per la valutazione della bontà di un *cluster* è l'indice di ***Dunn*** (***Dunn Index - DI***), introdotto da JC Dunn nel 1974. Si tratta di una metrica per la valutazione degli algoritmi di *clustering*, che fa parte del gruppo di indici di validità accanto all' indice ***Davies – Bouldin*** o l'indice di ***Silhouette***. Come tutti gli altri indici simili, lo scopo è identificare *cluster* che siano compatti, caratterizzati da una minima distanza *intra-cluster* tra i punti appartenenti ad uno stesso cluster, e allo stesso tempo caratterizzati da una massima distanza *iter-cluster*, tra i punti appartenenti a *cluster* diversi.

Il *Dunn Index* [33] è definito come il rapporto tra la distanza minima e la distanza massima:

Equazione 24 Dunn Index

$$Dunn\ Index = \frac{d_{min}}{d_{max}}$$

Si nota che per la definizione specifica del *Dunn Index* è necessario definire le modalità di calcolo della distanza minima e della distanza massima. Possono essere utilizzate diverse definizioni, tuttavia quella più adottata definisce $\boldsymbol{d_{min}}$come la distanza minima tra i punti di cluster diversi mentre $\boldsymbol{d_{max}}$ indica la distanza massima tra punti calcolata tra tutti i punti appartenenti ad uno stesso *cluster.*

In tal caso, un cluster compatto e ben separato sarà caratterizzato da un diametro del *cluster* (d_{max}) molto piccolo e la distanza tra i cluster dovrebbe essere molto grande (d_{min}). Pertanto, dato un *cluster* un alto valore di *Dunn Index* indica una alta qualità del *cluster,* quindi è indicativo di un buon risultato.

La distanza tra il cluster C_k e $C_{k'}$ è pari alla distanza minima ($\boldsymbol{d_{min}}$) tra i punti più vicini appartenenti a *cluster* diversi

$$d_{kk'} = \min_{\substack{i \in I_k \\ j \in I_{k'}}} ||M_i^k - M_j^{k'}||$$

M_i^kè il punto *i-esimo* appartenete al *cluster k,* mentre $M_j^{k'}$è il punto *j-esimo* appartenente al *cluster k'.*

La distanza d_{min} è pari alla minima distanza calcolata tra tutte le combinazioni di punti appartenenti a *cluster* diversi.

$$d_{min} = \min_{k \neq k'} d_{kk'}$$

Per ogni *cluster,* inoltre, si definisce D_k come la distanza massima registrata tra i punti apparteneti ad uno stesso *cluster* (definita a volte come *"diametro del cluster"*):

$$D_k = \max_{\substack{i,j \in I_k \\ i \neq j}} ||M_i^k - M_j^k||$$

La distanza d_{max} è pari alla massima distanza D_k calcolata per ogni cluster:

$$d_{max} = \max_{1 \leq k \leq K} D_k$$

5. *Clustering* di dataset *ST-Events* simulati

Come specificato nei capitoli precedenti, è stato scelto l'algoritmo gerarchico in quanto questo non richiede di fissare a priori il numero di cluster, ma viene identificato un insieme di cluster annidati. Il numero di cluster può esser selezionato successivamente in base ai seguenti tre criteri [14]:

- empirico: viene selezionato un numero di cluster tale che il fenomeno in esame può essere spiegato
- operativo: viene identificato un numero di cluster che corrisponde ad una scala temporale e spaziale rispetto alla quale il fenomeno in esame si sviluppa
- *fit*: viene selezionato un numero di cluster in modo da massimizzare un indice, ad esempio la *Silhouette.*

I primi due metodi richiedono la conoscenza del fenomeno e del contesto specifico, quindi non possono essere generalizzati; per questo motivo ne presente lavoro di tesi è stato implementato solo l'ultimo approccio *(fit)*.

Al fine di scegliere i fattori di scala e il numero di cluster Lambet et al. [14] utilizzano 50 dataset simulati, ognuno costituito da 1000 elementi; tali dataset sono stati simulati grazie a *make_blobs* [34]*,* modulo di *Sci-Kit Learn* in *Python.*

Le variabili simulate che costituiscono i dataset utilizzati da Lambet et al. [14] sono tre: due spaziali (*UTM Easting* e *UTM Northing,* del sistema UTM - *Universal Transerve Mercator* – proiezione universale trasversa di Mercatore) e una temporale (secondi dalla mezzanotte); il modulo *make_blobs* seleziona tali variabili in modo casuale, a partire da una distribuzione normale, caratterizzata da un certo valore centrale (media) e deviazione standard.

I valori centrali sono selezionati da Lambet et al. [14] in modo casuale a partire da una distribuzione uniforme, i cui valori limite sono rispettivamente:

- 500 000 e 780 000 per la coordinata x, ovvero *UTM easting*
- 370 000 e 500 000 per la coordinata y, ovvero *UTM northing*
- 0 e 86400 per la componente temporale, che rappresentano i secondi in un giorno

La deviazione standard delle tre variabili è stata scelta in modo casuale a partire da una distribuzione uniforme con estremo inferiore pari a 1000 ed estremo superiore pari a 7000.

Per individuare i valori più adatti dei parametri, Lambet et al. [14] considerano due indici, *Homogeneity score (*Equazione 17*)* e *Average random index (*Equazione 21*)*. Tali indici possono essere utilizzati quando è noto il *ground truth,* ossia per ogni punto è nota l'etichetta della classe reale. Entrambi gli indici sono indipendenti dai valori assoluti delle etichette (*cluster label*): una permutazione dei valori dell'etichetta del *cluster* non cambia i valori degli indici.

5.1. Simulazione Dataset

Partendo dalla metodologia proposta da Lambet et al. [14], sono stati generati i *dataset* di prova per analizzare gli effetti dei diversi parametri (ad esempio delle diverse metriche per il calcolo della distanza).

Sono stati simulati 20 dataset, ognuno contenente 200 elementi: è stata scelta una cardinalità del campione più bassa al fine di contenere i tempi di calcolo.

Tali dataset sono stati simulati grazie a *make_blobs,* modulo di *Sci-Kit Learn* in *Python.*

Le variabili simulate che costituiscono i *dataset* sono quattro: due spaziali (latitudine e longitudine), una temporale (secondi dalla mezzanotte) e una metrica; il modulo *make_blobs* seleziona tali variabili in modo casuale, a partire da una distribuzione normale, caratterizzata da un certo valore centrale (media) e deviazione standard.

Il numero dei valori centrali e i corrispondenti valori sono selezionati in modo casuale a partire da una distribuzione uniforme, i cui valori limite sono rispettivamente 0 e 1. Le distribuzioni uniformi sono generate grazie al modulo *uniform* della libreria *random* in *Python.*

La deviazione standard delle quattro variabili è selezionata in modo casuale a partire da una distribuzione uniforme, con valori compresi tra 0.01 e 0.10. Tali limiti sono stati fissati in modo da ottenere dataset non troppo dispersivi.

Le variabili così simulate assumono valori compresi tra 0 e 1; sono state dunque moltiplicate per i relativi limiti superiori, al fine di ricondurle ai relativi domini. La funzione *make_blobs* individua variabili casuali che seguono una distribuzione normale, dunque la probabilità che vengano selezionate variabili non comprese nel dominio $[-3\sigma; 3\sigma]$ è molto bassa, ma non nulla. Nello specifico: $99{,}7\% = P\{\mu - 3{,}00\,\sigma < X < \mu + 3{,}00\,\sigma\}$; poiché tale probabilità non è nulla, è stato necessario riportare nel dominio tutte le variabili casuali non ammissibili.

In particolare, la variabile che esprime la latitudine è stata ricondotta ad un dominio [0°,84°]. Questo perché successivamente sarà necessario trasformare tali coordinate geografiche espresse in termini di Latitudine e Longitudine nel sistema di coordinate UTM, in cui il valore massimo della coordinata UTM northing è circa 9'300'000 metri, che corrisponde alla latitudine 84°. Inoltre, è stato scelto come limite inferiore 0° perchè la libreria *utm* di Python utilizzata per la conversione supporta solo valori di latitudine dello stesso segno. Tuttavia, nell'emisfero sud i limiti sono 10'000'000 metri in prossimità dell'equatore e 1'100'000 metri in prossimità del 80° di latitudine. Riassumendo, le variabili considerate e i relativi domini sono:

- 0 e 84° per la coordinata Latitudine
- -180 e 180° per la coordinata Longitudine
- 1 e 86'400 per la dimensione tempo, in termini di secondi in un giorno a partire dalla mezzanotte

In particolare, è stato ritenuto opportuno modificare alcuni parametri rispetto a quanto presentato da Lambet et al. [14] poiché esprimere le coordinate spaziali in termini di latitudine e longitudine consente di:

- sviluppare un'analisi adatta ad un maggior numero di dataset, in quanto il formato 'latitudine/longitudine' è quello principalmente adottato per indicare le coordinate geografiche
- applicare la metrica di distanza Haversine senza applicare conversioni.
  Se invece si sceglie di utilizzare la distanza Euclidea è necessario trasformare le coordiate da formato 'latitudine/longitudine' a formato metrico, ad esempio UTM *Easting*/ UTM *Northing*.

Per valutare l'impatto dei vari parametri, oltre a considerare *Homogeneity score* e *Average random index* implementati da Lambet et al. [14] , è stato considerato un terzo indice, la *Silhouette,* descritta in modo formale successivamente.

Altre modifiche riguardano la metrica di distanza utilizzata; per la distanza spaziale è possibile scegliere tra:

- distanza euclidea, adatta per piccole distanze
- *Haversine distance,* adatta per distanze più elevate

Per la distanza temporale si può scegliere tra:

- distanza assoluta, quando è importante la data esatta
- distanza ciclica, quando è importante il momento della giornata, indipendentemente dalla data esatta.

Inoltre, è stato scelto di fissare i fattori di scala S_S, S_T e S_A pari ad 1, in quanto nel presente lavoro di tesi è presentata una metodologia generale per il *clustering* dei dati spazio-temporali non legata ad uno specifico ambito o *dataset,* dunque non si ha una particolare esigenza di modificare la scala delle componenti spaziali e temporali (o comunque si tratterebbe di una scelta legata all'ambito specifico della *data analysis*).

Per quanto riguarda i pesi da assegnare alla dimensione spaziale (w_1) e temporale (w_2) sono state eseguite simulazioni sia fissando tali parametri apriori (rispettando il vincolo unitario della somma), sia in modo tale che vengano scelti automaticamente, massimizzando uno degli indici. Discorso analogo per il parametro che indica il numero di *cluster.*

I parametri che sono stati fissati per effettuare le varie simulazioni sono sintetizzati nella Tabella 4 di seguito.

Tabella 4 Parametri dell'agglomerative clustering e del K-Medoids variati per le simulazioni

Parametro	***Dominio***
Metrica di distanza	*Euclidean distance* (Equazione 12) *Haversine distance* (Equazione 13*)*
Metrica temporale	*Absolute* (Equazione 14)
Peso delle dimensioni	- pesi impostati come input (in tal caso, il peso delle dimensioni è soggetto al vincolo della somma unitaria). - pesi fissati in modo automatico, tali da massimizzare l'indice di *Silhouette* e il *Dunn Index*
Criterio di *linkage* (per *agglomerative clustering)*	*"average":* usa la distanza media di ogni osservazione tra i due *sets* [35]
Numero di cluster	- il numero di cluster impostato come input - il numero di cluster viene fissato in modo automatico, tali da massimizzare l'indice di *Silhouette* e il *Dunn Index*

I *dataset* simulati sono stati clusterizzati con l'algoritmo *agglomerative e* con l'algoritmo *K-Medoids,* i cui parametri sono stati descritti nella Tabella 4 sopra riportata; inoltre, sono stati clusterizzati con l'algoritmo ST-DBSCAN, eseguito grazie all'omonima libreria di *Python* [19]. I parametri di quest'ultimo algoritmo sono:

- ***eps1***: massima distanza tra le componenti spaziali di due punti affinché vengano considerati appartenenti ad uno stesso cluster (*spatial density threshold*)
- ***eps2***: massima distanza tra le componenti temporali di due punti affinché vengano considerati appartenenti ad uno stesso cluster (*temporal density threshold*)
- ***min sample***: numero minimo di punti (*samples*) richiesti nell'intorno di un punto affinché venga considerato *core point*.

La libreria ST-DBSCAN richiede in input un dataset in cui il primo attributo è quello temporale, mentre i successivi vengono considerati come attributi spaziali. Per questo motivo, tutti i dataset simulati sono stati generati rispettando tale ordine. Si riporta un esempio della struttura del dataset generato (Tabella 5):

Tabella 5 Esempio di dataset ST-events (+ Metrica opzionale)

DATETIME	LATITUDE	LONGITUDE	METRIC [1]
10-07-1996 12:00	41.9028°	12.4964°	23
12-08-2020 12:00	43.8790°	12.7098°	31

I parametri dell'algoritmo ST-DBSCAN sono stati scelti in modo automatico, riprendendo quanto già implementato in *ADESCA* per l'algoritmo DBSCAN. In particolare, *eps1* e *eps2* vengono fissati in prossimità del gomito del grafico *k-dist*, ottenuto considerando rispettivamente solo gli attributi spaziali e solo l'attributo temporale. Il parametro *min sample* viene scelto applicando la funzione *minpnt* di ADESCA, con cui vengono prima calcolati i *minpoints* considerando rispettivamente solo l'attributo temporale e solo gli attributi spaziali, e successivamente il *min sample* viene fissato pari al minimo tra i due. L' algoritmo ST-DBSCAN così impostato non richiede nessun parametro fissato dall'utente.

5.1.1. PREPROCESSING

Poiché è stato scelto di considerare la metrica temporale *absolute* (Equazione 14) esprimendo la distanza fra due date come secondi che intercorrono fra queste, è necessario che l'attributo temporale venga prima convertito in *timestamp*, ovvero in un valore numerico intero che esprime il numero di secondi trascorsi da una data arbitraria; per convenzione, tale data è la mezzanotte del 1° gennaio 1970, momento che prende il nome di *epoch*.

Se invece si sceglie di considerare la metrica temporale *cyclical* (Equazione 11), considerando il tempo in modo ciclico e avendo l'intento di accomunare parti della giornata indipendentemente dalla data esatta, è necessario prima isolare l'ora dell'attributo temporale e poi trasformarlo successivamente in secondi in un giorno.

Per quanto riguarda la metrica spaziale, quando è stata scelta l'*euclidean distance* (Equazione 12), le coordinate in termini di latitudine e longitudine (gradi sessagesimali) devono essere convertire nel sistema di coordinate UTM. Tale conversione non è necessaria se si opta per la *Haversine distance*.

5.1.2. NORMALIZZAZIONE

Prima di applicare gli algoritmi di *clustering*, il *dataset* viene normalizzato, al fine di rendere i diversi attributi confrontabili. La normalizzazione è necessaria soprattutto quando sono analizzati dati eterogenei, come per esempio la misura della temperatura e della pressione, in diversi momenti e posizioni. Poichè le varie grandezze sono espresse su scale diverse, possono essere confrontate previa un'opportuna normalizzazione. I metodi di normalizzazione più usati in letteratura sono [5]:

- ***Min-Max o Range-based normalization***: i dati vengono riportati in un intervallo prefissato, tipicamente tra 0 e 1. Questa tipologia di normalizzazione è particolarmente indicata quando non si conosce la distribuzione dei dati oppure quando si ha conoscenza che la distribuzione non è gaussiana. Per ottenere la normalizzazione min-max, la trasformazione da effettuare è:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

  dove x_i è il valore che si vuole standardizzare, $min(x)$ e $max(x)$ sono rispettivamente valore minimo e valore massimo assunti dall'attributo x.

- ***Z-score o Standardization***: la standardizzazione dei dati è il processo di ridimensionamento di uno o più attributi in modo che abbiano un valore medio di 0 e una deviazione standard di 1. Secondo questo criterio, quindi, i dati normalizzati avranno le proprietà della distribuzione normale standardizzata. Questa tipologia di normalizzazione è particolarmente indicata perché facilita la

[1] Dimensione opzionale

gestione dei valori anomali, ma è utilizzabile in modo opportuno quando è noto che i dati seguono una distribuzione normale. Il valore ***z-score*** z_i è calcolato come segue:

$$z_i = \frac{x_i - \mu}{\sigma}$$

dove x_i è il valore che si vuole standardizzare, μ e σ sono rispettivamente media e deviazione standard dell'attributo x da normalizzare.

5.2. Risultati delle simulazioni

Variando i diversi parametri dell'algoritmo *agglomerative* e del *K-Medoids,* descritti precedentemente, sono state eseguite 20 simulazioni con altrettanti 20 dataset generati in modo casuale, secondo le modalità sopra descritte. I risultati ottenuti hanno mostrato che:

- il **peso** delle componenti spaziali e temporali che massimizzano un indice (*Silhouette* o *Dunn* Index) variano a seconda della distribuzione dei dati rispetto allo spazio e al tempo; quindi, non sarebbe del tutto corretto fissare a priori tali parametri, indipendentemente dal dataset. Al contrario, è preferibile fissare tali parametri sulla base di una prima analisi della distribuzione dei dati (in modo tale che, ad esempio, si attribuisce un basso peso alla componente temporale se i dati sono distribuiti in modo più o meno uniforme nel tempo, in quanto ciò potrebbe indicare una scarsa incidenza del tempo).

  Di conseguenza, per il cluster dei datatset reali *SF Park* e *Bikesharing* il peso delle componenti spaziali e temporali non è stato fissato a priori ma è stato considerato come un input modificabile.

- massimizzando la *Silhouette* e il *Dunn Index* come criterio per la scelta del **numero di cluster** si individua la maggior parte delle volte lo stesso numero di cluster. Dunque, il *Dunn Index* e la *Silhouette* possono essere considerati indici sostituibili.
  Di conseguenza, per il cluster dei datatset reali *SF Park* e *Bikesharing il numero di* cluster è stato determinato in modo da massimizzare il solo indice di *Silhouette* (non considerando il *Dunn Index*).

Inoltre, per i dataset simulati è stato possibil anche valutare la qualità del *cluster* con l'*Average Random Index (ARI)* e il *Homogeneity Score (h)*, utilizzabili quando è nota apriori l'etichetta di classe reale (*ground truth*). In questo caso, essendo i *dataset* simulati con la funzione *makeblobs* di *Python* tale informazione è nota. I due indici confrontano l'etichetta di classe (reale) con quella assegnata dai due algoritmi di *cluster.*

La metodologia introdotta nel capitolo precedente è stata implementata in *Python*, uno dei linguaggi di programmazione più utilizzati per l'applicazione di tecniche di *Data Mining* e *Machine Learning*.

Al fine di analizzare l'impatto dei vari parametri, come il peso della coordinata spaziale, il numero di cluster o la tipologia di metrica di distanza, sono stati preventivamente simulati dei dataset, i quali sono stati poi clusterizzati, modificando i diversi parametri in modo controllato. Tale approccio riprende quello utilizzato da Lambet et al. [14], i quali utilizzano 50 dataset simulati, ognuno costituito da 1000 elementi.

Nel caso specifico, i dataset sono stati simulati grazie a *make_blobs,* modulo di *Sci-Kit Learn* in *Python* [36].

È stato scelto di proseguire la fase implementativa tenendo in considerazione la tipologia di dataset spazio-temporale ***ST-Events****,* ovvero un dataset costituito da vettori che descrivono un evento associato a un *time-stamp* e ad una posizione fissa (dunque non è previsto nessun movimento o evoluzione nel tempo). In particolare, il dataset sintetico *ST-EVENTS* è caratterizzato da 4 attributi:

a. **tempo,** ovvero istante di tempo in cui il punto è rilevato; può essere espresso in secondi, ore o giorni
b. **spazio – latitudine** in cui il punto è rilevato
c. **spazio – longitudine** in cui il punto è rilevato

d. **metrica**: per metrica si intende qualsiasi altra dimensione che esprime una grandezza relativa al punto; il dataset può essere caratterizzato da una o più metriche e il dominio dipende dalla grandezza specifica.

Per ogni dataset simulato, sono stati implementati i seguenti algoritmi:

a. **K-Medoids,** al fine di testare un algoritmo *distance-based* di tipo partitivo *(partitional clustering)*
b. **Agglomerative,** al fine di testare un algoritmo *distance-based* di tipo gerarchico *(hierarchial clustering)*
c. **ST-DBSCAN,** al fine di testare un algoritmo *density-based*

Per le due tipologie di *clustering distance-based*, il numero di *cluster* è un parametro da definire in input ed è stato fissato in modo automatico, tale da massimizzare due indici utilizzati per valutare la coesione di un *cluster*, ovvero:

a. **silhouette index**
b. **dunn index.**

Per quanto riguarda invece l'**ST-DBSCAN,** i parametri da definire in input sono:

- **Min points:** numero minimo di punti (*samples*) richiesti nell'intorno di un punto affinché quest'ultimo venga considerato *core point*
- **Eps 1:** massima distanza tra le componenti **spaziali,** ed eventuali **metriche**, di due punti affinché vengano considerati come appartenenti ad uno stesso cluster (*spatial density threshold*)
- **Eps 2:** massima distanza tra le componenti **temporali** di due punti affinché vengano considerati appartenenti come ad uno stesso cluster (*temporal density threshold*)

I valori di **Min points, Eps1** e **Eps2** sono stati fissati in modo automatico in prossimità del gomito del *k-dist plot,* utilizzando le funzioni "*minpts*" e "*find_elbow*" implementate in ADESCA e descritta nei capitoli successivi.

Di seguito, si riportano i risultati ottenuti considerando due dataset sintetici (**dataset A** e **dataset B**), scelti a titolo di esempio tra quelli simulati. Sono stati considerati questi due in particolare, in quanto si tratta di dataset con una distribuzione di dati leggermente diversa: nel *dataset A* i dati sono più dispersivi e si estendono sia lungo la dimensione temporale, sia spaziale che metrica; invece, il dataset *B* presenta una distribuzione dei dati più compatta e ben separata. In tal modo è possibile mettere in luce le diverse influenze dei parametri che dipendono anche dalle caratteristiche del dataset.

5.2.1. Implementazione K-Medoids

L'algoritmo *K-Medoids* è un algoritmo di *clustering distance-based* di tipo partitivo. È stato implementato utilizzando il metodo ***scikit-learn-extra*** [37]***.*** Nel caso specifico, la funzione di distanza utilizzata è personalizzata, ovvero è calcolata utilizzando le funzioni descritte precedentemente nel paragrafo 3.5. In tal caso, il metodo ***scikit-learn-extra*** richiede in input la matrice di distanza, che è stata calcolata richiamando le suddette funzioni.

Quindi è stato necessario prima calcolare la matrice di distanza, utilizzando l'Equazione 16. Successivamente, è stato necessario applicare l'algoritmo *K-Medoids* attraverso il metodo ***KMedoids.fit*** di ***scikit-learn-extra,*** passando in input il numero di cluster desiderato e la matrice di distanza appena calcolata.

Nel caso dell'algoritmo *K-Medoids*, il numero di cluster da individuare è un parametro richiesto in input, che deve essere fissato dal *data analyst* o in modo automatico secondo un criterio deterministico. A tal proposito, sono stati implementati due automatismi che permettono di determinare il numero di cluster in modo da massimizzare due indici, ovvero:

- la Silhouette
- il Dunn Index

Di conseguenza, ogni dataset simulato è stato clusterizzato utilizzando il *K-Medoids* e ottenendo per quest'ultimo due risultati (un risultato ottenuto massimizzando la *Silhouette,* l'altro massimizzando il *Dunn Index*).

Accanto al numero di cluster, vi sono ulteriori **tre** parametri da fissare in input, che riguardano in tal caso il calcolo della matrice di distanza, ovvero:

- peso della coordinata temporale ($\boldsymbol{w_T}$); il peso della coordinata spaziale è linearmente dipendente, pari a $\boldsymbol{w_S = 1 - w_T}$
- metrica di distanza spaziale, scelta tra ***Haversine*** e ***Euclidean***
- metrica di distanza temporale, scelta tra ***Absolute*** e ***Cyclical.***

In questa prima fase, è stato scelto di non implementare un automatismo per la scelta del peso della coordinata temporale e per le due tipologie di metriche di distanza, in quanto sono parametri che verosimilmente possono essere variati dal *data analyst* sulla base dell'obiettivo che intende raggiungere o sulla base di altri criteri (fare riferimento al paragrafo 3.5).

Sono stati simulati due dataset (Dataset A e Dataset B) contenenti dati di tipo *ST-Events.*

Di seguito le principali fasi della simulazione:

1) è stata fissata la metrica di distanza spaziale (***Haversine* o *Euclidean***) e temporale (***Absolute****)*;
2) è stato incrementato progressivamente il peso della coordinata temporale, partendo da un valore pari a w_T*=0*, arrivando a w_T*=1* , con incremento di 0.05;
3) per ogni valore di w_T il dataset è stato clusterizzato 2 volte con algoritmo *K-Medoids,* ovvero fissando il numero di cluster in modo automatico tale da massimizzare la *Silhouette* e il *Dunn Index.*

Nei paragrafi successivi si riportano i risultati ottenuti clusterizzando il dataset A e il dataset B:

- con distanza spaziale sia Haversine che Euclidean
- fissando il numero di cluster massimizzando sia il Dunn Index che la Silhouette
- facendo prevalere sia la componente temporale (fissando $\boldsymbol{w_T = 0.95}$ e di conseguenza $\boldsymbol{w_S = 0.05}$) sia quella spaziale (fissando $\boldsymbol{w_S = 0.95}$ e di conseguenza $\boldsymbol{w_T = 0.05}$).

Per quanto riguarda la distanza temporale, è stata utilizzata solo la distanza temporale *absolute* (Equazione 10).

5.2.1.1. Dataset A – ST events: K Medoids

Il Dataset A è un dataset sintetico costituito da 200 punti (*ST-Events*) localizzati nell'emisfero boreale e concetrati nel Nord America, in Europa e nella zona dell'Oceano Pacifico in prossimità del Giappone ad est della Cina. Dalla rappresentazione geografica dei punti di seguito riportata emerge che, in linea generale, i punti si estendono in zone della Terra relativamente vaste e la componente spaziale sembra essere relativamente significativa. Considerando la sola distribuzione geografica, si possono identificare 3 macro aree (Nord America, Europa-Nord Africa e Oceano Pacifico). La macroarea Nord America potrebbe essere ulteriormente scomposta in *ST events* localizzati in Canada e negli Stati Uniti.

Considerando invece la componente temporale, dai grafici 3D della Latitudine, Longitudine e Secondi in un giorno si possono individuare 3 macrofasce temporali: tra 0s (ore 00:00) e 20'000 s (circa ore 05:30), tra 20'000 s e 60'000 s (circa ore 17:00) e tra 60'000 s e 86'400 s (circa ore 23:59).

Considerando invece la componente metrica, dai grafici 3D della Latitudine, Longitudine e metrica si possono individuare 2 macrofasce: tra 0 e 10 e tra 20 e 40.

5.2.1.1.1. Haversine distance con peso w_T=0.05

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset A e selezionando come metrica di distanza spaziale l'*Haversine.*

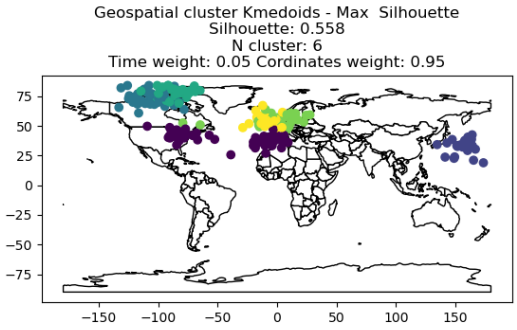
Inoltre, i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.05}$ (con prevalenza dello spazio) sono stati confrontati con quelli ottenuti fissando tale peso a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 23) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 24) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index.*

I diversi colori indicano l'appartenenza ad un cluster. In entrambi è stato ottenuto uno stesso risultato, identificando 6 *cluster*: segue che la scelta dell'indice come criterio di identificazione del numero di cluster in tal caso non è rilevante.

Avendo fissato come peso della componente spaziale $\boldsymbol{w_S = 0.95}$, tale dimensione prevale rispetto al tempo. Infatti, dai grafici di seguito emerge che ad esempio il cluster dell'Oceano Pacifico è ben distinto dagli altri; analogamente il cluster degli Stati Uniti è distinto dai cluster posizionati in prossimità del Nord America.

È stato ottenuto un indice di *Silhouette* pari a 0.558.



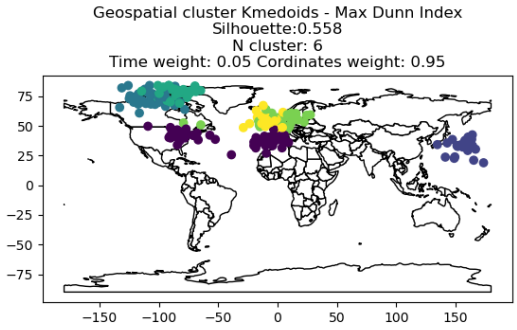


Figura 23: K-Medoids con n cluster identificato massimizzando la Silhouette (Dataset A) *Figura 24:* K-Medoids *con n cluster identificato massimizzando il Dunn Index (Dataset A)*

Poiché non è stata rilevata nessuna differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* per evitare la duplicazione di grafici, di seguito si riportano solo i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette.*

In particolare, sulla sinistra (Figura 25) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric. Si nota che, avendo un peso sempre unitario, la metrica è una componente discriminante per l'individuazione dei cluster (si noti ad esempio la distinzione in cluster diversi per quegli eventi con valori di metriche diverse).

Sulla destra (Figura 25) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno). Si nota che, avendo fissato un peso della componente temporale basso ($\boldsymbol{w_T = 0.05}$) i *cluster* individuati non hanno come discriminante principale il tempo (ad esempio, appartengono allo stesso *cluster* in viola sia punti con coordinata temporale bassa sia con quella alta).

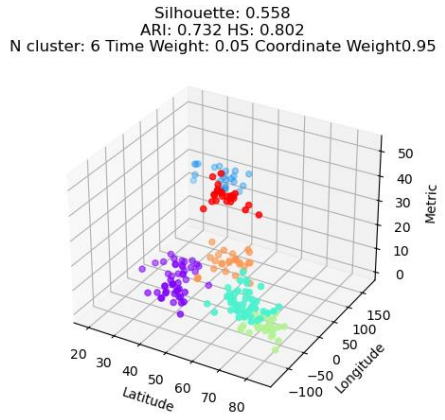


Figura 25:3D Lat, Long, Metric - KMedoids (Dataset A)

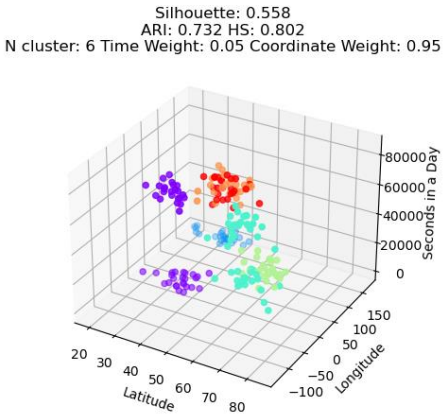


Figura 26: 3D Lat, Long, Time - KMedoids (Dataset A)

5.2.1.1.2. Haversine distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset A e selezionando come metrica di distanza spaziale l'*Haversine.*

Rispetto ai risultati descritti nel paragrafo precedente (5.2.1.1.1) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 27) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 27) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index.*

I diversi colori indicano l'appartenenza ad un cluster. In entrambi è stato ottenuto uno stesso risultato, identificando 4 *cluster*: segue che la scelta dell'indice come criterio di identificazione del numero di cluster anche in tal caso non è rilevante. Rispetto al caso precedente, sono stati individuati 2 cluster in meno.

Infatti, avendo fissato come peso della componente spaziale $\boldsymbol{w_S = 0.05}$, tale dimensione prevale di meno rispetto al caso precedente. Ad esempio, dai grafici di seguito emerge che, a differenza di prima, gli eventi relativi al Nord Africa appartengono allo stesso *cluster* di quelli ad Est della Cina.

In tal caso, è stato ottenuto un indice di *Silhouette* pari a 0.612, leggermente superiore al clustering precedente.

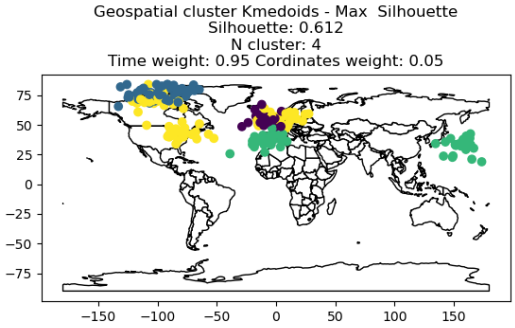


Figura 27:K-Medoids con cluster identificato massimizzando la Silhouette (Dataset A)

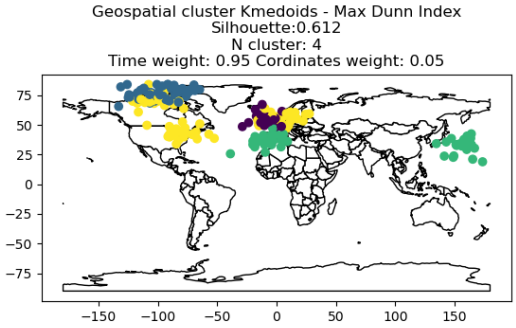


Figura 28: K-Medoids con n cluster identificato massimizzando il Dunn Index (Dataset A)

Poiché non è stata rilevata nessuna differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* per evitare la duplicazione di grafici, di seguito si riportano solo i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette.*

In particolare, sulla sinistra (Figura 29) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric. Si nota che, avendo un peso sempre unitario, la metrica è anche in tal caso una componente discriminante per l'individuazione dei cluster (si noti ad esempio la distinzione in cluster diversi per quegli eventi con valori di metriche diverse).

Sulla destra (Figura 29 Figura 24) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno). Si nota che, al contrario del caso precedente, avendo fissato un peso della componente temporale alto ($\boldsymbol{w_T = 0.95}$) i cluster individuati hanno come discriminante principale il tempo (ad esempio, appartengono allo stesso cluster in rosso punti con coordinata temporale alta).

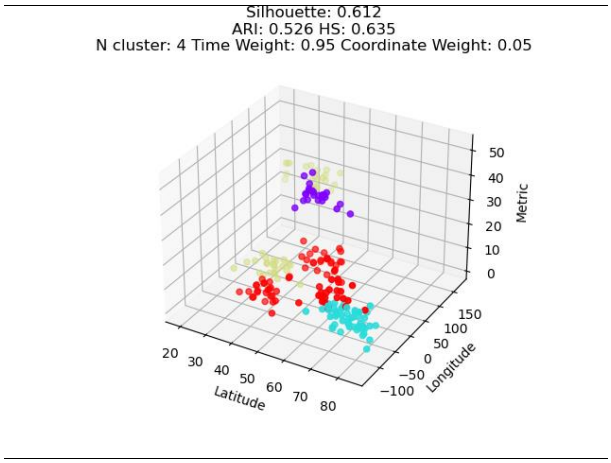


Figura 29: Lat, Long, Matric - KMedoids (Dataset A)

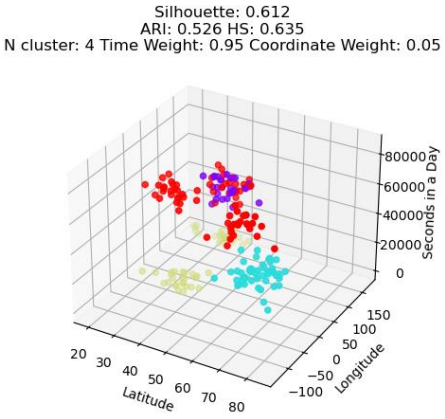


Figura 30: Lat, Long, Time - KMedoids (Dataset A)

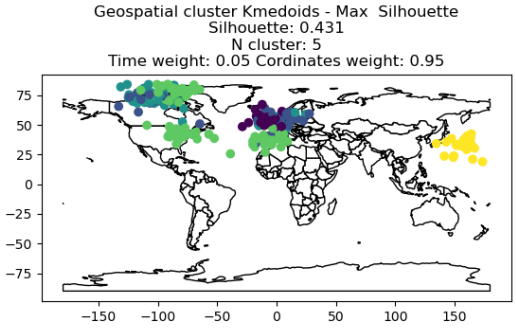
5.2.1.1.3. Euclidean distance con peso w_T =0.05

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset A e selezionando come metrica di distanza spaziale l'*Euclidean distance.* Tali risultati vengono confrontati con quelli riportati nel paragrafo 5.2.1.1.1 in cui viene scelta come metrica di distanza spaziale la *Haversine.*

Una prima differenza è che, utilizzando la distanza euclidea, si ottengono cluster diversi se si sceglie di determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index.*

Se si sceglie come criterio l'indice di *Silhouette* si ottengono 5 *cluster* (Figura 31); a differenza di quanto ottenuto nel paragrafo 5.2.1.1.1, in cui utilizzando la metrica spaziale *Haversine* a parità di altri fattori si identificano 6 cluster. La differenza principale tra i due risultati è che con la distanza euclidea non viene isolato il cluster relativo agli *ST Events* in prossimità degli Stati Uniti. Tale aspetto è compatibile con il fatto che l'*Euclidean distance* approssima la distanza tra 2 punti sulla Terra ad un segmento, riducendone di conseguenza tala distanza. Per quanto riguarda gli altri *cluster* non si apprezzano differenze sostanziali. L'indice di *Silhouette* è pari a 0.431, inferiore a quella ottenuta nel paragrafo 5.2.1.1.1 (pari a 0.558).

Se si sceglie come criterio il *Dunn Index* si identificano 3 *cluster* (Figura 31Figura 33), i quali non sembrano essere discriminati in base alla componente spaziale (sebbene il peso della coordinata spaziale sia pari a 0.95). Si ottiene infatti un basso indice di *Silhouette* (pari a 0.376).



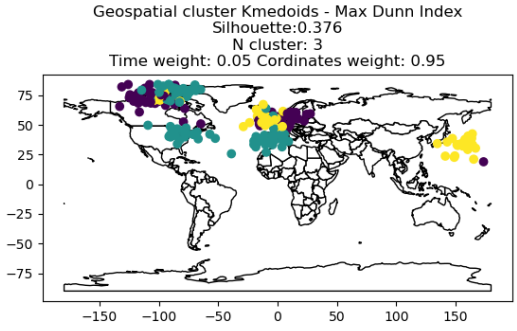


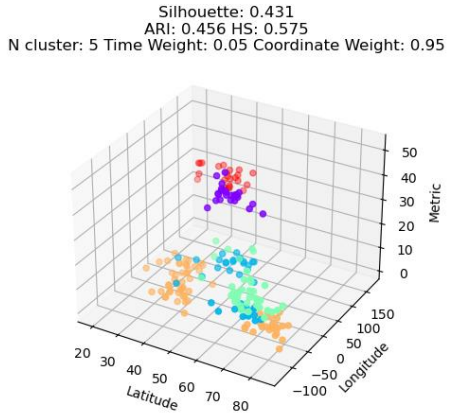
Figura 31:K-Medoids con cluster identificato massimizzando la Silhouette (Dataset A)

Figura 32: K-Medoids con n cluster identificato massimizzando il Dunn Index (Dataset A)

Poiché è stata rilevata differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* di seguito si riportano sia i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette* (sulla sinistra) che quelli ottenuti con la massimizzazione del *Dunn Index* (sulla destra).

In particolare, sulla sinistra (Figura 33Figura 29) si riporta il grafico 3D analogo a quello precedente ma ottenuto massimizzando l'indice di *Silhouette*. Si nota che, avendo un peso sempre unitario, la metrica è anche in tal caso una componente discriminante per l'individuazione dei cluster. Tuttavia, non è l'unica dimensione che sembra guidare il *cluster*: ad esempio si nota che gli *ST-Events* con metrica compresa tra 20 e 40 sono stati suddivisi in 2 cluster (in rosso e in viola*).*

Sulla destra (Figura 33 Figura 24) si riporta il grafico 3D degli *ST-Events,* clusterizzati massimizzando il *Dunn Index,* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric. Si nota che, avendo un peso sempre unitario, la metrica è in tal caso la componente maggiormente discriminante per l'individuazione dei cluster (si noti ad esempio la distinzione in *cluster* diversi per quegli eventi con valori di metriche diverse).



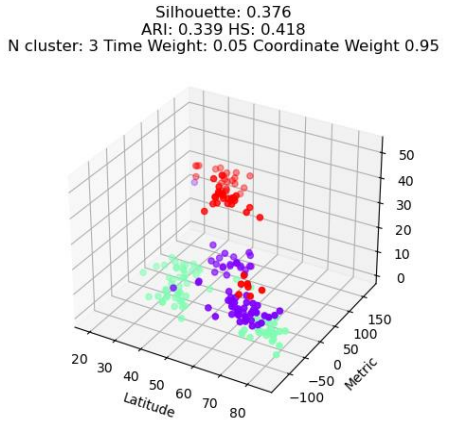


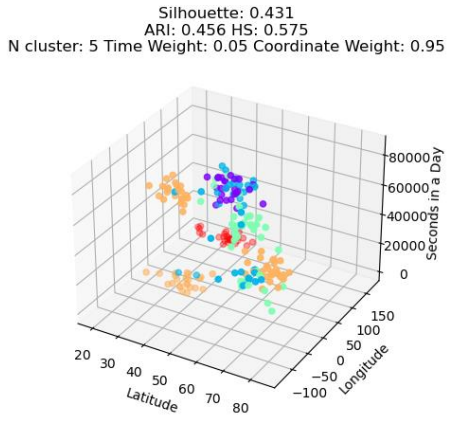
Figura 33: Lat, Long, Matric – KMedoids con cluster identificato massimizzando la Silhouette (Dataset A)

Figura 34: Lat,Long,Metric-KMedoids con cluster identificato massimizzando Dunn Index (Dataset A)

In particolare, sulla sinistra (Figura 35) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno), clusterizzati in modo da massimizzare la *Silhouette*. I risultati ottenuti sono molto simili rispetto a quelli descritti nel paragrafo 5.2.1.1.1.

Anche in tal caso, si nota che, avendo fissato un peso della componente temporale basso ($\boldsymbol{w_T = 0.05}$), i cluster individuati non hanno come discriminante principale il tempo (ad esempio, appartengono allo stesso cluster in arancione sia punti con coordinata temporale bassa sia con quella alta).

Sulla destra (Figura 35) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Time, clusterizzati in modo da massimizzare il *Dunn Index*. Anche in tal caso, i cluster individuati non sembrano avere come discriminante principale il tempo.



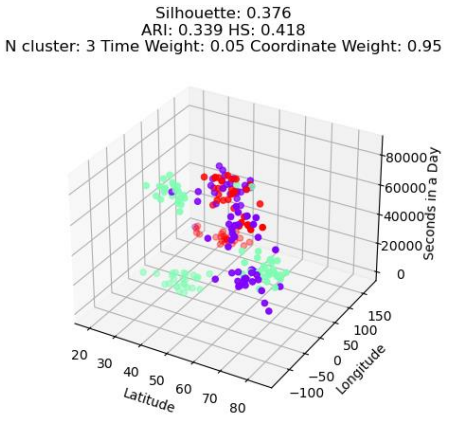


Figura 35: Lat, Long, Time – KMedoids con cluster identificato massimizzando la Silhouette (Dataset A) *Figura 36: Lat, Long, Time- KMedoids con cluster identificato massimizzando Dunn Index (Dataset A)*

5.2.1.1.4. Euclidean distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset A e selezionando come metrica di distanza spaziale l'*Euclidean distance.*

Rispetto ai risultati descritti nel paragrafo precedente (5.2.1.1.3) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 37) sono rappresentati i *cluster* ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 37) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index.* I diversi colori indicano l'appartenenza ad un *cluster*. In entrambi è stato ottenuto uno stesso risultato, identificando 4 *cluster*. Inoltre, tale risultato è analogo a quello descritto nel paragrafo 5.2.1.1.2, ottenuto a parità di condizioni ma utilizzando come metrica di distanza spaziale la *Euclidan distance* piuttosto che l'*Haversine.*

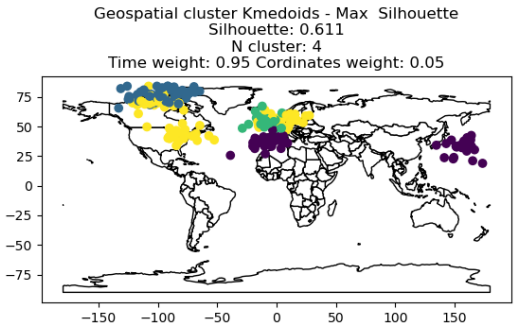


Figura 37:K-Medoids con cluster identificato massimizzando la Silhouette (Dataset A)

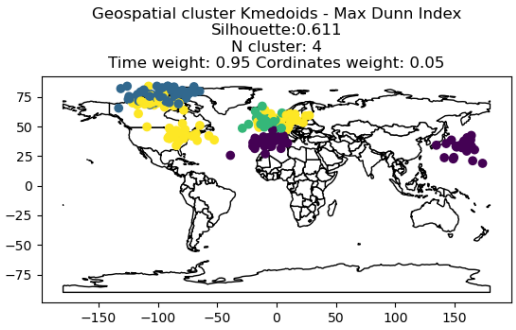


Figura 38: K-Medoids con n cluster identificato massimizzando il Dunn Index (Dataset A)

Per completezza si riportano anche i grafici 3D (Figura 39 e Figura 39) relativi ai cluster ottenuti con la massimizzazione della *Silhouette.* Si nota che i risultati sono analoghi a quelli ottenuti nel paragrafo 5.2.1.1.2. Per maggiori dettagli fare riferimento a tale paragrafo.

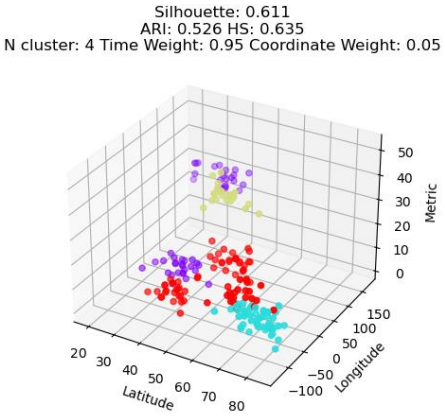


Figura 39: Lat, Long, Matric - KMedoids (Dataset A)

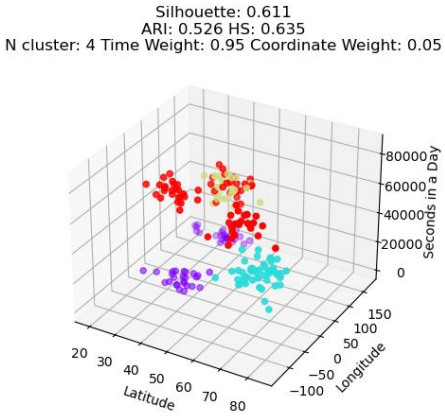


Figura 40: Lat, Long, Time - KMedoids (Dataset A)

In conclusione, dalla *clusterizzazione* del Dataset A è emerso che:

1) la metrica di distanza spaziale (*Euclidean* o *Haversine*) porta a cluster diversi, a parità di tutte le altre condizioni, solo in caso di basso peso della componente temporale $\boldsymbol{w_T = 0.05}$ (per maggiori dettagli fare riferimento ai paragrafi 5.2.1.1.1 e 5.2.1.1.3).
   In caso di alto peso della componente temporale $\boldsymbol{w_T = 0.95,}$ riducendo l'influenza dello spazio, la metrica di distanza spaziale (*Euclidean* o *Haversine*) non è una discriminante per la determinazione del *cluster.*

2) In particolare, nel caso di differenza, si ottengono *cluster* maggiormente coerenti e in linea con il peso fissato $\boldsymbol{w_T = 0.05}$ (quindi $\boldsymbol{w_S = 0.95}$) utilizzando l'*Haversine distance* (per maggiori dettagli fare riferimento ai paragrafi 5.2.1.1.1 e 5.2.1.1.3).
   Infatti, la differenza principale tra i due risultati è che con *l'Euclidean distance non* viene isolato il cluster relativo agli *ST Events* in prossimità degli Stati Uniti, al contrario del *cluster* ottenuto selezionando *l'Haversine.* Tale aspetto è compatibile con il fatto che l'*Euclidean distance* approssima la distanza tra 2 punti sulla Terra ad un segmento, riducendone di conseguenza tala distanza.

3) Utilizzando la distanza euclidea e $\boldsymbol{w_T = 0.05}$, si ottengono cluster diversi se si sceglie di determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index*. In tutti gli altri casi, non c'è differenza.

5.2.1.2. Dataset B – ST events: K-Medoids

Il Dataset B è un dataset sintetico costituito da 200 punti (*ST-Events*) localizzati nell'emisfero boreale e concetrati nell'Oceano Pacifico a ovest del Canada, nell'Oceano Pacifico a ovest della California, nel Nord Ovest dell'Africa e nel Nord Est della Russia. Dalla rappresentazione geografica dei punti di seguito riportata emerge che, in linea generale, i punti si estendono in zone della Terra meno vaste e più concentrate rispetto al Dataset A; in tal caso, la componente spaziale sembra essere maggiormente significativa. Considerando la sola distribuzione geografica, si possono identificare 4 macro aree: Oceano Pacifico a ovest del Canada, Oceano Pacifico a ovest della California, Nord Ovest dell'Africa e Nord Est della Russia.

Considerando invece la componente temporale, dai grafici 3D che mostrano Latitudine, Longitudine e Secondi in un giorno si possono individuare 2 macrofasce temporali: tra 0s (ore 00:00) e 20'000 s (circa ore 05:30), tra 20'000 s e 86'400 s (circa ore 23:59).

Considerando invece la componente metrica, dai grafici 3D che mostrano Latitudine, Longitudine e Metrica si possono individuare 2 macrofasce: una con metrica maggiore di 25, l'altra con metrica inferiore a 25.

5.2.1.2.1. Haversine distance con peso w_T =0.05

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset B e selezionando come metrica di distanza spaziale l'*Haversine.*

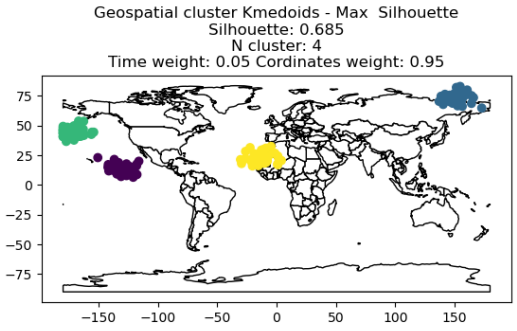
Inoltre, i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.05}$ (con prevalenza dello spazio) sono stati confrontati con quelli ottenuti fissando tale peso a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 41) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 41) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index.*

I diversi colori indicano l'appartenenza ad un cluster. In entrambi è stato ottenuto uno stesso risultato, identificando 4 *cluster*. Analogamente a quanto rilevato per il Dataset A, anche per il Dataset B la scelta dell'indice come criterio di identificazione del numero di cluster non è rilevante.

Avendo fissato come peso della componente spaziale $\boldsymbol{w_S = 0.95}$, tale dimensione prevale rispetto al tempo. Infatti, dai grafici di seguito emerge che sono stati identificati 4 *cluster* ben distinti, coincidenti con le 4 macroaree geografiche

È stato ottenuto un indice di *Silhouette* pari a 0.685.



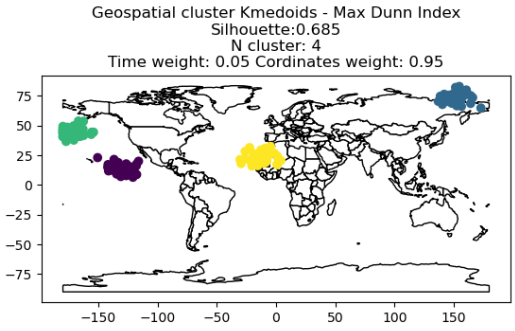


Figura 41: K-Medoids con n cluster identificato massimizzando la Silhouette (Dataset B) *Figura 42:* K-Medoids *con n cluster identificato massimizzando il Dunn Index (Dataset B)*

Poiché non è stata rilevata nessuna differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* per evitare la duplicazione di grafici, di seguito si riportano solo i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette.*

In particolare, sulla sinistra (Figura 43) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric. Si nota che, avendo un peso sempre unitario, la metrica è una componente discriminante per l'individuazione dei cluster (si noti ad esempio la distinzione in cluster viola da quello azzurro). Al contrario, la distinzione tra il cluster in rosso e quello in marroncino non è dettata da differenze di metrica, bensì da differenza spaziali e temporali.

Sulla destra (Figura 43 Figura 24) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno).

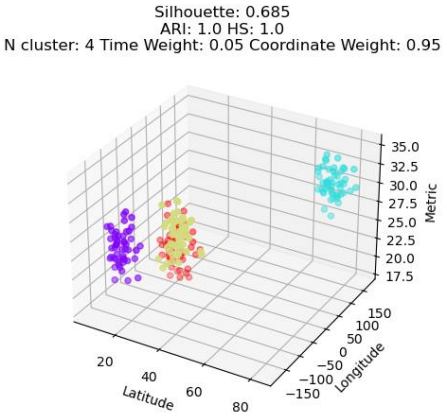


Figura 43:3D Lat, Long, Metric - KMedoids (Dataset B)

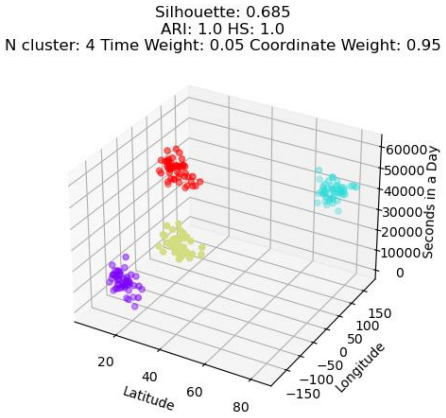


Figura 44: 3D Lat, Long, Time - KMedoids (Dataset B)

5.2.1.2.2. Haversine distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset B e selezionando come metrica di distanza spaziale l'*Haversine.*

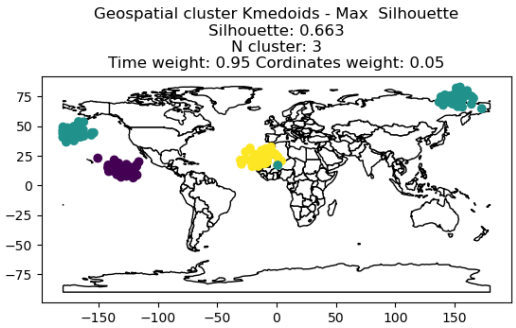
Rispetto ai risultati descritti nel paragrafo precedente (5.2.1.2.1) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 45) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 45) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index.*

I diversi colori indicano l'appartenenza ad un cluster. In entrambi è stato ottenuto uno stesso risultato, identificando 3 *cluster*: segue che la scelta dell'indice come criterio di identificazione del numero di cluster anche in tal caso non è rilevante. Rispetto al caso precedente, è stato individuato 1 cluster in meno.

Infatti, avendo fissato come peso della componente spaziale $\boldsymbol{w_S = 0.05}$, tale dimensione prevale di meno rispetto al caso precedente. Ad esempio, dai grafici di seguito emerge che, a differenza di prima, gli *ST-Events* relativi al Nord Est della Russia appartengono allo stesso *cluster* di quelli dell'Oceano Pacifico a Ovest del Canada.

In tal caso, è stato ottenuto un indice di *Silhouette* pari a 0.663, leggermente inferiore al *clustering* ottenuto nel paragrafo precedente (5.2.1.2.1 – con *Silhouette* pari a 0.658).



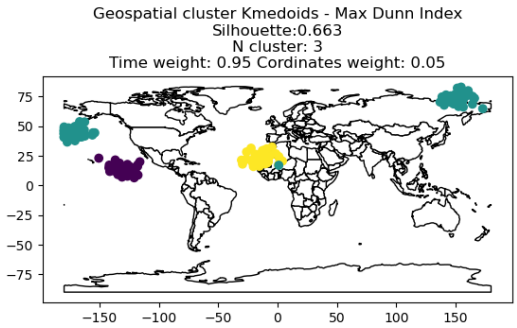


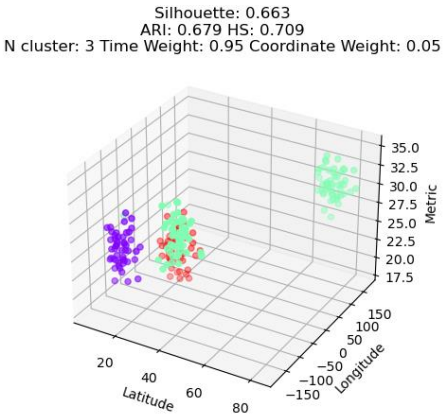
Figura 45: K-Medoids con n cluster identificato massimizzando la Silhouette (Dataset B) *Figura 46:* K-Medoids *con n cluster identificato massimizzando il Dunn Index (Dataset B)*

Anche in tal caso, poiché non è stata rilevata nessuna differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* per evitare la duplicazione di grafici, di seguito si riportano solo i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette.*

In particolare, sulla sinistra (Figura 47) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric.

Sulla destra (Figura 47) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno).

L'unica differenza rilevabile rispetto al caso precedente, come già sottolineato sopra, è che gli *ST-Events* relativi al Nord Est della Russia appartengono allo stesso *cluster* di quelli dell'Oceano Pacifico a Ovest del Canada. Questo perché si riduce l'importanza della componente spaziale e aumenta quella della componente temporale e metrica.



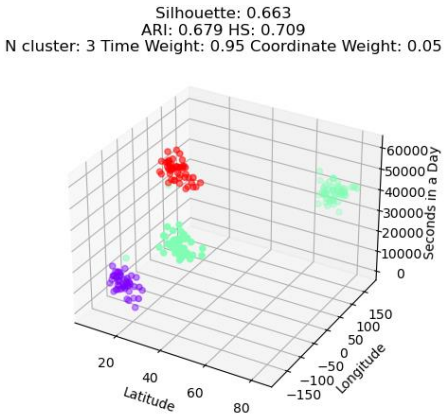


Figura 47:3D Lat, Long, Metric - KMedoids (Dataset B)

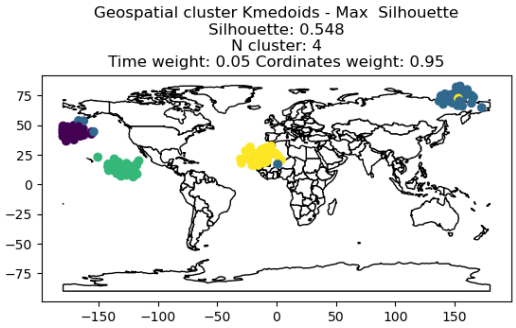
Figura 48: 3D Lat, Long, Time - KMedoids (Dataset B)

5.2.1.2.3. Euclidean distance con peso w_T =0.05

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset B e selezionando come metrica di distanza spaziale l'*Euclidean distance.* Tali risultati vengono confrontati con quelli riportati nel paragrafo 5.2.1.2.1 in cui viene scelta come metrica di distanza spaziale la *Haversine.*

In linea generale i risultati ottenuti sono gli stessi (a meno di qualche punto); inoltre, vengono individuati sempre 4 *cluster* sia se si sceglie di determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index.*

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 49) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 49) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index*.



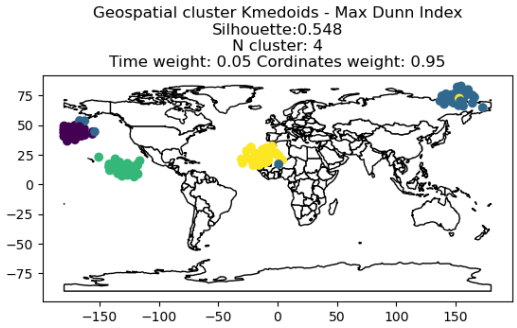


Figura 49: K-Medoids con n cluster identificato massimizzando la Silhouette (Dataset B) *Figura 50:* K-Medoids *con n cluster identificato massimizzando il Dunn Index (Dataset B)*

Poiché non è stata rilevata differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* di seguito si riportano i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette,* al fine di evitare duplicazioni di grafici.

In particolare, sulla sinistra (Figura 51) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metrica. I risultati ottenuti sono analoghi a quelli descritti nel paragrafo 5.2.1.2.1 (per maggiori dettagli fare riferimento al paragrafo).

Sulla destra (Figura 51) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno).

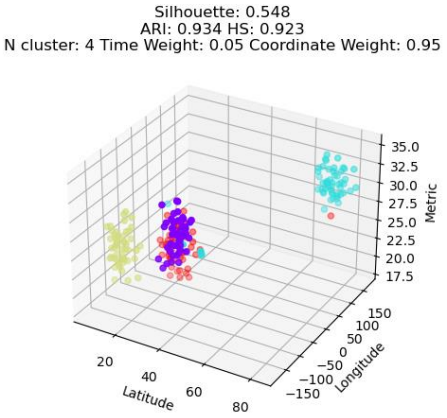


Figura 51:3D Lat, Long, Metric - KMedoids (Dataset B)

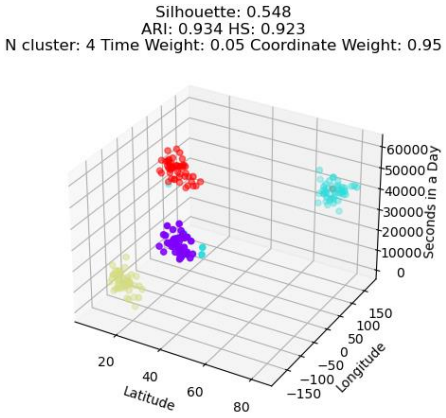


Figura 52: 3D Lat, Long, Time - KMedoids (Dataset B)

5.2.1.2.4. Euclidean distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *K-Medoids* al Dataset B e selezionando come metrica di distanza spaziale l'*Euclidean distance.*

Rispetto ai risultati descritti nel paragrafo precedente (5.2.1.2.3) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 53) sono rappresentati i cluster ottenuti sempre con il *K-Medoids* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 53) sono stati rappresentati i cluster ottenuti con il *K-Medoids* massimizzando il *Dunn Index*.

I diversi colori indicano l'appartenenza ad un *cluster*. In entrambi è stato ottenuto uno stesso risultato, identificando 3 *cluster*. Inoltre, tale risultato è analogo a quello descritto nel paragrafo 5.2.1.2.2, ottenuto a parità di condizioni ma utilizzando come metrica di distanza spaziale la *Euclidan distance* piuttosto che l'*Haversine.*

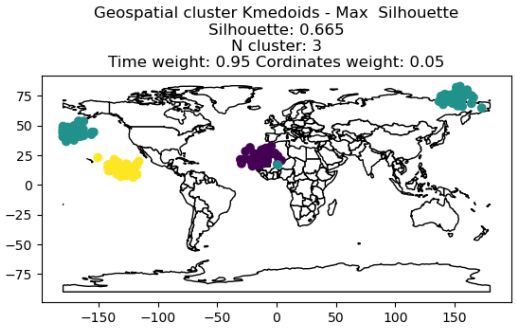


Figura 53: K-Medoids con n cluster identificato massimizzando la Silhouette (Dataset B)

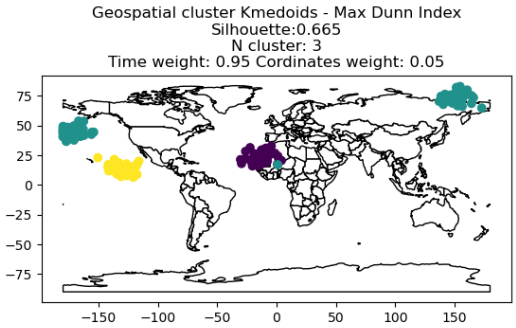


Figura 54: K-Medoids *con n cluster identificato massimizzando il Dunn Index (Dataset B)*

In particolare, sulla sinistra (Figura 55) si riporta il grafico 3D degli ST-Events rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric. I risultati ottenuti sono analoghi a quelli descritti nel paragrafo 5.2.1.2.2 (per maggiori dettagli fare riferimento al paragrafo).

Sulla destra (Figura 55) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno).

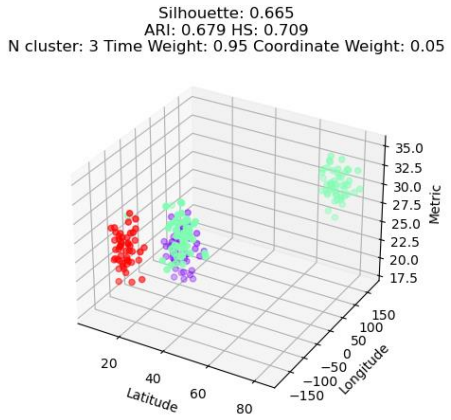


Figura 55:3D Lat, Long, Metric - KMedoids (Dataset B)

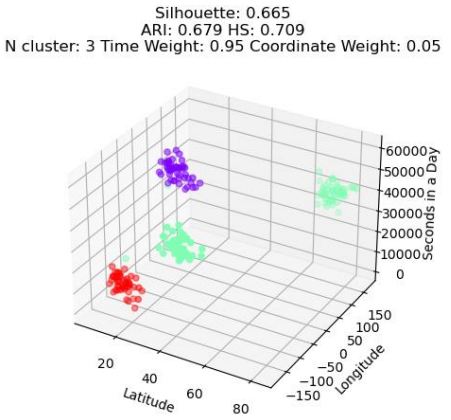


Figura 56: 3D Lat, Long, Time - KMedoids (Dataset B)

In conclusione, dalla *clusterizzazione* del Dataset B è emerso che:

1) le metriche di distanza spaziale *Euclidean* e *Haversine* generano uno stesso risultato di *clustering*, a parità di tutte le altre condizioni.

2) si ottiene uno stesso risultato di clustering determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index.*

Di conseguenza, per il cluster dei datatset reali *SF Park* e *Bikesharing* è stata considerata come sola metrica di distanza spaziale la *Haversine distance,* mentre il numero di cluster è stato determinato in modo da massimizzare il solo indice di *Silhouette* (non considerando il *Dunn Index*).

5.2.2. Implementazione Agglomerative

L'algoritmo *Agglomerative* è un algoritmo di *clustering distance-based* di tipo gerarchico *bottom-up*. È stato implementato utilizzando il metodo di ***sklearn*** [38] al fine di *clusterizzare* il Dataset A e il Dataset B, precedentemente *clusterizzati* con *K-Medoids.* In tal modo, è possibile confrontare i risultati ottenuti dall'applicazione di 2 algoritmi di *clustering* diversi su Dataset uguali.

5.2.2.1. Dataset A – ST events: Agglomerative

5.2.2.1.1. Haversine distance con peso w_T =0.05

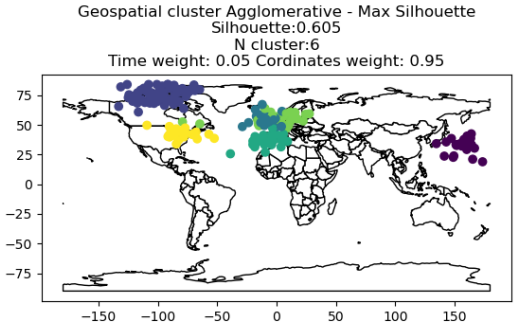
I due grafici di seguito riportati rappresentano gli *ST-Events* sul planisfero. Sulla destra (Figura 57) sono stati rappresentati i cluster ottenuti con *l'Agglomerative* massimizzando il Dunn Index mentre sulla sinistra (Figura 57) sono rappresentati i cluster ottenuti sempre con l'*Agglomerative* ma massimizzando l'indice di *Silhouette.*

I diversi colori indicano l'appartenenza ad un cluster. A differenza di quanto ottenuto con il *K-Medoids* (5.2.1.1.1) la scelta dell'indice come criterio di identificazione del numero di cluster è rilevante.

Se si sceglie come criterio l'indice di *Silhouette* si ottengono 6 cluster (Figura 57), che coincidono con i cluster individuati con il *K-Medoids* (paragrafo 5.2.1.1.1). Quindi in tal caso, non si rilevano differenze significative applicando i 2 algoritmi *Agglomerative* e *K-Medoids.*

È stato ottenuto un indice di *Silhouette* pari a 0.605, simile a quanto ottenuto a con il *K-Medoids* (0.558) nel paragrafo 5.2.1.1.1. Al contrario, se si sceglie come criterio il *Dunn Index* si identificano 2 cluster (Figura 57), i quali non sembrano essere discriminati in base alla componente spaziale (sebbene il peso della coordinata

spaziale sia pari a 0.95). Si ottiene infatti un basso indice di Silhouette (pari a 0.496). Invece, utilizzando il *K-Medoids*, a parità di tutti gli altri fattori, si identificano 6 cluster, ottenendo un indice di *Silhouette* di 0.558.



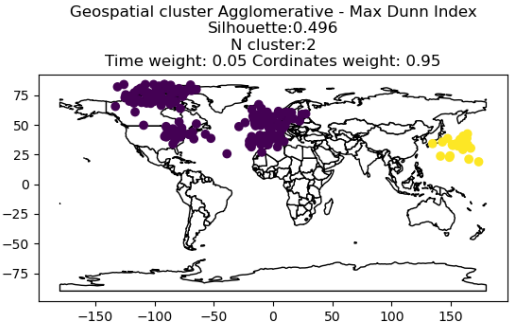
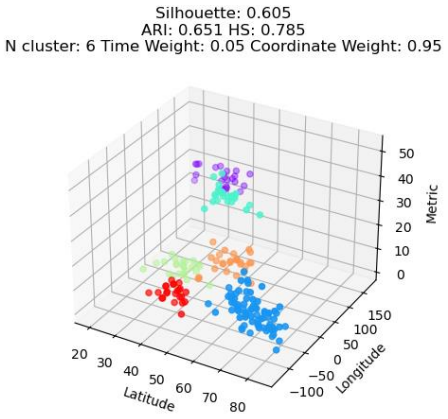


Figura 57: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset A) *Figura 58:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset A)*

Avendo fissato come peso della componente temporale $\boldsymbol{w_T = 0.05}$, tale dimensione prevale di meno rispetto allo spazio e alla metrica che ha sempre un valore unitario. Ad esempio, dai grafici di seguito relativi al cluster ottenuto massimizzando l'indice di *Silhouette* (Figura 59 e Figura 59Figura 61 e Figura 61), emerge che i *cluster* possono essere discriminati principalmente dai valori assunti dalla metrica.



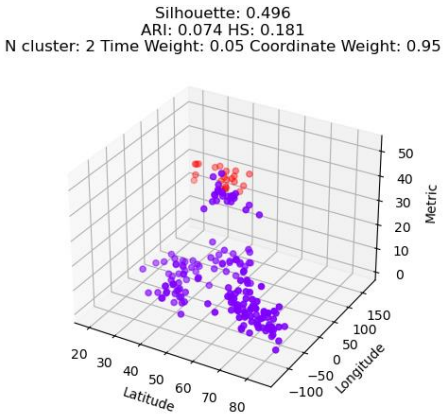
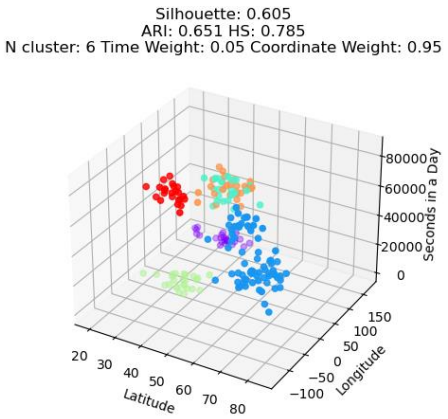


Figura 59: Lat,Long,Matric – Agglomerative , cluster identificato massimizzando la Silhouette (Dataset A) *Figura 60: Lat,Long,Metric-Agglomeratives, cluster identificato massimizzando Dunn Index (Dataset A)*



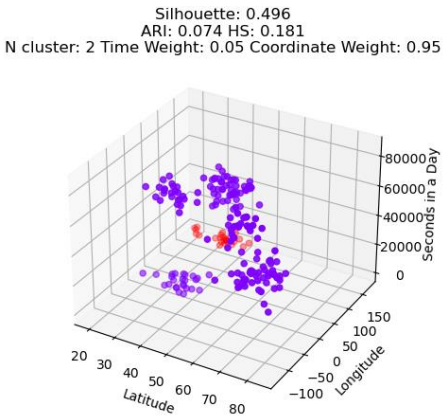


Figura 61: Lat,Long,Time – Agglomerative , cluster identificato massimizzando la Silhouette (Dataset A) *Figura 62: Lat,Long,Time-Agglomeratives, cluster identificato massimizzando Dunn Index (Dataset A)*

5.2.2.1.2. Haversine distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *Agglomerative* al Dataset A e selezionando come metrica di distanza spaziale l'*Haversine.*

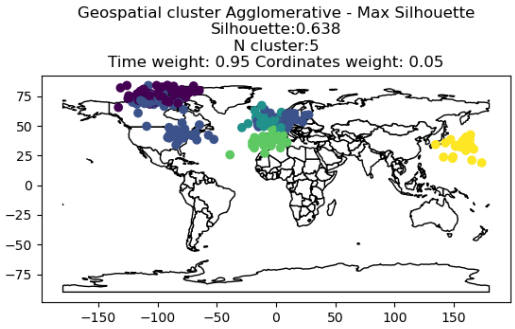
Rispetto ai risultati descritti nel paragrafo precedente (5.2.2.1.1) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla destra (Figura 63) sono stati rappresentati i cluster ottenuti con l'*Agglomerative* massimizzando il *Dunn Index* mentre sulla sinistra (Figura 63) sono rappresentati i cluster ottenuti sempre con *l'Agglomerative ma* massimizzando l'indice di *Silhouette.*

I diversi colori indicano l'appartenenza ad un cluster; si ottengono cluster diversi se si sceglie di determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index.*

Se si sceglie come criterio l'indice di *Silhouette* si ottengono 5 cluster (Figura 63 Figura 24) a differenza dei 4 *cluster* individuati applicando il *K-Medoids* (paragrafo 5.2.1.1.2). Il valore di *Silhouette* ottenuto è 0.638 (circa uguale al valore di *Silhouette* ottenuto con il *K-Medoids* a parità di altri parametri (pari a 0.612).

Se si sceglie come criterio il *Dunn Index* si identificano 3 cluster (Figura 63), a differenza dei 4 *cluster* individuati applicando il *K-Medoids* (paragrafo 5.2.1.1.2). Il valore di *Silhouette* ottenuto è 0.586 (inferiore ma circa uguale al valore di *Silhouette* ottenuto con il *K-Medoids* a parità di altri parametri, pari a 0.612).



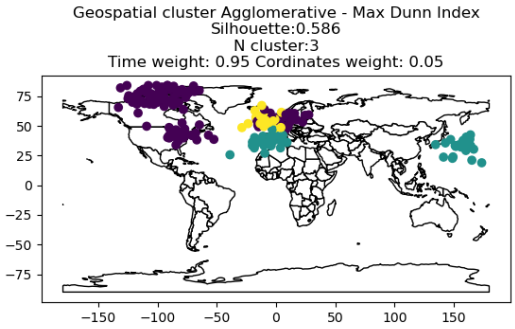
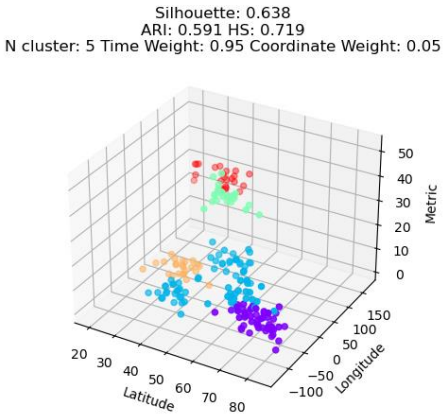


Figura 63: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset A) *Figura 64:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset A)*

Avendo fissato come peso della componente temporale $\boldsymbol{w_T = 0.95}$, tale dimensione prevale rispetto allo spazio, accanto alla metrica che ha sempre un valore unitario. Ad esempio, dai grafici di seguito (Figura 65 e Figura 65, Figura 67 e Figura 67) emerge che, i *cluster* possono essere discriminati dai valori assunti dalla metrica e dal tempo.



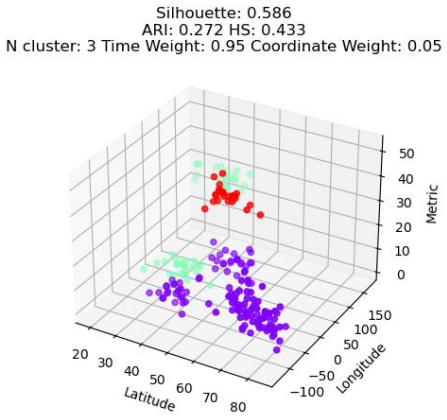
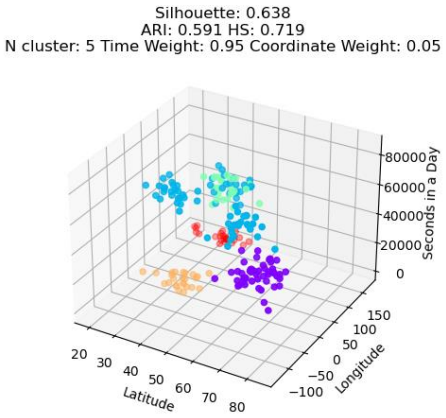


Figura 65: Lat,Long,Matric – Agglomerative , cluster identificato massimizzando la Silhouette (Dataset A) *Figura 66: Lat,Long,Metric-Agglomeratives, cluster identificato massimizzando Dunn Index (Dataset A)*



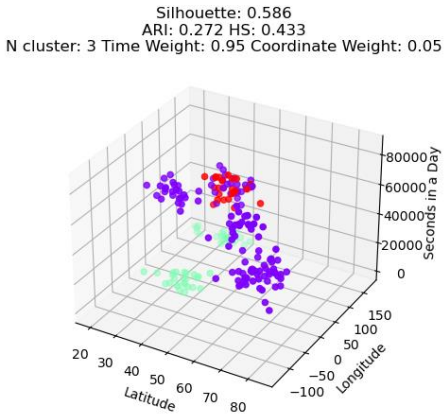
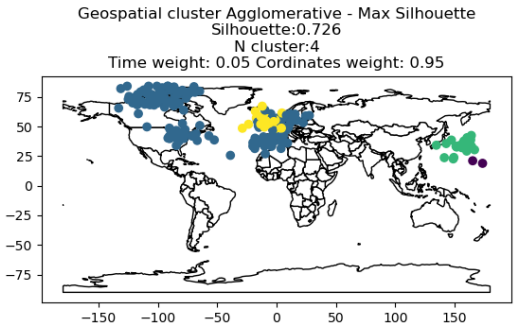


Figura 67: Lat,Long,Time – Agglomerative , cluster identificato massimizzando la Silhouette (Dataset A) *Figura 68: Lat,Long,Time-Agglomeratives, cluster identificato massimizzando Dunn Index (Dataset A)*

5.2.2.1.3. Eucidean distance con peso w_T =0.05

I due grafici di seguito riportati rappresentano gli *ST-Events* sul planisfero. Sulla sinistra (Figura 69) sono rappresentati i cluster ottenuti sempre con l'*Agglomerative* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 69) sono stati rappresentati i cluster ottenuti con *l'Agglomerative* massimizzando il Dunn Index.

I diversi colori indicano l'appartenenza ad un cluster. A differenza di quanto ottenuto con il *K-Medoids* (5.2.1.1.3) la scelta dell'indice come criterio di identificazione del numero di cluster non è rilevante.



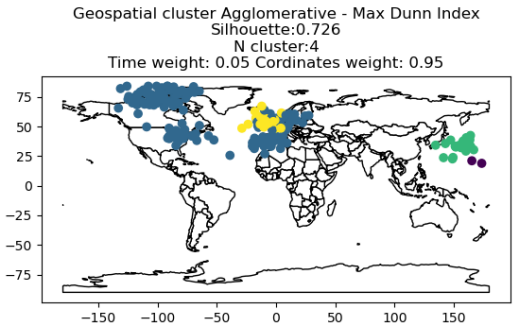
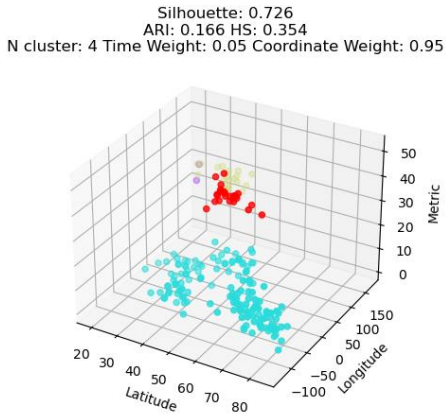


Figura 69: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset A) *Figura 70:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset A*

Poiché non è stata rilevata differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* di seguito si riportano i grafici 3D relativi ai cluster ottenuti con la massimizzazione della *Silhouette,* al fine di evitare duplicazioni di grafici.

In particolare, sulla sinistra (Figura 71) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Metric.

Sulla destra (Figura 71 Figura 24) si riporta il grafico 3D degli *ST-Events* rappresentati rispetto alle dimensioni Latitudine, Longitudine e Tempo (espresso in secondi in un giorno).



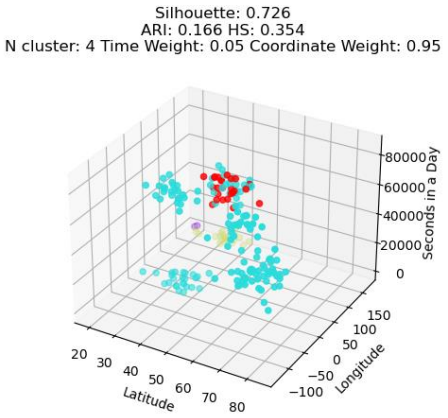


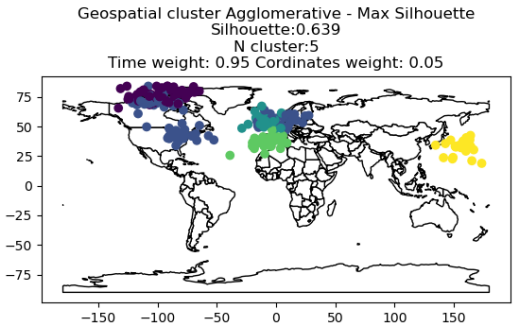
Figura 71:3D Lat, Long, Metric - Agglomerative (Dataset A)

Figura 72: 3D Lat, Long, Time – Agglomerative (Dataset A)

5.2.2.1.4. Euclidean distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *Agglomerative* al Dataset A e selezionando come metrica di distanza spaziale l'*Euclidean distance.* Tali risultati vengono confrontati con quelli riportati nel paragrafo 5.2.2.1.2 in cui viene scelta come metrica di distanza spaziale la *Haversine.* In linea generale i risultati ottenuti sono gli stessi, come emerge dalle figure: Figura 75, Figura 75, Figura 77, Figura 77.

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 73) sono rappresentati i cluster ottenuti sempre con l'*Agglomertive* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 73) sono stati rappresentati i cluster ottenuti con *l'Agglomerative* massimizzando il *Dunn Index*.



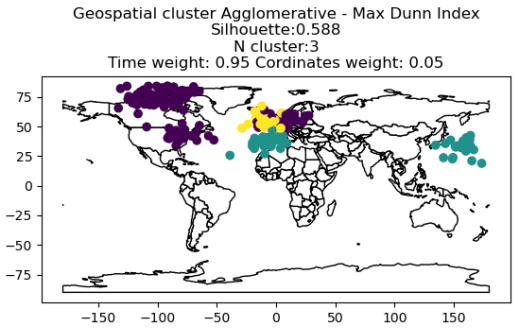
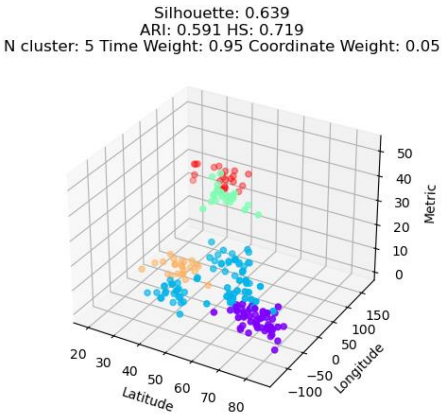


Figura 73: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset A) *Figura 74:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset A*



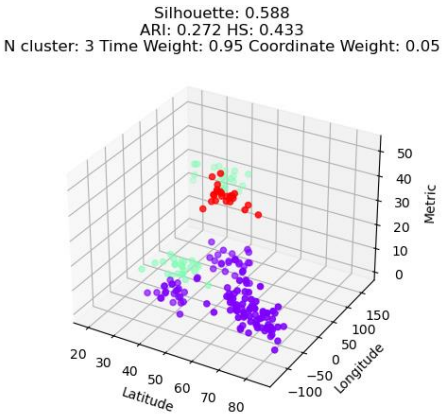
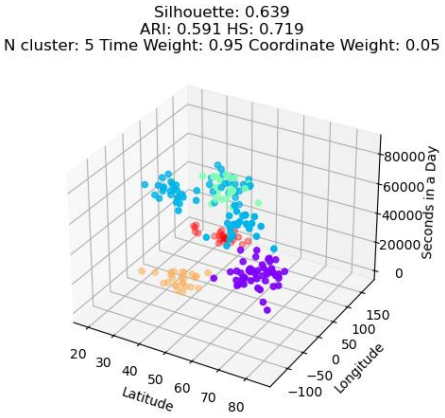


Figura 75: Lat,Long,Metric – Agglomerative cluster identificato massimizzando la Silhouette (Dataset A) *Figura 76: Lat Long, Metric- Agglomerative cluster identificato massimizzando Dunn Index (Dataset A)*



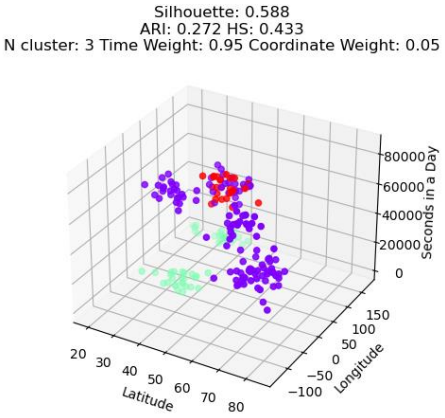


Figura 77: Lat, Long, Time – Agglomerative cluster identificato massimizzando la Silhouette (Dataset A) *Figura 78: Lat, Long, Time- Agglomerative cluster identificato massimizzando Dunn Index (Dataset A)*

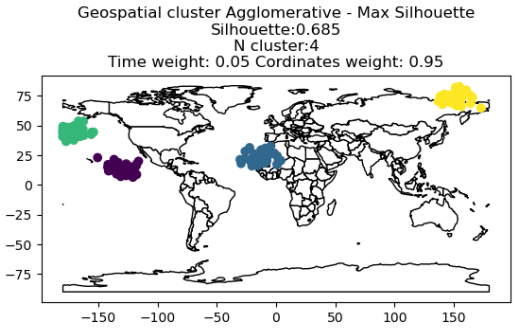
5.2.2.2. Dataset B – ST events like: Agglomerative

5.2.2.2.1. Haversine distance con peso w_T =0.05

I due grafici di seguito riportati rappresentano gli *ST-Events* sul planisfero. Sulla sinistra (Figura 79) sono rappresentati i cluster ottenuti sempre con l'*Agglomerative* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 79) sono stati rappresentati i cluster ottenuti con *l'Agglomerative* massimizzando il Dunn Index.

I diversi colori indicano l'appartenenza ad un cluster. A differenza di quanto ottenuto con il *K-Medoids* (5.2.1.2.1) la scelta dell'indice come criterio di identificazione del numero di cluster è rilevante. Se si sceglie come criterio l'indice di *Silhouette* si ottengono 4 cluster (Figura 79 Figura 24), che coincidono con i cluster individuati con il *K-Medoids* (paragrafo 5.2.1.2.1). Quindi in tal caso, non si rilevano differenze significative applicando i 2 algoritmi *Agglomerative* e *K-Medoids.* È stato ottenuto un indice di *Silhouette* pari a 0.685, che coincide con quello ottenuto applicando il *K-Medoids* nel paragrafo 5.2.1.2.1.

Al contrario, se si sceglie come criterio il *Dunn Index* si identificano 2 cluster (Figura 79), i quali non sembrano essere discriminati in base alla componente spaziale (sebbene il peso della coordinata spaziale sia pari a 0.95). Si ottiene infatti un più basso indice di Silhouette (pari a 0.589). Invece, utilizzando il *K-Medoids*, a parità di tutti gli altri fattori, si identificano 4 cluster, ottenendo un indice di *Silhouette* di 0.685.



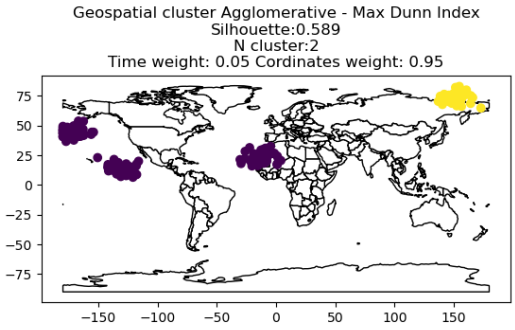
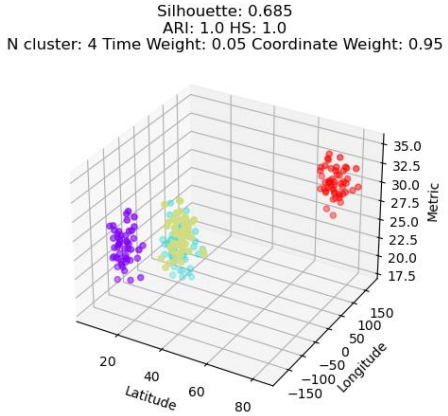


Figura 79: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset B) *Figura 80:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset B)*

Avendo fissato come peso della componente temporale $\boldsymbol{w_T = 0.05}$, tale dimensione prevale di meno rispetto allo spazio e alla metrica che ha sempre un valore unitario. Ad esempio, dai grafici di seguito riportati (Figura 81, Figura 81, Figura 83, Figura 83) emerge che i *cluster* possono essere discriminati dai valori assunti dalla metrica.



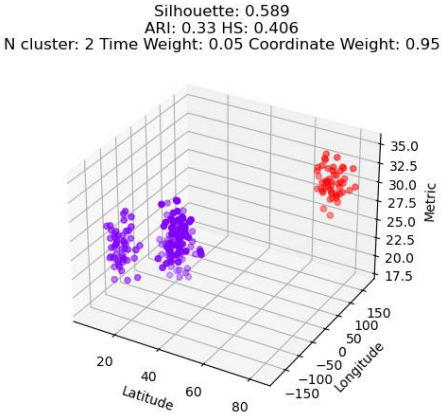
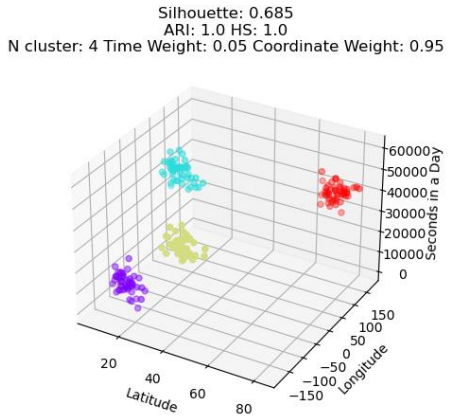


Figura 81: Lat,Long, Metric – Agglomerative cluster identificato massimizzando la Silhouette (Dataset B) *Figura 82: Lat,Long, Metric-Agglomerative cluster identificato massimizzando Dunn Index (Dataset B)*



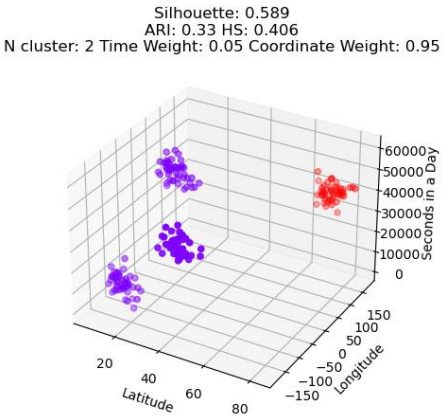


Figura 83: Lat, Long, Time – Agglomerative cluster identificato massimizzando la Silhouette (Dataset B) *Figura 84: Lat, Long, Time- Agglomerative cluster identificato massimizzando Dunn Index (Dataset B)*

5.2.2.2.2. Haversine distance con peso w_T =0.95

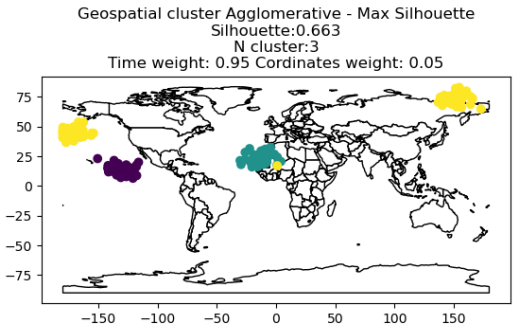
Di seguito si riportano i risultati ottenuti applicando l'algoritmo *Agglomerative* al Dataset B e selezionando come metrica di distanza spaziale l'*Haversine.*

Rispetto ai risultati descritti nel paragrafo precedente (5.2.2.2.1) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 85) sono rappresentati i cluster ottenuti sempre con *l'Agglomerative ma* massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 85) sono stati rappresentati i cluster ottenuti con l'*Agglomerative* massimizzando il *Dunn Index.*

I diversi colori indicano l'appartenenza ad un cluster; si ottengono cluster uguali se si sceglie di determinare il numero di cluster in modo automatico massimizzando l'indice di *Silhouette* piuttosto che il *Dunn Index*.

Nel caso specifico, si ottengono 3 cluster (Figura 87 e Figura 87) a differenza dei 4 *cluster* individuati applicando il *K-Medoids* (paragrafo 5.2.1.1.2). Il valore di *Silhouette* ottenuto è 0.663 (circa uguale al valore di *Silhouette* ottenuto con il *K-Medoids* a parità di altri parametri (pari a 0.612).



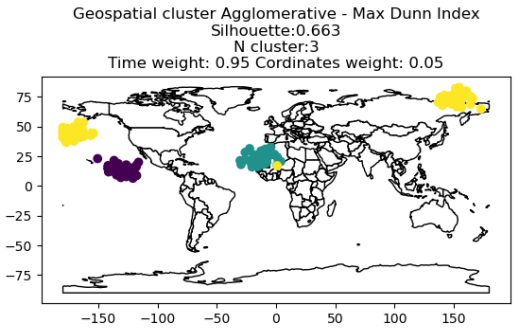


Figura 85: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset B) *Figura 86:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset B)*

Poiché non è stata rilevata differenza nella determinazione dei cluster massimizzando la *Silhouette* piuttosto che il *Dunn Index,* di seguito si riportano i grafici 3D (Figura 87 e Figura 87) relativi ai cluster ottenuti con la massimizzazione della *Silhouette,* al fine di evitare duplicazioni di grafici.

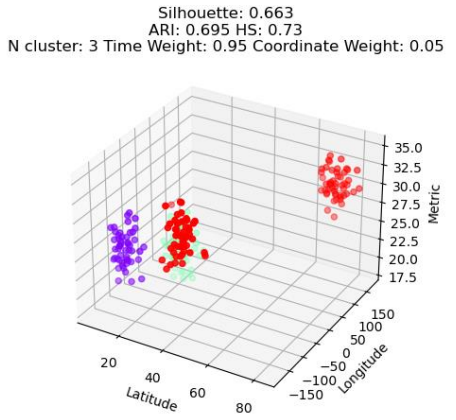


Figura 87:3D Lat, Long, Metric - Agglomerative (Dataset A)

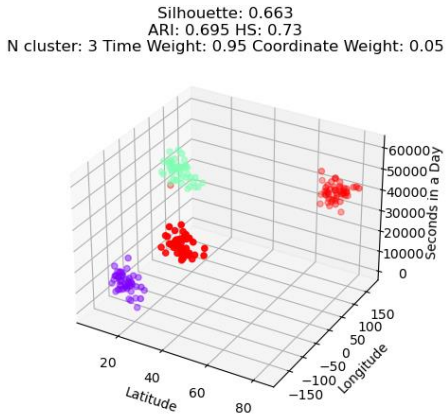


Figura 88: 3D Lat, Long, Time – Agglomerative (Dataset A)

5.2.2.2.3. Eucidean distance con peso w_T =0.05

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *Agglomerative* al Dataset B e selezionando come metrica di distanza spaziale l'*Euclidean distance.* Tali risultati vengono confrontati con quelli riportati nel paragrafo 5.2.2.2.1 in cui viene scelta come metrica di distanza spaziale la *Haversine.*

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 89) sono rappresentati i cluster ottenuti sempre con *l'Agglomerative ma* massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 89) sono stati rappresentati i cluster ottenuti con l'*Agglomerative* massimizzando il *Dunn Index*.

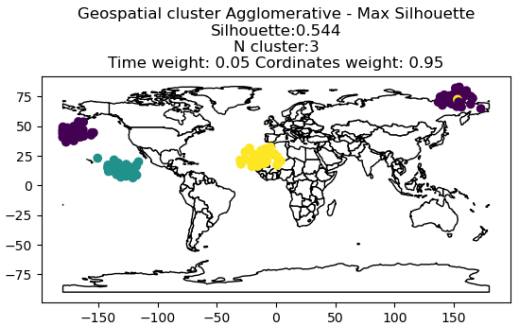
Analogamente a quanto rilevato nel paragrafo 5.2.2.2.1, la scelta dell'indice come criterio di identificazione del numero di cluster è rilevante.

In particolare, se si sceglie come criterio il *Dunn Index* si ottiene lo stesso risultato riportato nel paragrafo 5.2.2.2.1, dove vengono identificati 2 cluster.

Se si sceglie come criterio di identificazione del numero di cluster la massimizzazione dell'indice di *Silhouette,* il cluster *Agglomerative* con distanza spaziale *euclidea* (paragrafo 5.2.2.2.1) identifica 3 cluster (Figura 89, Figura 91, Figura 93) e un indice di *Silhouette* pari a 0.544.

Al contrario, il cluster *Agglomerative* con distanza spaziale *Haversine* (paragrafo 5.2.2.2.1) intercetta 4 cluster e indice di *Silhouette* pari a 0,685. La differenza fondamentale è che il cluster con la distanza spaziale euclidea non distingue in 2 cluster diversi la macroarea di *ST-Events* in prossimità del Nord-Est della Russia e la macroarea a Ovest del Canada in prossimità dell'Oceano Pacifico.

Se si sceglie come criterio di identificazione del numero di cluster la massimizzazione del Dunn Index, si ottengono 2 cluster (Figura 89, Figura 91, Figura 93), in analogia al risultato ottenuto nel paragrafo 5.2.2.2.1, dove è stata utilizzata la distanza *Haversine.*



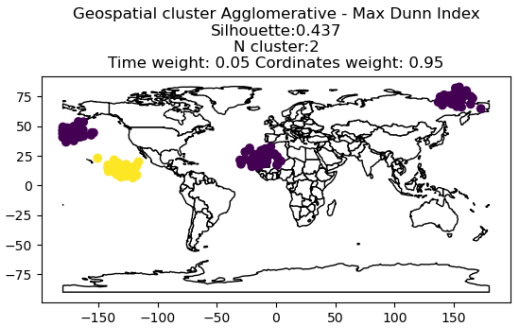
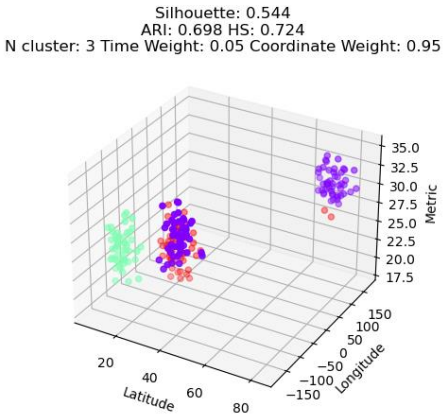


Figura 89: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset B) *Figura 90:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset B)*



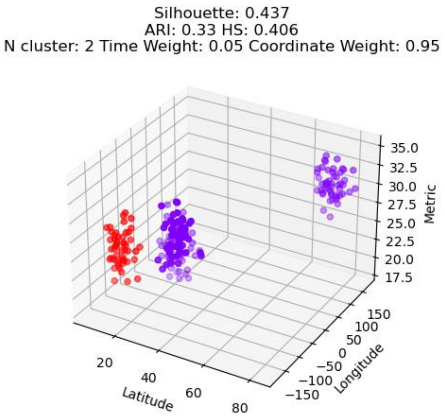
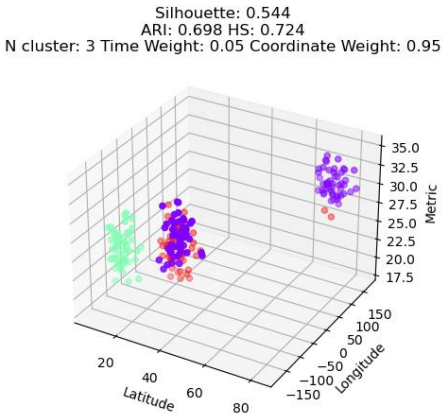


Figura 91: Lat,Long,Metric – Agglomerative cluster identificato massimizzando la Silhouette (Dataset B) *Figura 92: Lat Long, Metric- Agglomerative cluster identificato massimizzando Dunn Index (Dataset B)*



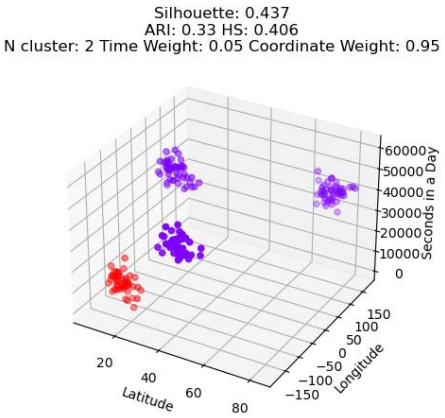


Figura 93: Lat,Long,Metric – Agglomerative cluster identificato massimizzando la Silhouette (Dataset B) *Figura 94: Lat Long, Metric- Agglomerative cluster identificato massimizzando Dunn Index (Dataset B)*

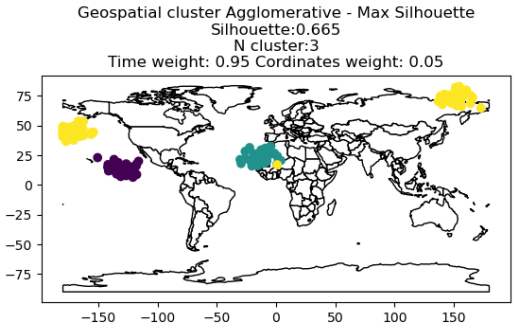
5.2.2.2.4. Euclidean distance con peso w_T =0.95

Di seguito si riportano i risultati ottenuti applicando l'algoritmo *Agglomerative* al Dataset B e selezionando come metrica di distanza spaziale l'*Euclidean distance.*

Rispetto ai risultati descritti nel paragrafo precedente (5.2.2.2.3) in cui si clusterizza considerando come prevalente la componente spaziale, di seguito si riportano i risultati ottenuti fissando il peso della componente temporale pari a $\boldsymbol{w_T = 0.95}$ (con prevalenza del tempo).

I due grafici di seguito riportati rappresentano gli *ST Events* sul planisfero. Sulla sinistra (Figura 95) sono rappresentati i cluster ottenuti sempre con l'*Agglomerative* ma massimizzando l'indice di *Silhouette* mentre sulla destra (Figura 95) sono stati rappresentati i cluster ottenuti con *l'Agglomerative massimizzando* il *Dunn Index*.

I diversi colori indicano l'appartenenza ad un cluster. In entrambi è stato ottenuto uno stesso risultato, identificando 3 *cluster* (Figura 97 e Figura 97). Inoltre, tale risultato è analogo a quello descritto nel paragrafo 5.2.2.2.2, ottenuto a parità di condizioni ma utilizzando come metrica di distanza spaziale la *Euclidian distance* piuttosto che l'*Haversine.*



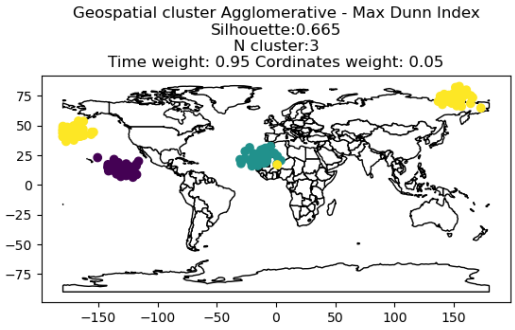
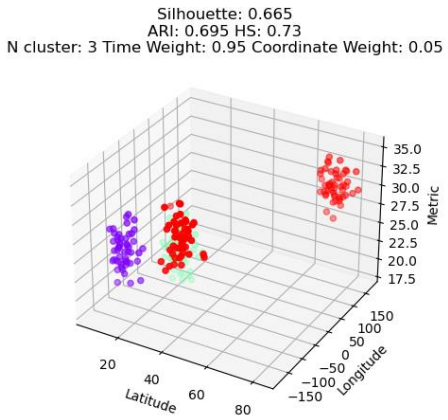


Figura 95: Agglomerative con n cluster identificato massimizzando Silhouette (Dataset B) *Figura 96:* Agglomerative *con n cluster identificato massimizzando Dunn Index (Dataset B)*



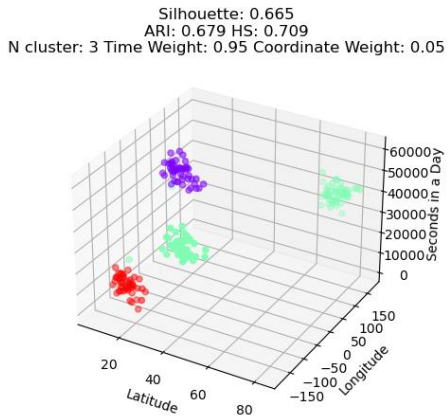


Figura 97:3D Lat, Long, Metric - Agglomerative (Dataset B))

Figura 98: 3D Lat, Long, Time – Agglomerative (Dataset B)

In conclusione è possible affermare che:

1) i risulati ottenuti con il cluster agglomerative sono maggiormente soggetti al criterio di scelta del numero di cluster (scegliendo se massimizzare l'indice di *Silhouette* o il *Dunn Index*);
2) in generale si ottengono dei valori di *Silhouette* maggiori utilizzando la metrica di distanza *Haversine*
3) non esiste una combinazione ottimale di parametri tali per cui viene sempre massimizzato un indice di cluster (in questo caso la *Silhouette*).
   La giusta combinazione dei parametri dipende dalla tipolgia di dataset e dagli obiettivi del *data analyst.*

Di seguito si riporta la Tabella 6 riassuntiva dei risultati ottenuti e descritti nei paragrafi precedenti:

Tabella 6:Risultati delle simulazioni

DATASET	Algoritmo	Metrica di distanza	Time weight	n cluster max Silhouette	Silhouette con n cluster max Silhouette	n cluster max Dunn Index	Silhouette con n cluster max Dunn Index
A	K-Medoids	*Haversine*	*0,05*	*6*	*0,558*	*6*	*0,558*
A	K-Medoids	Haversine	0,95	4	0,612	4	0,612
A	K-Medoids	Euclidean	0,05	5	0,431	3	0,376
A	K-Medoids	Euclidean	0,95	4	0,611	4	0,611
B	K-Medoids	Haversine	0,05	4	0,685	4	0,685
B	K-Medoids	Haversine	0,95	3	0,663	3	0,663
B	K-Medoids	Euclidean	0,05	4	0,548	4	0,548
B	K-Medoids	Euclidean	0,95	3	0,665	3	0,665
A	Agglomerative	Haversine	0,05	6	0,605	2	0,496
A	Agglomerative	Haversine	0,95	5	0,638	3	0,586
A	Agglomerative	Euclidean	0,05	4	0,726	4	0,726
A	Agglomerative	Euclidean	0,95	5	0,639	3	0,588
B	Agglomerative	Haversine	0,05	4	0,685	2	0,589
B	Agglomerative	Haversine	0,95	3	0,663	3	0,663
B	Agglomerative	Euclidean	0,05	3	0,544	2	0,437
B	Agglomerative	Euclidean	0,95	3	0,665	3	0,665

5.2 Implementazione ST-DBSCAN

L'algoritmo **ST-DBSCAN** [39] (***Spatio Temporal DBSCAN***) è un'estensione dell'algoritmo DBSCAN, il quale viene adattato a dati spazio-temporali; infatti, separa le informazioni "spaziali" da quelle "non-spaziali", individuando due metriche di distanza separate. ST-DBSCAN rispetto al tradizionale DBSCAN, introduce un secondo parametro del raggio di prossimità, ovvero quello temporale $\boldsymbol{\varepsilon_2}$, oltre a quello spaziale $\boldsymbol{\varepsilon_1}$**.**

Tale algoritmo è stato implementato utilizzando il metodo "***ST_DBSCAN***" [40], che richiede in input i seguenti parametri:

- **Min points:** numero minimo di punti *samples* richiesti nell'intorno di un punto affinché venga considerato core.
- **Eps 1:** massima distanza tra le componenti **spaziali,** ed eventuali **metriche**, di due punti affinché vengano considerati appartenenti ad uno stesso cluster (*spatial density threshold*).
- **Eps 2:** massima distanza tra le componenti **temporali** di due punti affinché vengano considerati appartenenti ad uno stesso cluster (*temporal density threshold)*

L'approccio tradizionale si basa sull'utilizzo del grafico chiamato ***k − dist,*** in cui si visualizzano i punti ordinati in ordine crescente in base alla distanza dal loro k − esimo punto, dove k è specificato in input. Il principio su cui si basa il *k − dist plot* consiste nel fatto che per i core points i k − esimi punti più vicini saranno indicativamente alla stessa distanza, mentre i *noise points* avranno il k − esimo punto più lontano. Di conseguenza, quando nella curva si verifica un cosiddetto "*gomito*" (da cui il nome della funzione ***find elbow***), ovvero un repentino cambio della distanza questo segnala la separazione tra core e noise points. Quindi, il punto in corrispondenza del quale si verifica il cambio della pendenza coincide con un valore adatto di Eps. Di conseguenza, selezionando l'Eps così identificato e assumendo Minpoints pari a k, i punti che nel k − dist hanno un'ordinata minore di Eps sono etichettati come core, mentre gli altri sono etichettati come border o noise points.

Nel caso specifico, vengono calcolati due ***eps:***

- ***Eps1,*** che tiene conto delle coordinate spaziali ed eventuali metriche

- ***Eps2,*** che tiene conto della coordinate temporale

Per deteriminare ***Eps1*** e ***Eps2*** è stato dunque necessario inanzitutto isolare le dimensioni di interesse e poi, separatamente per ogni vettore sono stati calcolati *Minpoints* ed *Eps.*

In particolare, è stato prima calcolato il ***Minpoints****,* con il modulo "**minpts**" implementato in ADESCA; di seguito si descrive l'algoritmo che porta alla determinazione del "**minpts**".

All'interno di un ciclo *for,* viene calcolato il vettore distanza tra un punto del dataset dato in input e il suo *i-esimo* vicino, utilizzando il modulo ***NearestNeighbors*** di ***sklearn.neighbors*** [41]***.***

Successivamente, viene calcolato il vettore distanza tra il punto in questione e il suo *i-esimo+1* vicino.

A questo punto, viene calcolato l'incremento percentuale tra i due vettori distanza precedentemente definiti. Gli incrementi di ogni dimensione sono salvati in una variabile detta "*sum*". Infine la somma degli incrementi viene rapportata alla lunghezza del vettore distanza, ovvero alla dimensione del punto.

Se tale incremento è inferiore al 3% allora l'*i-esimo* vicino viene fissato come *Minpoints.* Questo perchè, la differenza tra le distanze dell'*i-esimo* vicino e dell'*i-esimo + 1* vicino è trascurabile (inferiore al 3%).

Altrimenti, il ciclo viene iterato.

Una volta fissato Minpoints viene calcolata la distanza tra ogni punto e il suo k-esimo vicino, utilizzando il metodo ***NearestNeighbors*** di ***sklearn.neighbors*** [41]. Il metodo ***NearestNeighbors*** restituisce due array, uno che contiene la distanza dai punti più vicini e l'altro che contiene l'indice per ciascuno di quei punti.

Successivamente le distanze sono ordinate in senso decrescente e viene trovato il punto minimo della curva con la funzione ***find_elbow.*** L'indice corrispondente al punto minimo così determinato corrisponde all'epsilon (sia ***Eps1*** che ***Eps2***).

Infine, il *Minpoints* è stato scelto come il minimo tra i due minpoints precedentemente determinati:

A questo punto viene clusterizzato il dataset normalizzato, utilizzando il metodo "***ST_DBSCAN***" [40].

Si riportano due esempi di grafici *k-dist* generati con l'algoritmo appena descritto, applicato a due dataset *ST-Events* simulate (dataset A e dataset B).

5.3.1. Dataset A – ST events like: ST-DBSCAN

La Figura 99 mostra il *k-dist* generto per il Dataset A:

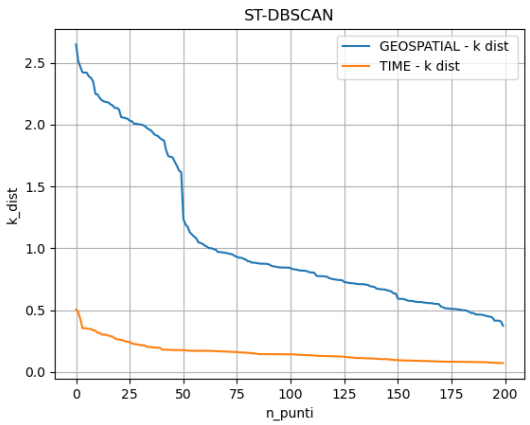


Figura 99: k-Dist plot per Dataset A

Nella Tabella 7 di seguito si riportano i parametric fissati e gli indici:

Tabella 7: K-Dist Dataset A

eps1	eps2	min point	cluster number	Silhouette	DUNN INDEX
1,049	0,182	27	3	0,202	0,034

I grafici 3D riportati di seguito (Figura 100 e Figura 100) mostrano il risultato del *cluster* ottenuto applicando ST-DBscan al Dataset A.

Si nota che vengono identificati 3 cluster, di cui:

1) il cluster in rosso sembra essere caratterizzato da valori bassi di metrica e alti valori di tempo.
2) il cluster in turchese sembra essere concentrato in una specifica zona spaziale e caratterizzato da bassi valori di metrica e temporali
3) il cluster in viola è meno denso.

Con l'ST-DBscan si ottiene un valore relativamente basso di *Silhouette* (pari a 0.202) se paragonato al valore di *Silhouette* ottenuto con l'*Agglomerative clustering* o con il *K-Medoids.*

Il dataset A infatti è meno denso, dunque non risulta molto adatto ad essere clusterizzato con algoritmi appartenenti alla famiglia *density-based* (come il ST-DBSCAN).

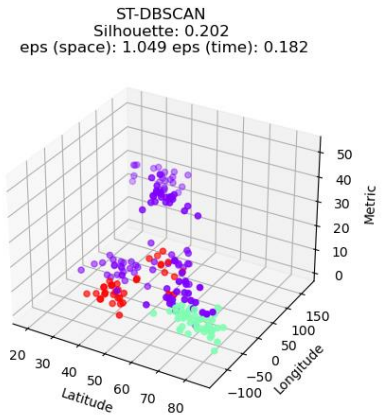


Figura 100 Lat, Long, Metric; ST-DBscan, Dataset A

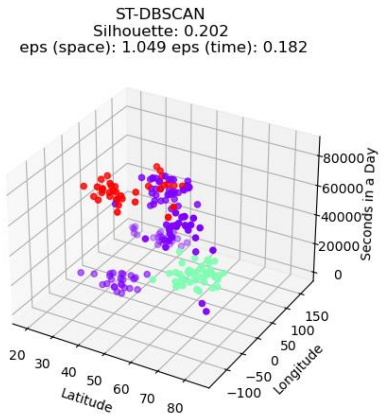


Figura 101: Lat, Long, Time; ST-DBscan, Dataset A

5.3.2. Dataset B – ST events like: ST-DBSCAN

La Figura 102 mostra il *k-dist* generto per il Dataset B:

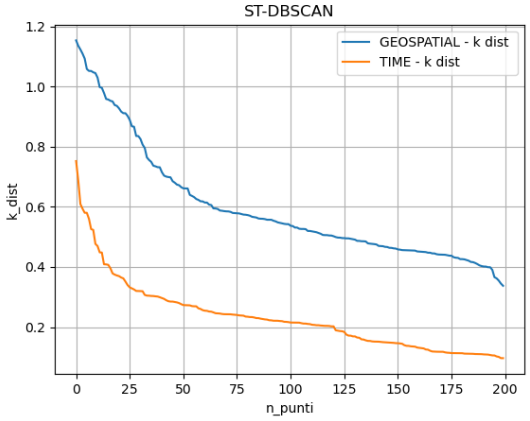


Figura 102: K-Dist plot per Dataset B

Nella Tabella 8 di seguito si riportano i parametric fissati e gli indici:

Tabella 8: K-Dist per Dataset B

eps1	eps2	min point	cluster number	Silhouette	DUNN INDEX
0,626	0,307	25	5	0,547	0,027

I grafici 3D riportati di seguito (Figura 103 e Figura 103) mostrano il risultato del *cluster* ottenuto applicando ST-DBSCAN al Dataset B.

Si nota che vengono identificati 5 cluster, di cui 4 corrispondono ad aree particolamente dense, mentre il cluster viola sembra intercettare gli outlier di ogni cluster denso.

A differenza del Dataset A, il Dataset B è più adatto all'utilizzo del ST-DBSCAN, in quanto presenta zone maggiormente dense. Si ottiene un valore relativamente alto di *Silhouette* (pari a 0.547) se paragonato al valore di *Silhouette* ottenuto con l'*Agglomerative clustering* o con il *K-Medoids.*

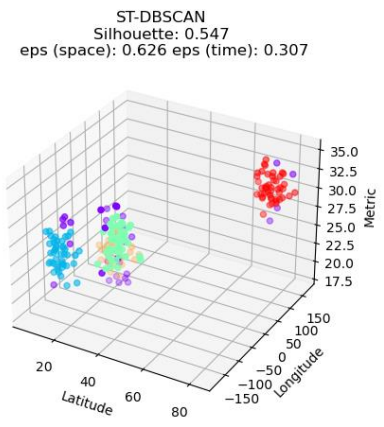


Figura 103: Lat, Long, Metric; ST-DBscan, Dataset B

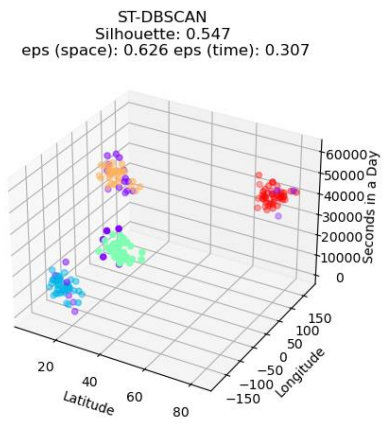


Figura 104: Lat, Long, Time; ST-DBscan, Dataset B

5.3.3. Dataset A: Analisi di sensitività - ST-DBSCAN

Al fine di valutare la sensitività dell'algoritmo ST-DBSCAN ai parametri ***Eps1*** e ***Eps2,*** tali parametri sono stati variati progressivamente in modo da valutare l'impatto di tale variazione sul cluster ottenuto.

In particolare, i dataset in analisi sono stati *ri-clusterizzati* fissando l'incremento di ***Eps*** del 50%, ovvero fissando:

a) $\boldsymbol{Eps1\_add\_delta = Eps1 + {Eps1}/{2}}$
b) $\boldsymbol{Eps1\_meno\_delta = Eps1 - {Eps1}/{2}}$
c) $\boldsymbol{Eps2\_add\_delta = Eps2 + {Eps2}/{2}}$
d) $\boldsymbol{Eps2\_meno\_delta = Eps2 - {Eps2}/{2}}$

Di seguito i risultati:

a) $\boldsymbol{Eps1\_add\_delta = Eps1 + Eps1/2}$

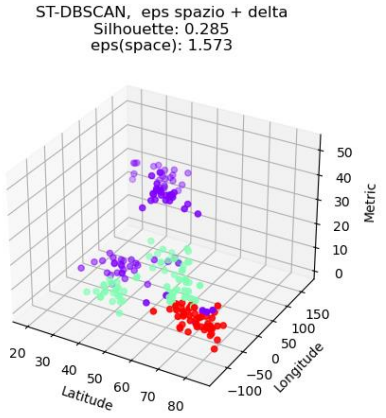


Figura 105 Lat, Long, Metric; ST-DBSCAN, Dataset A - Eps1+delta

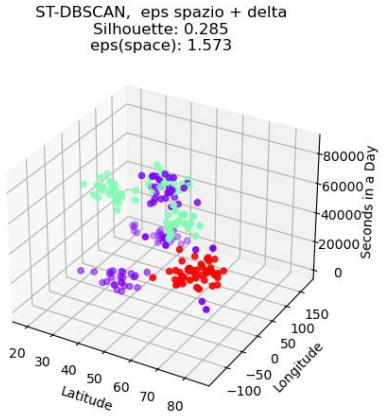


Figura 106 Lat, Long, Time; ST-DBSCAN, Dataset A - Eps1+delta

Aumentando $\boldsymbol{Eps1}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la Silhouette registra un leggero aumento (Figura 105 e Figura 105).

b) $\boldsymbol{Eps1\_meno\_delta = Eps1 - Eps1/2}$

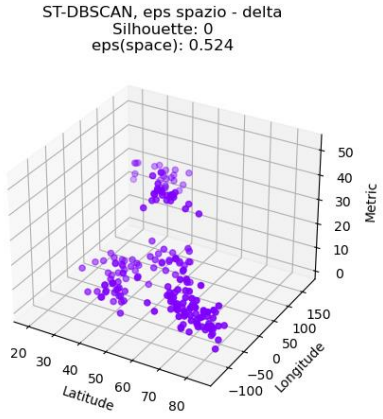


Figura 107 Lat, Long, Metric; ST-DBSCAN, Dataset A - Eps1-delta

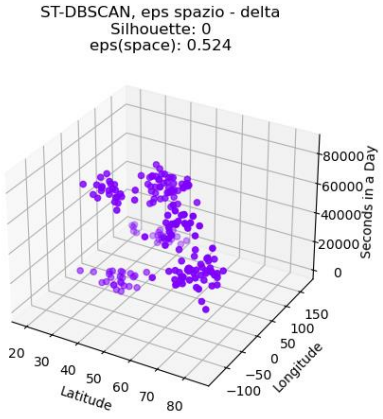


Figura 108 Lat, Long, Time; ST-DBSCAN, Dataset A - Eps1-delta

Riducendo $\boldsymbol{Eps1}$ di un valore pari al 50%, si ottiene una variazione significativa: il valore di $\boldsymbol{Eps1}$ si riduce così tanto che non viene identificato nessun *cluster* (Figura 107 e Figura 107).

c) $\boldsymbol{Eps2\_add\_delta = Eps2 + Eps2/2}$

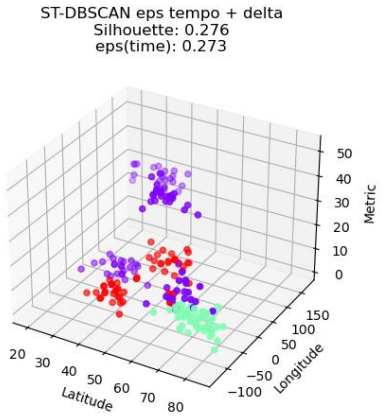


Figura 109 Lat, Long, Metric; ST-DBSCAN, Dataset A – Eps2+delta

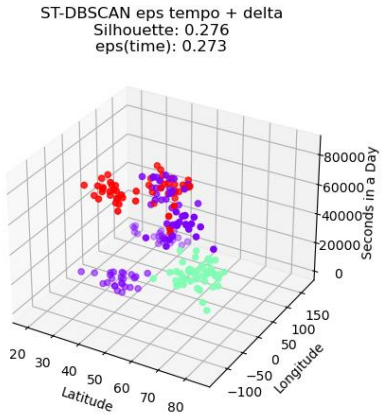


Figura 110 Lat, Long, Time; ST-DBSCAN, Dataset A – Eps2+delta

Aumentando $\boldsymbol{Eps2}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la *Silhouette* registra un leggero aumento (Figura 109 e Figura 109).

d) $\boldsymbol{Eps2\_meno\_delta = Eps2 - Eps2/2}$

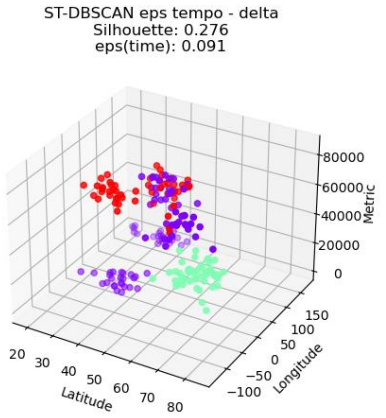


Figura 111 Lat, Long, Metric; ST-DBSCAN, Dataset A – Eps2-delta

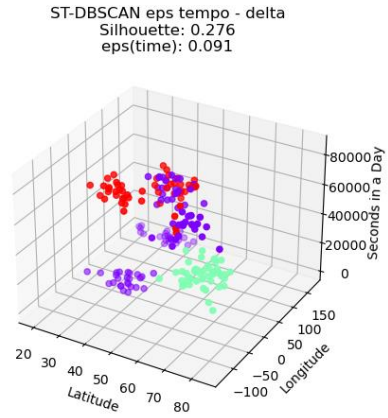


Figura 112 Lat, Long, Time; ST-DBSCAN, Dataset A – Eps2-delta

Riducendo $\boldsymbol{Eps2}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la *Silhouette* registra un leggero aumento (Figura 111 e Figura 111).

La Tabella 9 riassume i risuati ottenuti:

Tabella 9 Analisi di sensibilità – ST-DBSCAN (Dataset A)

	eps1	eps2	min point	cluster number	Silhouette	DUNN INDEX
0	1,048849	0,181756	27	3	0,202	0,034
a	1,573273	0,181756	27	3	0,285	0,037
b	0,524424	0,181756	27	1	0	0
c	1,048849	0,272633	27	3	0,276	0,031
d	1,048849	0,090878	27	3	0,276	0,031

5.3.4. Dataset B: Analisi di sensitività – ST-DBSCAN

a) $\boldsymbol{Eps1\_add\_delta = Eps1 + Eps1/2}$

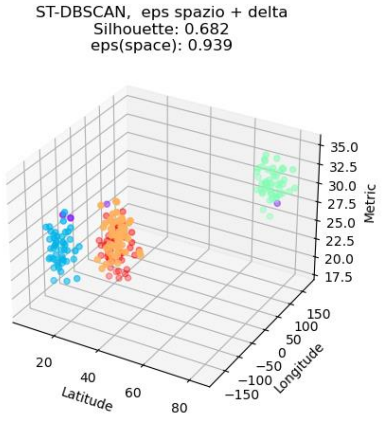


Figura 113 Lat, Long, Metric; ST-DBSCAN, Dataset B – Eps2+delta

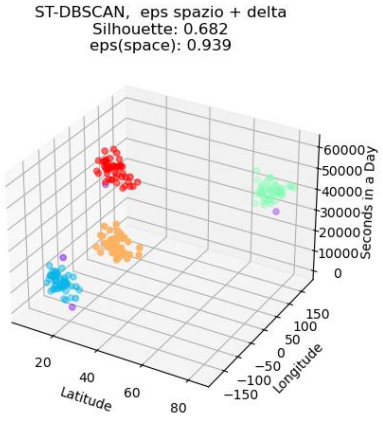


Figura 114 Lat, Long, Time; ST-DBSCAN, Dataset B – Eps2+delta

Aumentando $\boldsymbol{Eps1}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la *Silhouette* registra un leggero aumento (Figura 113 e Figura 113).

b) $\boldsymbol{Eps1\_meno\_delta = Eps1 - Eps1/2}$

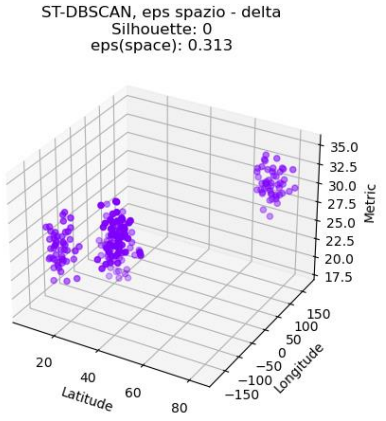


Figura 115 Lat, Long, Metric; ST-DBSCAN, Dataset B – Eps2-delta

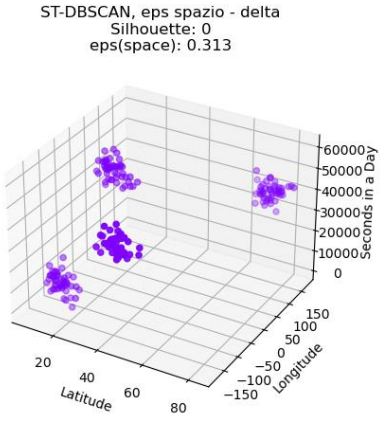
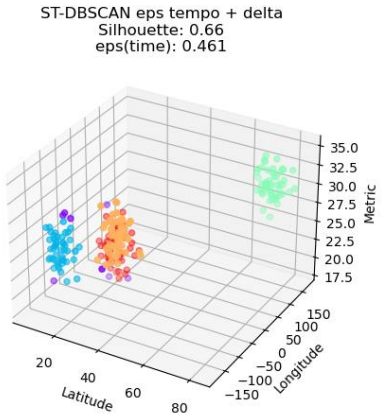


Figura 116 Lat, Long, Time; ST-DBSCAN, Dataset B – Eps2-delta

Riducendo $\boldsymbol{Eps1}$ di un valore pari al 50%, si ottiene una variazione significativa: il valore di $\boldsymbol{Eps1}$ si riduce così tanto che non viene identificato nessun *cluster* (Figura 115 e Figura 115).

c) $\boldsymbol{Eps2\_add\_delta = Eps2 + Eps2/2}$



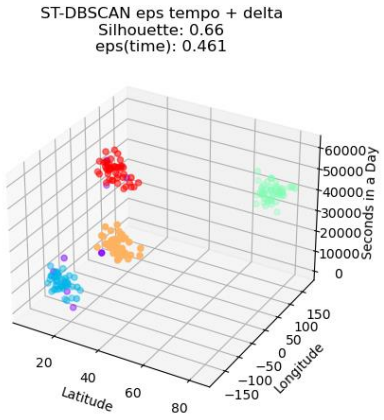
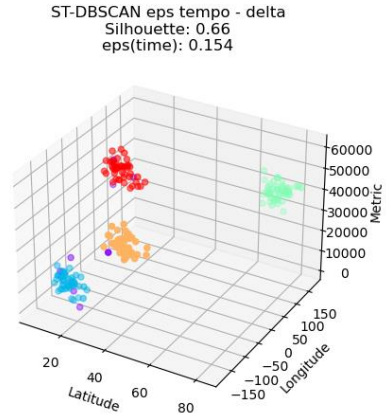


Figura 117 Lat, Long, Metric; ST-DBSCAN, Dataset B – Eps2+delta

Figura 118 Lat, Long, Time; ST-DBSCAN, Dataset B – Eps2+delta

Aumentando $\boldsymbol{Eps2}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la *Silhouette* registra un leggero aumento (Figura 117 e Figura 117).

d) $\boldsymbol{Eps2\_meno\_delta = Eps2 - Eps2/2}$



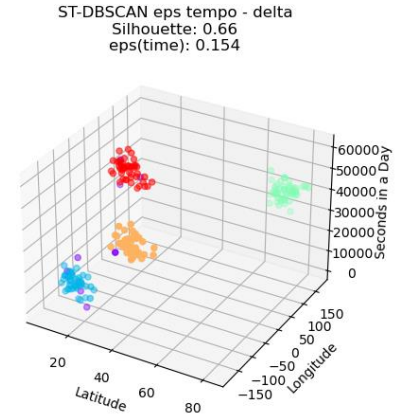


Figura 119 Lat, Long, Metric; ST-DBSCAN, Dataset B – Eps2-delta

Figura 120 Lat, Long, Time; ST-DBSCAN, Dataset B – Eps2-delta

Aumentando $\boldsymbol{Eps2}$ di un valore pari al 50%, non si ottiene una variazione significativa: il numero di cluster resta invariato mentre la *Silhouette* registra un leggero aumento (Figura 119 e Figura 119).

La Tabella 10 riassume i risuati ottenuti:

Tabella 10 Analisi di sensibilità – ST-DBSCAN (Dataset B)

	eps1	eps2	min point	cluster number	silhouette	DUNN INDEX
0	0,626	0,307	25	5	0,547	0,027
a	0,939	0,307	25	5	0,682	0,100
b	0,313	0,307	25	1	0	0
c	0,626	0,461	25	5	0,660	0,042
d	0,626	0,154	25	5	0,660	0,042

6. Clustering di dataset spazio-temporali reali

Nei paragrafi di seguito si riportano i risultati ottenuti dal *clustering* di alcuni dataset reali (non simulati) caratterizzati da attributi spaziali e temporali.

6.1. Georeferenced Time Series

Le *Georeferenced time series* (GTS) possono essere rappresentate con una matrice in cui le righe indicano la posizione fissa (ad esempio, le coordinate spaziali in termini di latitudine e longitudine) e le colonne indicano l'istante di tempo (esempio, data e ora) relativo al valore nella cella. Il metodo sviluppato per il clustering di *georeferenced timeseries* è basato sull'analisi di Chen et al. [22], i quali analizzano i dati sull'inquinamento dell'aria di alcune città in Cina, ottenuti dal sito web *PM25.in* forniti da *China National Environmental Monitoring Center* (CNEMC). In particolare, il *dataset* è costituito da 363 città cinesi (che rappresentano le righe del *dataset*), per ognuna delle quali è disponibile il valore medio giornaliero di PM2.5 rilevato per ogni stazione. Il *dataset* ricopre un periodo temporale che va dal 1° marzo 2015 al 28 febbraio 2018. Il *dataset* originale è stato diviso da Chen et al. [22] in base all'attributo temporale secondo le quattro stagioni, vale a dire la primavera (dal 1° marzo al 31 maggio), estate (dal 1° giugno al 31 agosto), autunno (dal 1° settembre al 31 novembre) e inverno (dal 1° dicembre al 28 febbraio). Per ogni gruppo, come già detto, ogni città rappresenta un record mentre le concentrazioni media di PM2.5 di ogni giorno sono gli attributi. Infatti:

- il gruppo Primavera ed Estate ha 276 attributi[2]
- il gruppo Estate ha 276 attributi [3]
- il gruppo Autunno ha 273 attributi [4]
- il gruppo Inverno ha 271 attributi. [5]

La funzione di similarità considerata da Chen et al. [22] per il *clustering* è la *cosine similarity*:

$$\cos(d_i, d_j) = \frac{d_i^t d_j}{|d_i d_j|}$$

Equazione 25 Cosine similarity

dove d_ie d_j sono i record *i* e *j*, corrispondenti a due generiche città. Di conseguenza l'indice per misurare la qualità del *cluster* ottenuto è definito nel seguente modo:

$$I_2 = \sum_{i=1}^{k} \sqrt{\sum_{d_i, d_j \in S_i} \cos(d_i, d_j)}$$

dove k è il numero di *cluster*, S_i è il *cluster* i-esimo e d_i e d_j sono due record appartenenti al *cluster* Si. Quindi in prima analisi Chen et al. [22] non hanno preso in considerazione le coordinate geografiche delle diverse stazioni per eseguire il *clustering*, ma solo le relative *timeseries*.

Tale approccio è stato riprodotto per analizzare un ***dataset* contenente i valori orari della temperatura atmosferica,** misurati nell'arco di 5 anni (Figura 121) e relativi a 50 città degli Stati Uniti e del Canada e 6 città di Israele; inoltre, per ogni città sono disponibili le coordinate geografiche, in termini di latitudine e longitudine.

[2] $276 = 3\ anni * [31\ giorni\ (marzo) + 30\ giorni\ (aprile) + 31 giorni\ (maggio)]$
[3] $276 = 3\ anni * [30\ giorni\ (giugno) + 31 giorni\ (luglio) + 31 giorni\ (agosto)]$
[4] $273 = 3\ anni * [30\ giorni\ (settembre) + 31\ giorni\ (ottobre) + 30 giorni\ (novembre)]$
[5] $271 = 3\ anni * [31\ giorni\ (dicembre) + 31\ giorni\ (gennaio) + 28\ giorni\ (febbraio)] + 1 giorno\ (29/02/2016)$

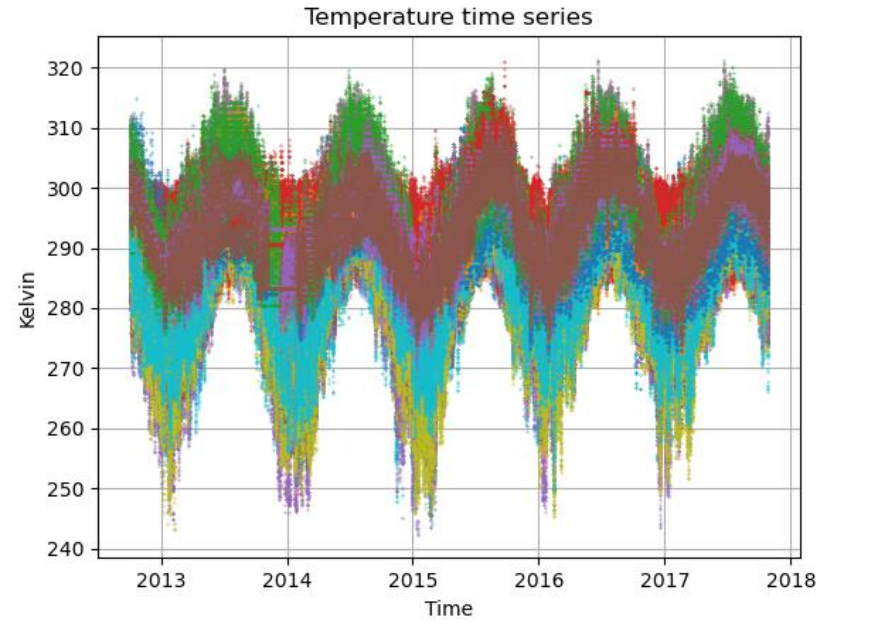


Figura 121 Temperature time series

Dopo aver isolato e normalizzato i dati relativi alla temperatura, escludendo quindi le coordinate geografiche di ogni stazione meteorologica, è stato eseguito l'algoritmo di *Agglomerative clustering*. È stata utilizzata come metrica di distanza la *cosine similarity* (Equazione 25), mentre il numero di cluster (8) è stato individuato in modo da massimizzare l'indice di *Silhouette* (0.495). Il risultato ottenuto è stato confrontato con quello ottenuto con l'*agglomerative clustering* con distanza euclidea (Equazione 2) e con il *K-Means*. Anche in questi casi, il numero di cluster (rispettivamente 3 e 11) è stato individuato in modo da massimizzare l'indice di *Silhouette* (rispettivamente 0.313 e 0.291).

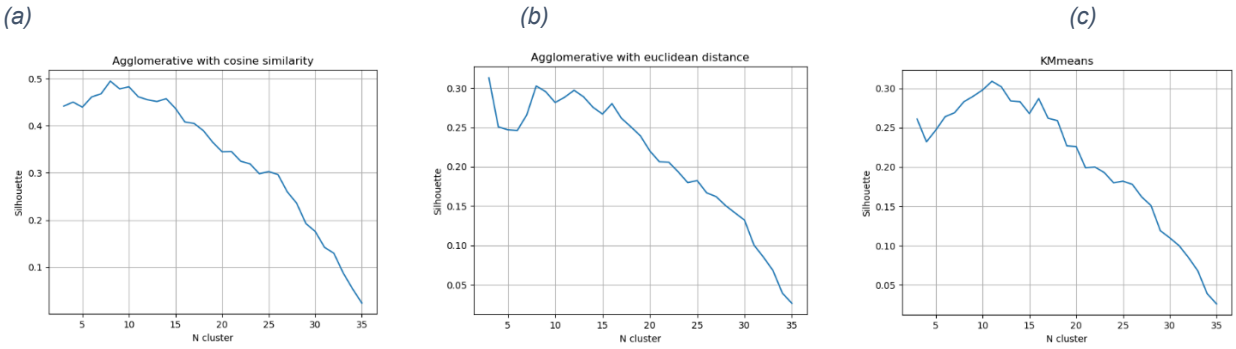


Figura 122 Silhouette in: (a) Agglomerative with cosine similarity (b) Agglomerative with euclidean distance (c) K-Means

La Figura 122 riporta l'andamento dei tre indici di *Silhouette;* si nota che la *cosine similarity* con *agglomerative clustering* è la combinazione più performante.

- ***Agglomerative – cosine similarity***

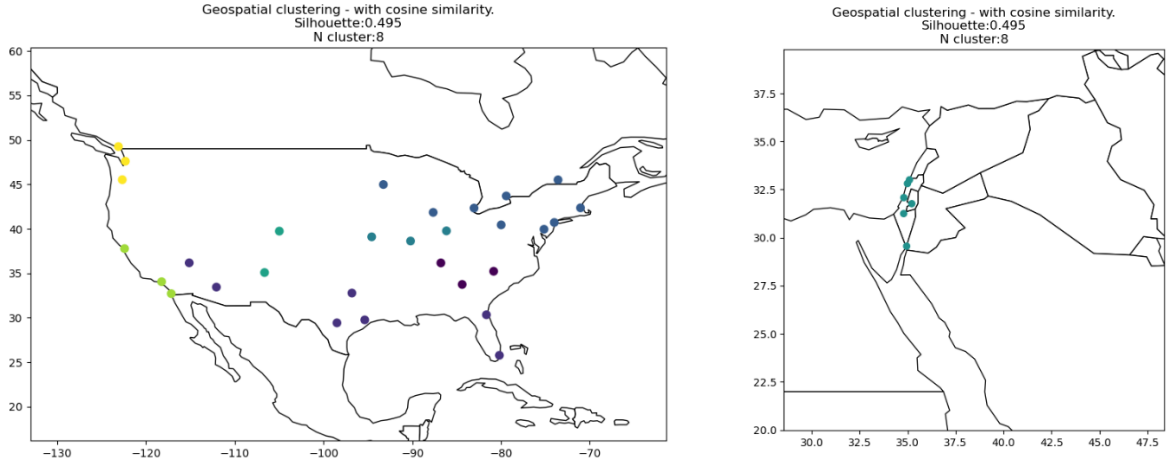


Figura 123: Risultati dell'Agglomerative clustering, con metrica di distanza cosine similarity

- ***Agglomerative – euclidean distance***

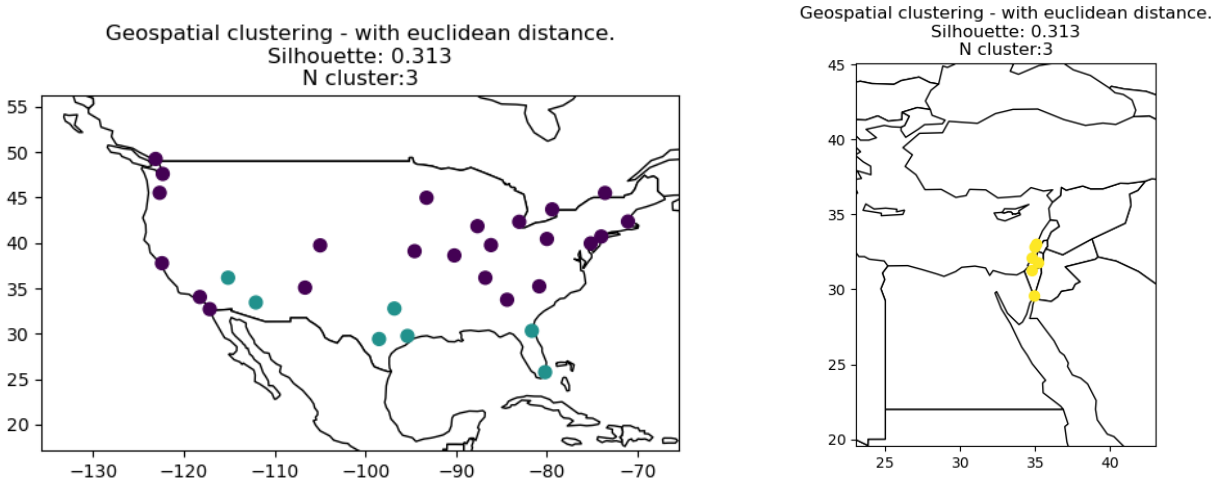


Figura 124: Risultati dell'Agglomerative clustering con metrica di distanza euclidea

- ***K-Means***

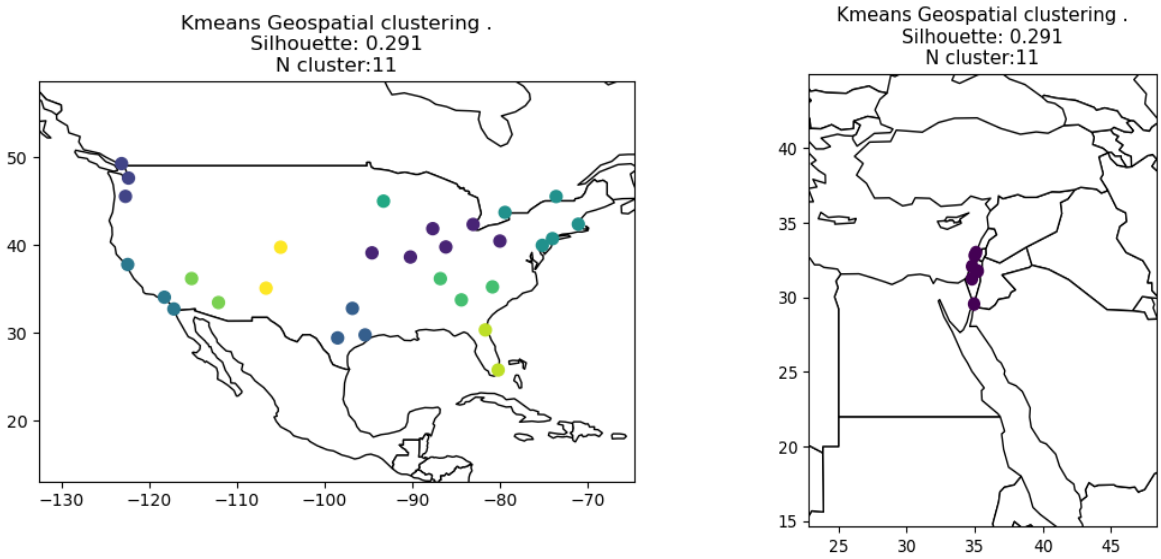


Figura 125: Risultati del K-Means con distanza euclidea

I risultati dei diversi *cluster* sono riportati sui planisferi (Figura 123, Figura 124, Figura 125): ogni *cluster* è rappresentato da un cerchio il cui colore riflette l'appartenenza ad un *cluster*. Si sottolinea che il *clustering* è stato eseguito considerando solo l'andamento della temperatura nel tempo, senza considerare la localizzazione geografica (date dalle coordinate latitudine e longitudine). Tuttavia, si nota come i *cluster* ottenuti rispecchino in un certo qual modo la distribuzione geografica (quindi città vicine vengono raggruppate in uno stesso *cluster*) e le caratteristiche climatiche (ad esempio, la temperatura oceanica è diversa da quella continentale, per cui vengono individuati cluster diversi). Un risultato analogo è stato raggiunto da Chen et al. [22], citato in precedenza, analizzando le *timeseries* dei livelli di $PM_{2.5}$ rilevati in diverse città della Cina; infatti, senza considerare l'informazione geografica, le città geograficamente adiacenti sono state aggregate in stessi *cluster*, possedendo caratteristiche simili (in quel caso la similarità è legata ai valori temporali delle concentrazioni di $PM_{2.5}$).

Questa peculiarità potrebbe non essere una caratteristica generale delle *timeseries,* ma presumibilmente è relativa a dati che riguardano condizioni ambientali, come *timeseries* di pressioni, temperatura, umidità, livelli di precipitazioni, caratteristiche dei venti o misure legate all'inquinamento.

In altri contesti, invece, non considerando anche la posizione, si potrebbe rischiare di perdere delle informazioni, ottenendo modelli di *cluster* meno precisi.

6.2. TRAIETTORIE GPS

Il metodo "*ST-EVENTS*" descritto precedentemente nel capitolo 5 e utilizzato per *clusterizzare dataset* simulati, può essere facilmente esteso e utilizzato per il *cluster* di traiettorie. A differenza del dato definito *ST-EVENTS,* le *trajectories* sono la sequenza delle posizioni spaziali in cui è stato l'oggetto, insieme ai relativi *timestamp*; quindi la traiettoria di ogni oggetto è un insieme di *ST-EVENTS,* con posizione non fissa ma mobile, registrata con continuità nel tempo.

È stato eseguito un *agglomerative clustering* utilizzando la metrica di distanza dell'Equazione 9 e i relativi parametri, descritti nella Tabella 4, su un sottoinsieme del *GPS trajectory dataset*, ovvero un dataset raccolto da *Microsoft Research Asia* durante il progetto *Geolife* [2] *,* in cui sono stati registrati tramite *smartphone* i dati relativi agli spostamenti di 182 utenti volontari in un periodo di 5 anni. Si tratta di un *dataset* in cui ogni *record*

è un punto della traiettoria, le cui *features* sono: tempo (data e ora), latitudine, longitudine e altitudine. Nel caso specifico, l'altitudine è la quarta dimensione, considerata come 'metrica'.

Il *dataset* complessivo contiene 17621 traiettorie raccolte con il GPS dello *smartphone* di 182 utenti. Le traiettorie sono relative prevalentemente a più 30 città in Cina, ma anche alcune città in Europa e negli USA. Il *dataset* è organizzato in cartelle, in ognuna della quale sono presenti i GPS *log files* di ciascun utente, suddivisi in base ai giorni.

In prima analisi è stato selezionato un sottoinsieme del *dataset* dalle dimensioni molto ridotte, considerando le traiettorie di uno stesso utente, raccolte in tre giorni diversi. È stato eseguito il *clustering* sia su tutti e tre i giorni, sia separatamente per ogni giorno.

- **TRAIETTORIE 24/10/2008, 26/10/2008, 04/11/2008 (1179 *records*)**

È stato eseguito l'*agglomerative clustering* con la metrica di distanza dell'Equazione 9, fissando il peso della coordinata temporale pari a quello delle coordinate spaziali ($\boldsymbol{w_1 = w_2 = 0.50}$), ottenendo un indice di *Silhouette* pari a 0.863. I risultati ottenuti sono riportati in Figura 126 e in Figura 127; si nota come sono stati individuati quattro cluster, di cui due (quello in viola e quello in blu) sono relativi ad uno stesso intervallo di tempo; al contrario il cluster in giallo è relativo alla prima parte della giornata, mentre quello in verde è relativo alla parte finale. Si potrebbe inferire che l'utente sia andato dalla posizione del cluster giallo a quello verde, passando per il blu e il viola.

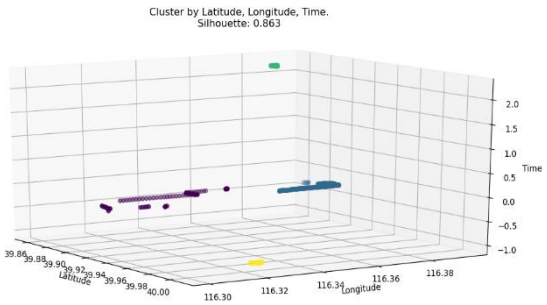


Figura 126: Risultati del clustering delle traiettorie di 24/10/2008, 26/10/2008, 04/11/2008

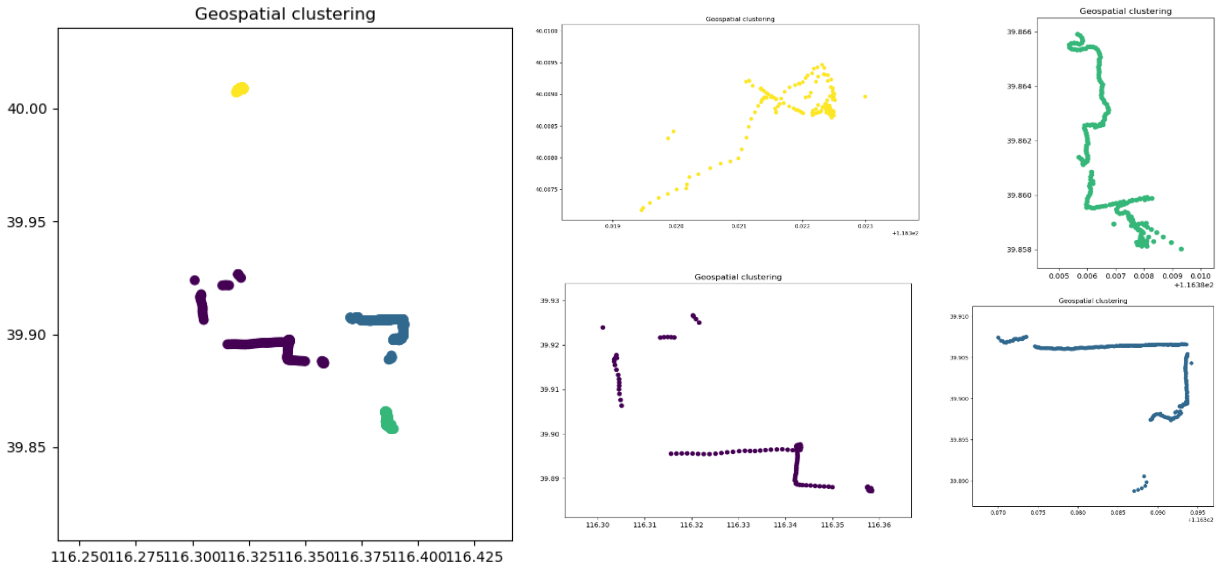


Figura 127: Risultati del clustering delle traiettorie di 24/10/2008, 26/10/2008, 04/11/2008

- **TRAIETTORIE 24/10/2008 (244 *records*)**

È stato eseguito l'*agglomerative clustering* considerando le traiettorie del giorno 24/10/2008 (corrispondenti al *cluster* giallo individuato precedentemente). In questo caso, tuttavia, è stata aggiunta anche la dimensione '***altitudine***' del punto, accanto alla latitudine, longitudine e il *timestamp*. L'altitudine viene in generale definita '***metrica***'.

Inizialmente non sono stati fissati i parametri legati al numero di *cluster* e peso delle *features,* lasciando che questi venissero scelte in automatico in modo da massimizzare la *Silhouette.* È stato attribuito un peso

massimo alla metrica – altitudine $\boldsymbol{w_3 = 1}$, mentre non sono state considerate le altre dimensioni $\boldsymbol{w_1 = w_2 = 0}$. I risultati sono riportati in Figura 128. Si nota come il cluster giallo e il cluster azzurro siano localizzati ad altitudini nettamente differenti rispetto al cluster viola.

Successivamente è stato fissato il peso della dimensione tempo pari a quello delle coordinate $\boldsymbol{w_1 = w_2 = 0.5}$, non considerando la metrica (altitudine) $\boldsymbol{w_3 = 0}$. I risultati sono riportati in Figura 129. Si nota come l'unico cluster ottenuto precedentemente analizzando le traiettorie dei tre giorni (cluster giallo in Figura 127), è stato così separato in quattro cluster, dei quali, analizzando il grafico in funzione del tempo, si potrebbe inferire che l'utente sia partito dal cluster viola (Figura 129, in basso) e sia arrivato al cluster in blu, passando per il verde e il giallo.

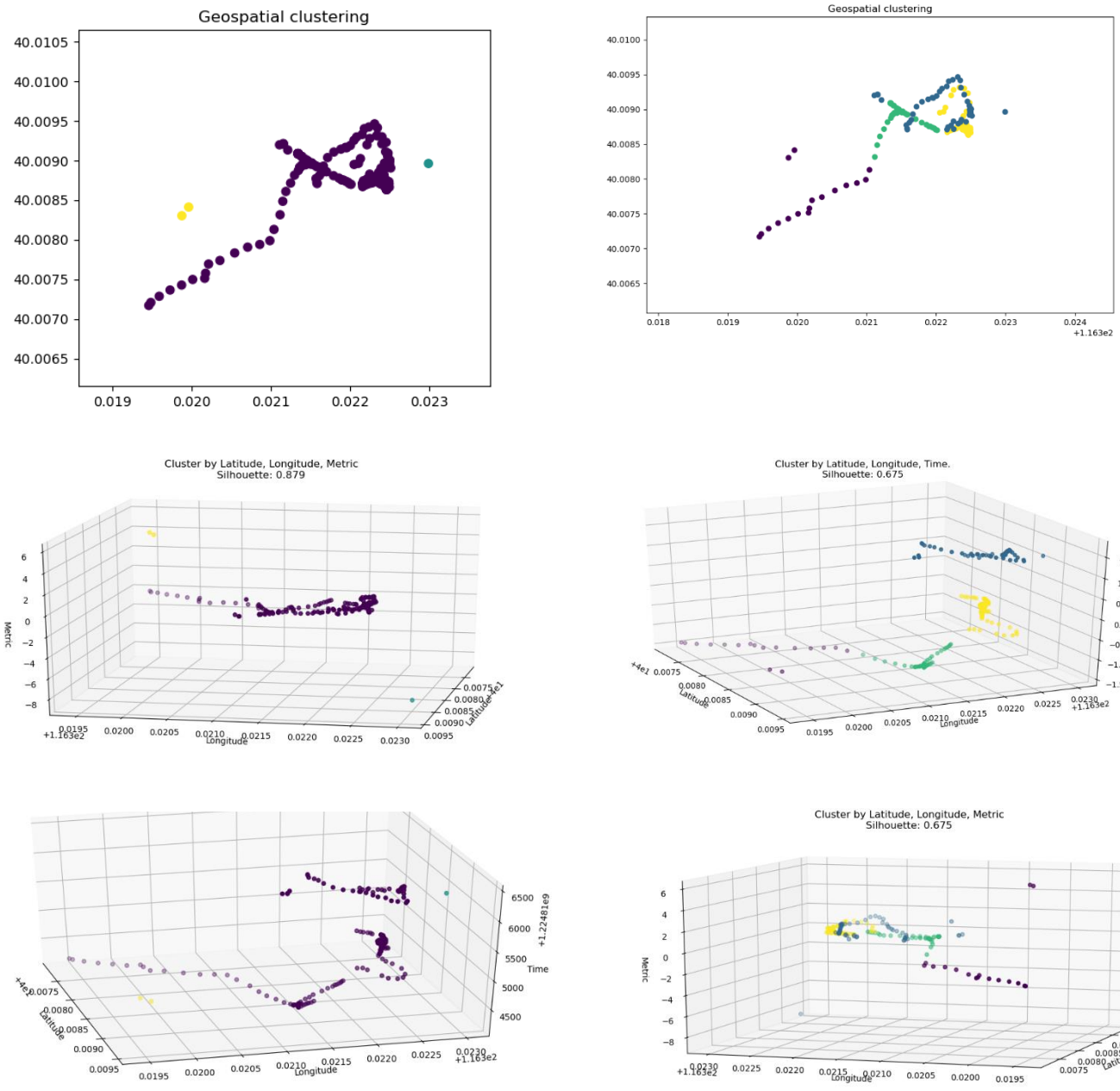


Figura 129: Risultati del clustering delle traiettorie del 24/10/2008 con metric w_3=1; space, coordinate w_1=w_2=0

Figura 128: Risultati del clustering delle traiettorie del 24/10/2008 con metric w_3=0, space, coordinate w_1=w_2=0.5

SF PARK – BIKE SHARING

Il modello di *clustering* portato avanti in questo nei paragrafi successivi (6.3 e 6.4) può essere considerato un'estensione di quello adottato per i dataset simulati di tipo *ST-EVENTS,* analizzati nel capitolo 5. Infatti, oltre a considerare la dimensione temporale (con una *feature* di tipo *date, time* e/o *datetime*), una o più dimensioni generiche, definite 'metriche', si considerano due *features* che indicano la posizione, ognuna caratterizzata da una coppia di coordinate geografiche, ad esempio espresse in termini di latitudine e longitudine. Le due posizioni potrebbero modellare, ad esempio, il punto di partenza e quello di arrivo, in un *dataset* in cui ogni *record* può corrispondere ad un percorso eseguito da una bicicletta messa a disposizione da un servizio di *bikesharing,* oppure può corrispondere ad un parcheggio, la cui superficie si estende tra i due punti. Entrambi i dataset verranno analizzati successivamente.

Prima di far ciò, è necessario sottolineare che rispetto al modello base, verrà modificata la metrica di distanza, la quale sarà così definita:

$$\boldsymbol{D_H}\,(s_i, s_j, \quad e_h, e_k) = \boldsymbol{D_{HS}} + \boldsymbol{D_{HE}}$$

$$\boldsymbol{D_{HS}}\,(s_i, s_j) = 2\arcsin\left[\sqrt{sin^2\frac{s_{1i} - s_{1j}}{2} + cos(s_{1i})\,cos(s_{1j})\,sin^2\frac{s_{2i} - s_{2j}}{2}}\right]$$

$$\boldsymbol{D_{HE}}(e_h, e_k) = 2\arcsin\left[\sqrt{sin^2\frac{e_{1h} - e_{1k}}{2} + cos(e_{1h})\,cos(e_{1k})\,sin^2\frac{e_{2h} - e_{2k}}{2}}\right]$$

Dove:

$s_i\ (s_{1i}, s_{2i})$ è il punto di partenza e $e_h\ (e_{1h}, e_{2h})$ è il punto di arrivo del segmento $\boldsymbol{\overline{s_i e_h}}$

$s_j\ (s_{1j}, s_{2j})$ è il punto di partenza e $e_k\ (e_{1k}, e_{2k})$ è il punto di arrivo del segmento $\boldsymbol{\overline{s_j e_k}}$

6.3. SF-PARK

6.3.1. Descrizione dataset

Nel 2011, l'Agenzia municipale dei trasporti di San Francisco (SFMTA - San Francisco Municipal Transportation Agency) ha avviato un progetto di *smart parking*, denominato *SF-PARK*, il cui obiettivo era il miglioramento della gestione dei parcheggi lungo le strade di San Francisco, principalmente attraverso aggiustamenti dei prezzi reattivi alla domanda. Per fare questo è stata necessaria la raccolta e l'analisi dei dati relativi alla disponibilità dei parcheggi su strada. A tal fine, circa 8.000 posti auto sono stati dotati di specifici sensori nell'asfalto, trasmettendo periodicamente informazioni sulla disponibilità del parcheggio.

Il *dataset* analizzato nei paragrafi successivi corrisponde ad una parte di un primo *dataset*, acquisito mediante API pubblica del progetto *SF-PARK,* che riporta la disponibilità di alcuni parcheggi di San Francisco misurata grazie a tali sensori posti sotto l'asfalto. Il dataset "*SF-PARK*" in analisi è disponibile direttamente all' *URL:* https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YLWCSU. Il periodo considerato va dal **13/06/2013** ore **00:04** al **24/07/2013** ore **23:58** con una frequenza di campionamento di **5 minuti**.

Il *dataset* considerato è caratterizzato dai seguenti attributi rilevanti per l'analisi:

- *segmentid*: ID del parcheggio (anche chiamato *block face* dall'progetto *SF-PARK*)
- *streetname*: nome della strada con il *range* dei numeri civici compresi
- *startx, starty, endx, endy:* coordinate latitudine e longitudine di inizio e fine del parcheggio
- *timestamp:* corrispondente all'istante di tempo in cui il dato è stato acquisito, arrotondato al minuto
- *capacity:* il numero totale di posti disponibili all'interno del parcheggio (il numero di posti può variare nel tempo, a casa delle restrizioni del numero di parcheggi)
- *occupied:* numero complessivo di posti occupati all'interno del parcheggio.

È stato introdotto un ulteriore attributo che mettesse in relazione la capacità del parcheggio con il numero effettivo di posti occupati, è stato definito il seguente attributo:

$$metric = capacity\ utilization\ rate = \frac{occupied}{capacity}$$

Valori nulli, ad esempio $capacity = 0\ e\ occupied = 0$ sono stati eliminati.

Il dataset detto "*SF-PARK*" è caratterizzato da **420** parcheggi, detti anche *block face,* identificati da un punto di partenza e da un punto di arrivo; questi corrispondono ad una coppia di coordinate latitudine-longitudine.

Ogni parcheggio, quindi, può essere rappresentato come un segmento, identificato a sua volta dal nome della strada e dal range di numeri civici compresi. Ad esempio, il *segment* con **ID *326042*** corrisponde al parcheggio in ***Beach St*** che comprende i numeri civici dal **400** al **496**. Le coordinate di partenza sono *latitudine 37.807308* e *longitudine -122.415832,* mentre quelle di arrivo sono *latitudine 37.807142*. e *longitudine -122.417134.* La Figura 130 e la Figura 131 di seguito mostrano il dettaglio di *google maps* del segments ID in questione.

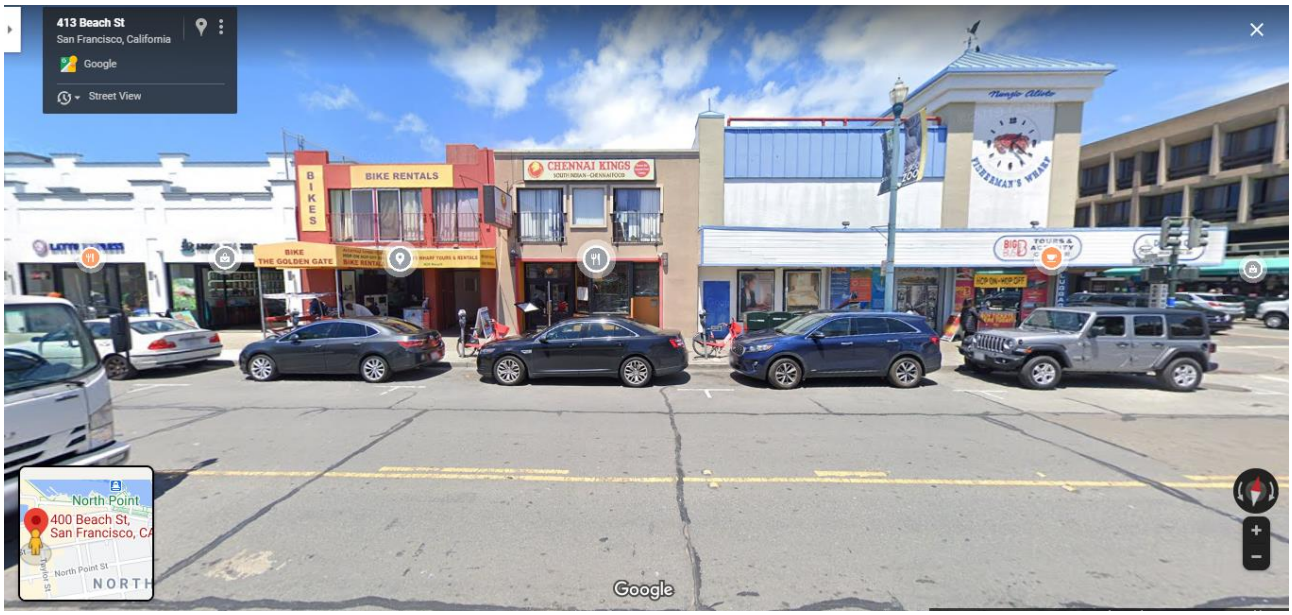


Figura 130: Dettaglio di google maps del segments ID 326042 (numero civico 413)

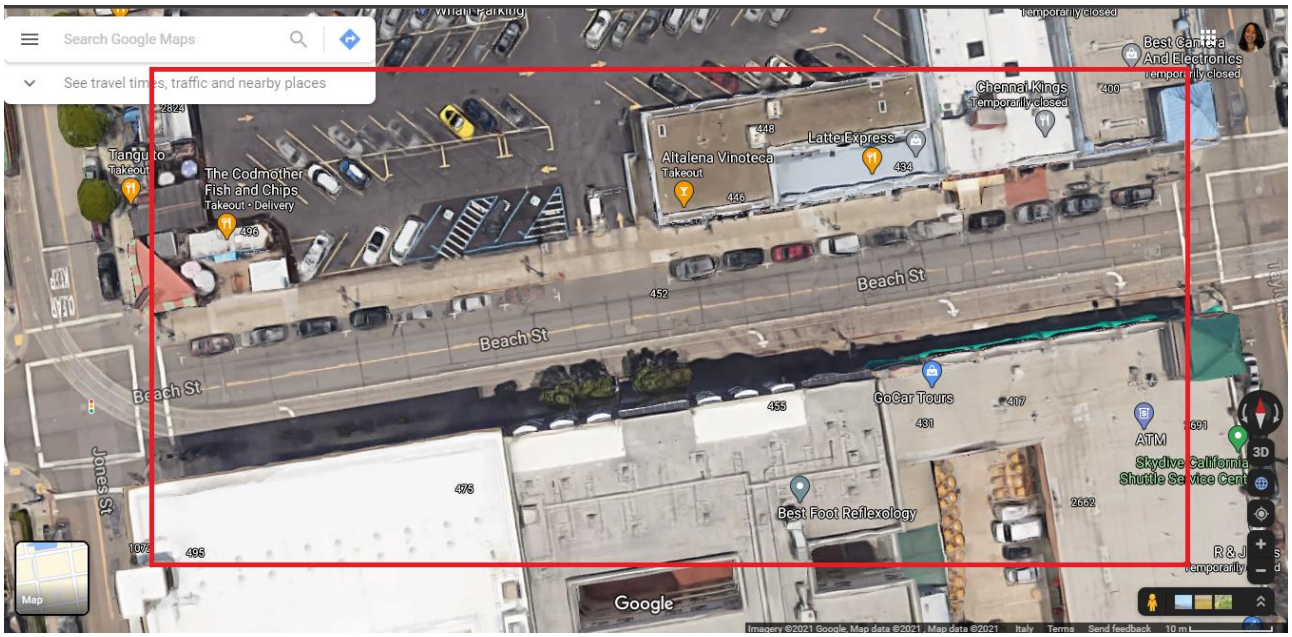


Figura 131: Dettaglio di google maps del segments ID 326042 (con numeri civici dal 400 al 496)

Il periodo considerato va dal **13/06/2013** ore **00:04** al **24/07/2013** ore **23:58** con una frequenza di campionamento di **5 minuti** (per un totale di **42** giorni in analisi).

Per ogni *segment* ID (in totale 420), ogni giorno sono rilevati 288 misure relative al numero di parcheggi occupati $\left(\mathbf{288}\ \boldsymbol{rilevazioni/giorno} = \frac{\mathbf{60}\ {}^{minuti}/_{ora} * \mathbf{24}\ {}^{ore}/_{giorno}}{\mathbf{5}\ {}^{minuti}/_{rilevazione}}\right)$.

In totale il dataset è costituito da 5.080.320 records $(288 * 42 * 420)$.

Gli attributi presi in considerazione sono:

1. *timestamp,* che corrisponde alla dimensione temporale
2. *startx* che corrisponde alla coordinata geografica (latitudine) di partenza/inizio del parcheggio
3. *starty,* che corrisponde alla coordinata geografica (longitudine) di partenza/inizio del parcheggio
4. *endx* che corrisponde alla coordinata geografica (latitudine) di arrivo/fine del parcheggio
5. *endy,* che corrisponde alla coordinata geografica (longitudine) di arrivo/fine del parcheggio
6. *capacity utilization rate,* che corrisponde alla metrica.

A causa delle elevate dimensioni del dataset in analisi (5.080.320 *records* x 6 *features*), non è stato possibile performare gli algoritmi di *clustering Agglomerative* e *K-Medoids* in tempi ragionevoli, utilizzando l'intero dataset così costituito. Per ridurre la dimensionalità, il *dataset* è stato partizionato rispetto all'attributo *capacity utilization rate; dopodichè,* per ogni intervallo è stato eseguito un campionamento (stratificato rispetto ai parcheggi) su cui sono stati performati i suddetti algoritmi di *clustering*.

La Tabella 11 mostra gli intervalli identificati per la partizione.

Di seguito si riporta la tabella riassuntiva:

Tabella 11: partizione del dataset SF-Park

ID	capacity utilization rate	#records	% TOT
1	1>=x>=0.95	394.258	7,8%
2	0,95>x>=0.80	473.357	9,3%
3	***0,80>x>=0.60***	***1.083.540***	***21,3%***
4	0,60>x>=0.50	676.969	13,3%
5	0,50>x>=0,40	355.939	7,0%
6	0,40>x>=0,30	445.430	8,8%
7	0,30>x>=0,25	237.587	4,7%
8	0,25>x>=0	919.005	18,1%
	record contenenti valori nulli (eliminati)	*494.235*	9,7%

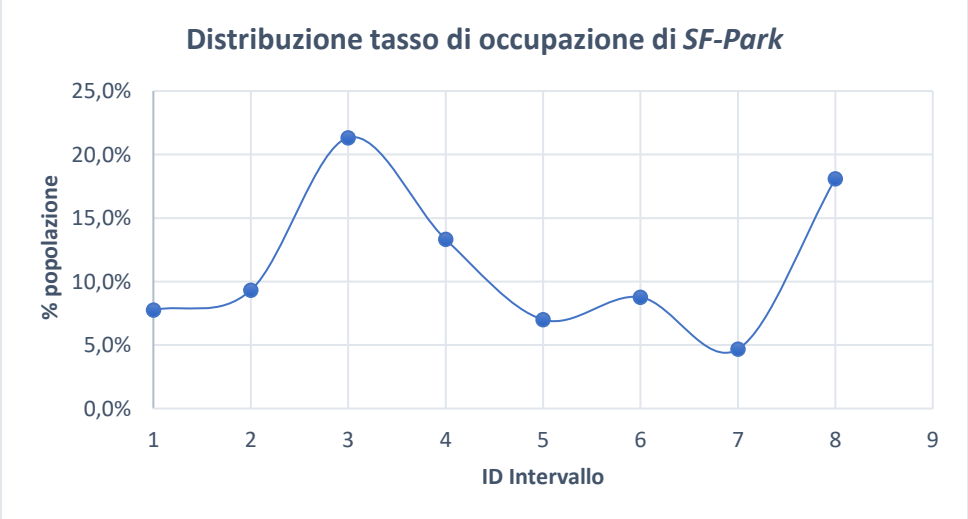


Figura 132: Distribuzione del tasso di occupazione di SF-Park

Si nota (Figura 132) che il 21.3% dei parcheggi totali rientra nella fascia target con *capacity utillization rate* compreso tra il 60% e l'80%, obiettivo del progetto *SF-Park* (per maggiori informazioni circa tale aspetto fare riferimento al paragrafo 6.3.5). Tuttavia, il 18.1% dei parcheggi presenta un basso tasso di occupazione, compreso tra lo 0% e il 25%. Al contrario, circa il 16% dei parcheggi, presenta un *capacity utillization rate* superiore alla soglia target e maggiore del 80%.

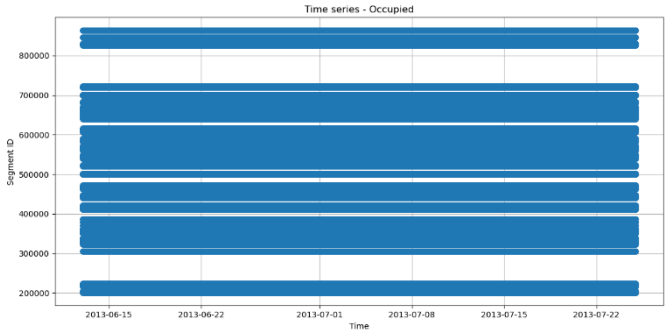


Figura 133: Distribuzione nel tempo dei dati rilevati di ogni Segment ID

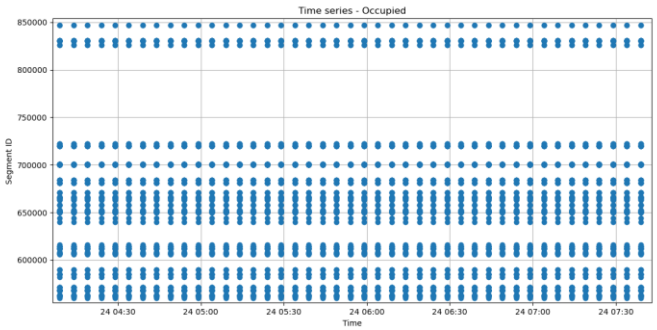


Figura 134 Zoom di Figura 133

Come affermato precedentemente, per ogni intervallo è stato eseguito un campionamento (stratificato rispetto ai parcheggi) su cui sono stati performati i suddetti algoritmi di *clustering*. Si sono ottenuti 8 dataset di tipo ST-Events, caratterizzato da una componente temporale (data e ora), quattro coordinate spaziali (latitudine e longitudine che identificano il luogo di inizio e fine del parcheggio) e il tasso di utilizzo del parcheggio misurato nel relativo istante temporale. Dunque, il dataset campione non è più di tipo *georeferenced timeseries,* come quello di partenza (in Figura 133 e Figura 134); si nota che il dataset iniziale contiene i valori delle metriche continui nel tempo per ogni parcheggio-*segment id*).

Dopo effettuato il campione per ogni partizione, normalizzato i dati (in questo caso è stata utilizzata una normalizzazione *MinMax*) e scelto i vari parametri definiti in Tabella 4, sono stati performati gli algoritmi di l'*agglomerative clustering* e il *K-Medoids.*

6.3.2. Partizione con capacity utilization rate tra 100% e 95%

Si riportano i risultati della partizione 1.

A partire dal dataset di partenza, sono stati isolati i punti con *capacity utilization rate* compreso tra 1 e 0.95**,** che sono risultati essere pari a 394.258 (pari al 7.8 % del totale). Questi potrebbero essere ricondotti a

malfunzionamenti dei sensori oppure al contrario, potrebbe non trattarsi di errori di rilevazione ma di parcheggi molto affollati, quasi completamente occupati.

Sulla base di tale partizione, è stato estratto un campione casuale, stratificato rispetto al *segment ID,* costituito da 100 items. Si riporta nella Figura 135 il grafico dell'andamento del *capacity utilization rate* (definita in generale *"Metric"*) nel periodo considerato. Si nota che:

- il campione estratto è caratterizzato da soli due parcheggi con tasso di occupazione diverso da 1
- i giorni con tasso di occupazione pari a 1 sono in linea generale uniformemente distribuiti nel periodo considerato.

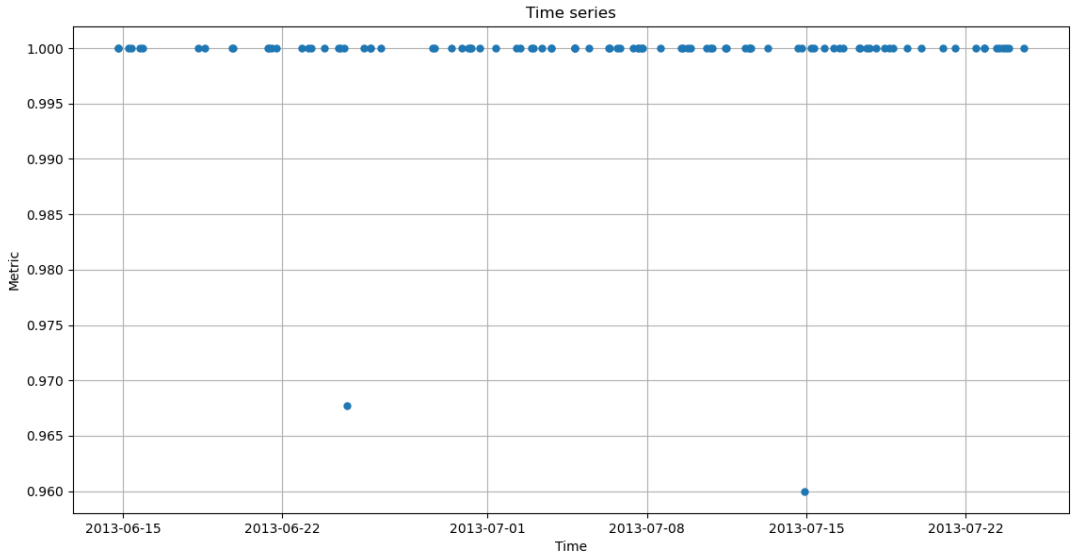


Figura 135

Sono stati applicati gli algoritmi *agglomerative clustering* e *K-Medoids* sul campione individuato, utilizzando la metrica di distanza presentata precedentemente e richiamata di seguito:

$$\boldsymbol{D(i,j) = \frac{1}{\sum_{i=1}^{n} w_i}\left(w_1 * D_S(s_i, e_i; s_j, e_j) + w_2 * D_T(t_i, t_j) + \sum_{a=3}^{q} D_a(a_i, a_j)\right)}$$

1) considerando una metrica di distanza spaziale ***haversine*** $\boldsymbol{D_S}$ **,** pari a:

$$\boldsymbol{D_S}\,(s_i, s_j, \quad e_i, e_j) = \boldsymbol{D_{S_S} + D_{S_E}}$$

$$\boldsymbol{D_{S_S}}\,(s_i, s_j) = 2\arcsin\left[\sqrt{sin^2\frac{lat_{s_i} - lat_{s_j}}{2} + cos(lat_{s_i})\,cos\left(lat_{s_j}\right)sin^2\frac{long_{s_i} - long_{s_j}}{2}}\right]$$

$$\boldsymbol{D_{S_E}}(e_i, e_j) = 2\arcsin\left[\sqrt{sin^2\frac{lat_{e_i} - lat_{e_j}}{2} + cos(lat_{e_i})\,cos\left(lat_{e_j}\right)sin^2\frac{long_{e_i} - long_{e_j}}{2}}\right]$$

2) considerando una metrica di distanza temporale **"*absolute*"** $\boldsymbol{D_T}$ **,** pari a:

$$\boldsymbol{D_T(t_i, t_j) = abs(t_i - t_j)}$$

Nel caso specifico ogni record del dataset rappresenta un parcheggio, che può essere rappresentato come un vettore $\boldsymbol{i}$, caratterizzato da:

un punto di partenza s_i caratterizzato da una coppia di coordinate di latitudine e longitudine (lat_{s_i}, lon_{s_i})

- un punto di arrivo e_i caratterizzato da una coppia di coordinate di latitudine e longitudine (lat_{e_i}, lon_{e_i})
- una dimensione temporale (ora-data) t_i che indica l'istante della misura
- una metrica a_i che indica il tasso di occupazione del parcheggio, ovvero il numero di posti occupati in quel momento rispetto al numero totali di posti disponibili

Per ogni coppia di parcheggi, rappresentati da un vettore i da un vettore j, viene calcolata la distanza $D(i,j)$. Tale distanza è alla base degli algoritmi *Agglomerative* e *K-Medoids* applicati.

6.3.2.1. Risultati cluster con $w_1 = w_2 = 0,50$

Al fine di attribuire lo stesso peso alla componente spaziale e a quella temporale è stato impostato $w_1 = w_2 = 0,50$. Il numero di cluster è stato individuato automaticamente, in modo da massimizzare l'indice di *Silhouette.*

Agglomerative clustering – partizione SF Park con tasso di occupazione tra 100% e 95%

L'algoritmo *Agglomerative clustering* ha individuato 4 cluster, con un indice di *Silhouette* pari a 0.441. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riportano diverse tipologie di grafico. I diversi colori indicano i *cluster* di appartenenza, il simbolo ▲ indica la posizione iniziale del parcheggio, mentre il simbolo ▼ indica la posizione finale del parcheggio.

In particolare, di seguito si riporta:

a) la Figura 136 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo: si nota che la metrica è una dimensione discriminante soprattutto per il cluster in blu, caratterizzato da metrica diversa da 1, a differenza di tutti gli altri cluster;

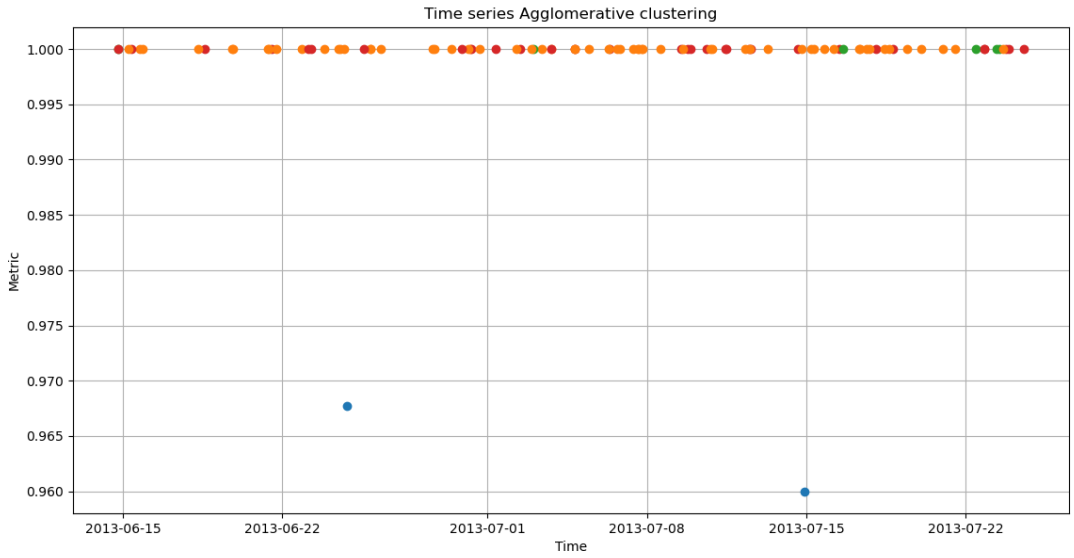


Figura 136

b) la Figura 137 mostra il grafico in 3 dimensioni della latitudine, longitudine e metrica:
si nota che la dimensione spaziale è una discriminante soprattutto per il 3 cluster (blu, verde e giallo), caratterizzati da stessa metrica ma posizionati in aree geografiche diverse.

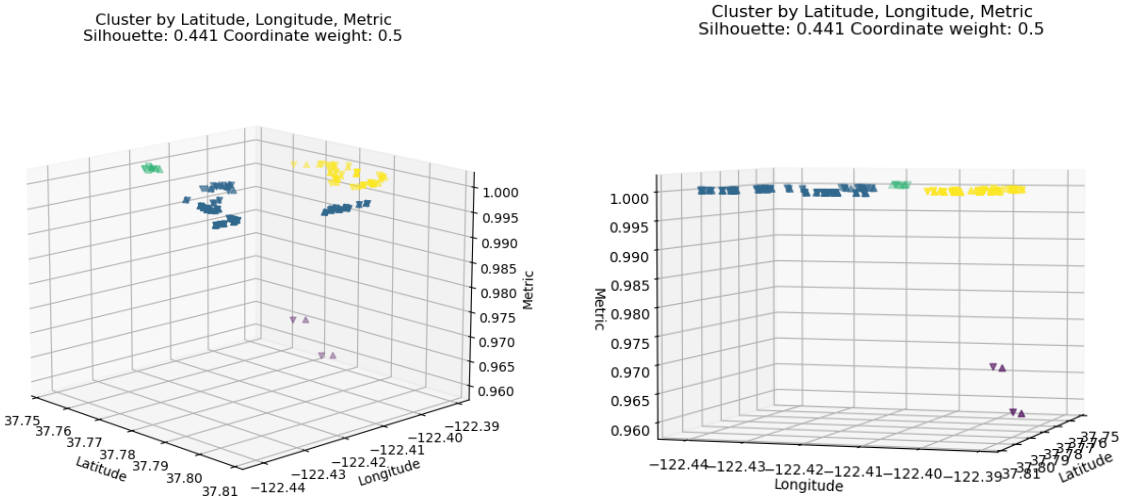


Figura 137

c) la Figura 138 mostra il grafico in 3 dimensioni della latitudine, longitudine e tempo:
si nota che la dimensione temporale non sembra essere una discriminante per l'identificazione dei 4 cluster.

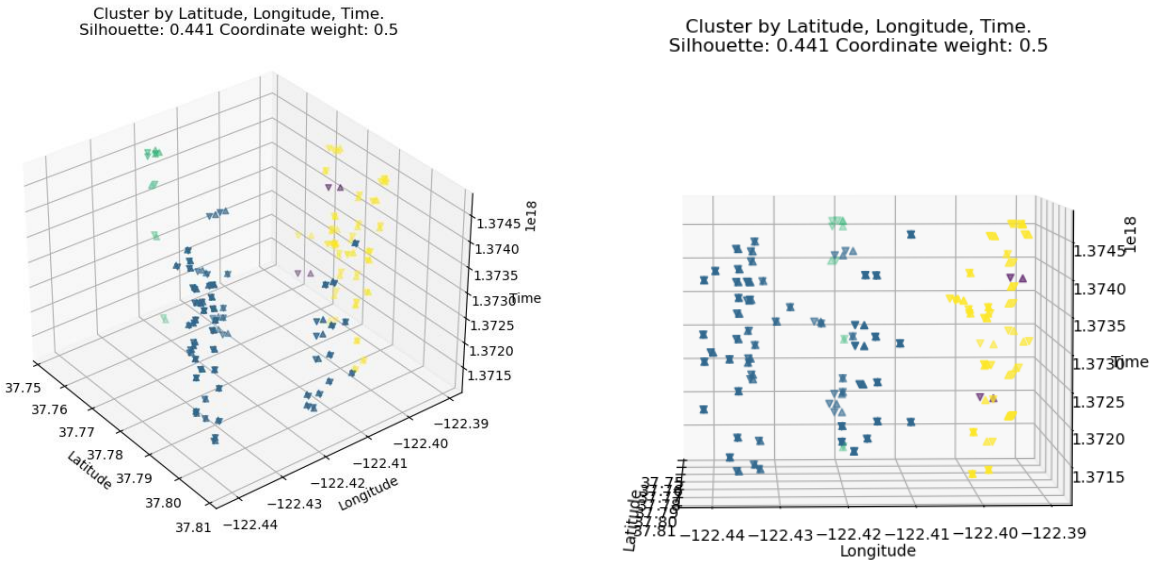


Figura 138

d) la Figura 139 che mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*

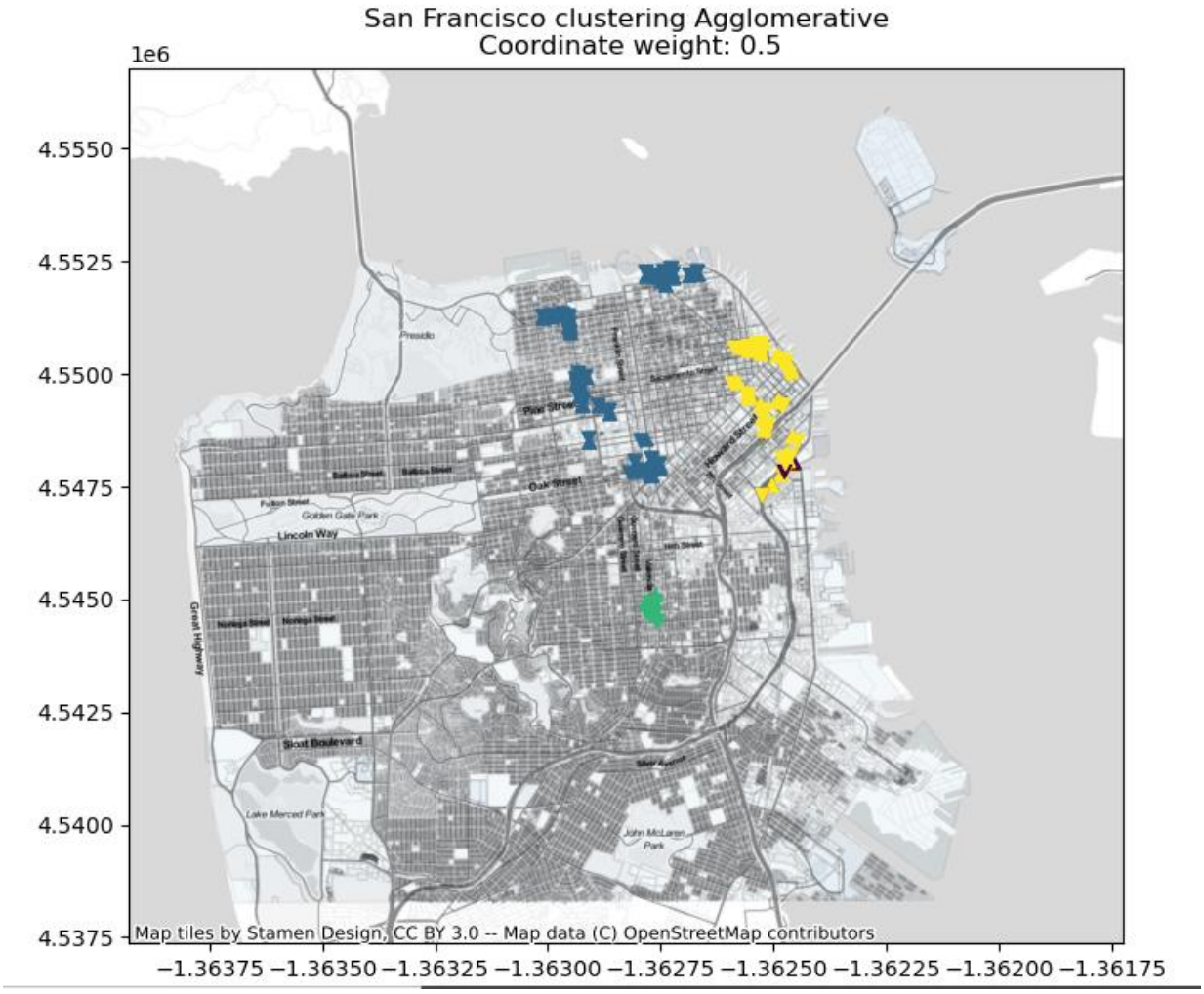


Figura 139

K-Medoids– partizione SF Park con tasso di occupazione tra 100% e 95%

Sullo stesso campione è stato applicato l'algoritmo *K-Medodis,* al fine di confrontare i risultati ottenuti applicando precedentemente l'*agglomerative clustering*. In linea generale, si nota che il tempo è una dimensione maggiormente discriminante per l'individuazione dei cluster con il *K-Medoids.*

A parità di peso della componente spaziale e temporale ($\boldsymbol{w_1 = w_2 = 0,50}$) e metrica di distanza, applicando il *K-Medoids* sono stati identificati 9 *cluster,* con un indice di *Silhouette* pari a **0.315**. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 140 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo: si nota che la metrica non sembra essere la dimensione discriminante rilevante

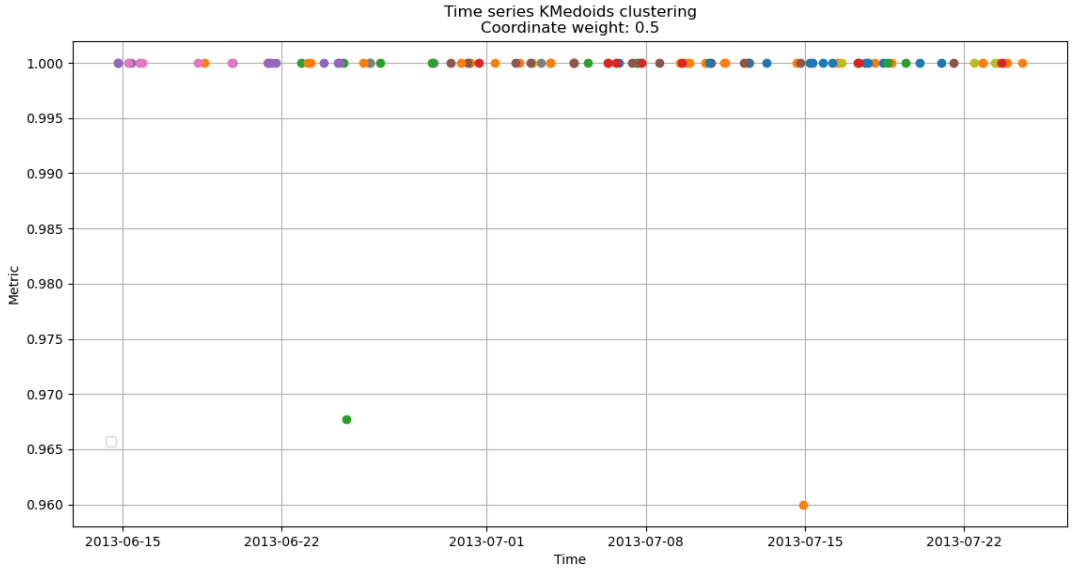


Figura 140

b) la Figura 141 che mostra il grafico in 3 dimensioni della latitudine, longitudine e metrica: si nota che la dimensione spaziale sembra essere una discriminante più importante rispetto alla metrica;

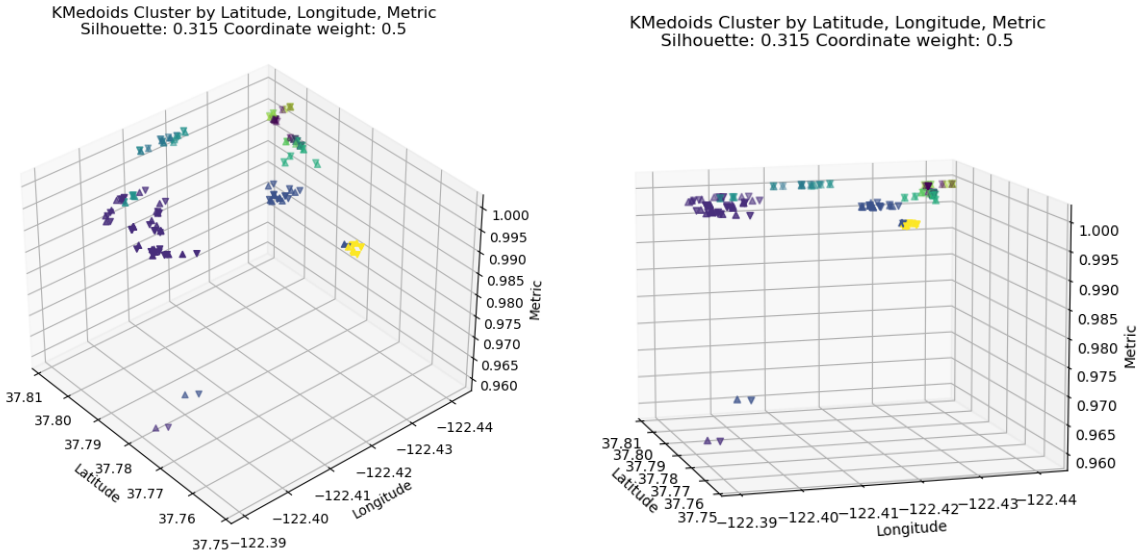


Figura 141

c) la Figura 142 mostra il grafico in 3 dimensioni della latitudine, longitudine e tempo:
si nota che la dimensione temporale sembra essere una discreta discriminante per l'identificazione dei 9 cluster.

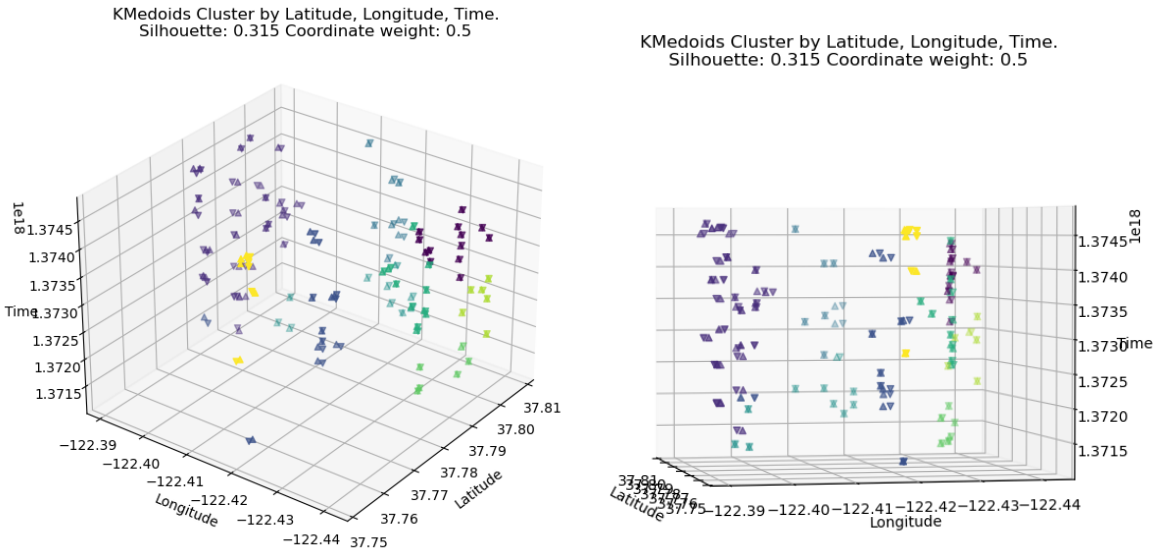


Figura 142

d) la Figura 143 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi clusterizzati:

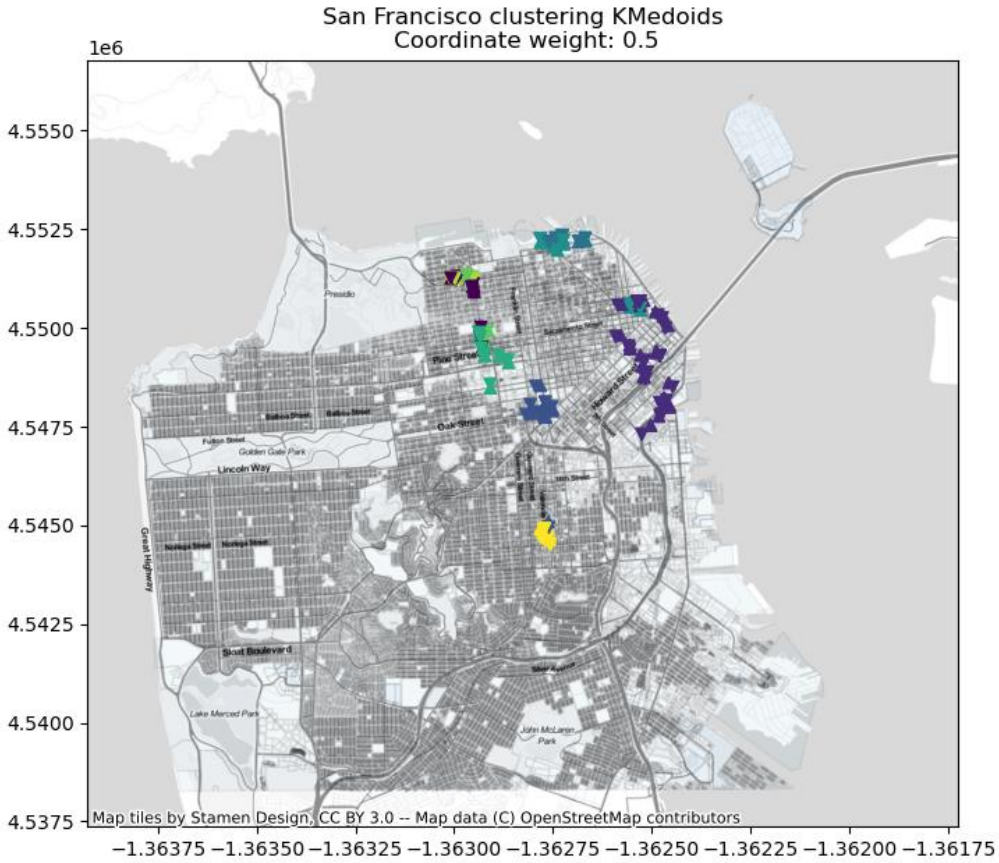


Figura 143

Si nota che il *K-Medoids* identifica come cluster parcheggi appartenenti alla stessa area geografica, ma tale dimensione non prevale nettamente rispetto alle altre (come accade per *l'Agglomerative*). Infatti, ad esempio si nota che i parcheggi riportati nella Figura 144 (che non è altro che uno zoom dell'area riquadrata nella mappa precedente) appartengono a cluster diversi, anche se hanno stessa metrica e posizione geografica simile; questo perché sono caratterizzati da tempi diversi. Questa clusterizzazione è coerente con l'intenzione di dare equa importanza alle dimensioni temporali e spaziali (motivo per il quale è stato impostato $\boldsymbol{w_1 = w_2 = 0,50}$).

Di seguito si riporta uno zoom della mappa riportata precedentemente; si nota che parcheggi anche vicini geograficamente appartengono a cluster diversi, in quanto raggiungono uno stesso tasso di occupazione (pari ad 1) in istanti temporali diversi.

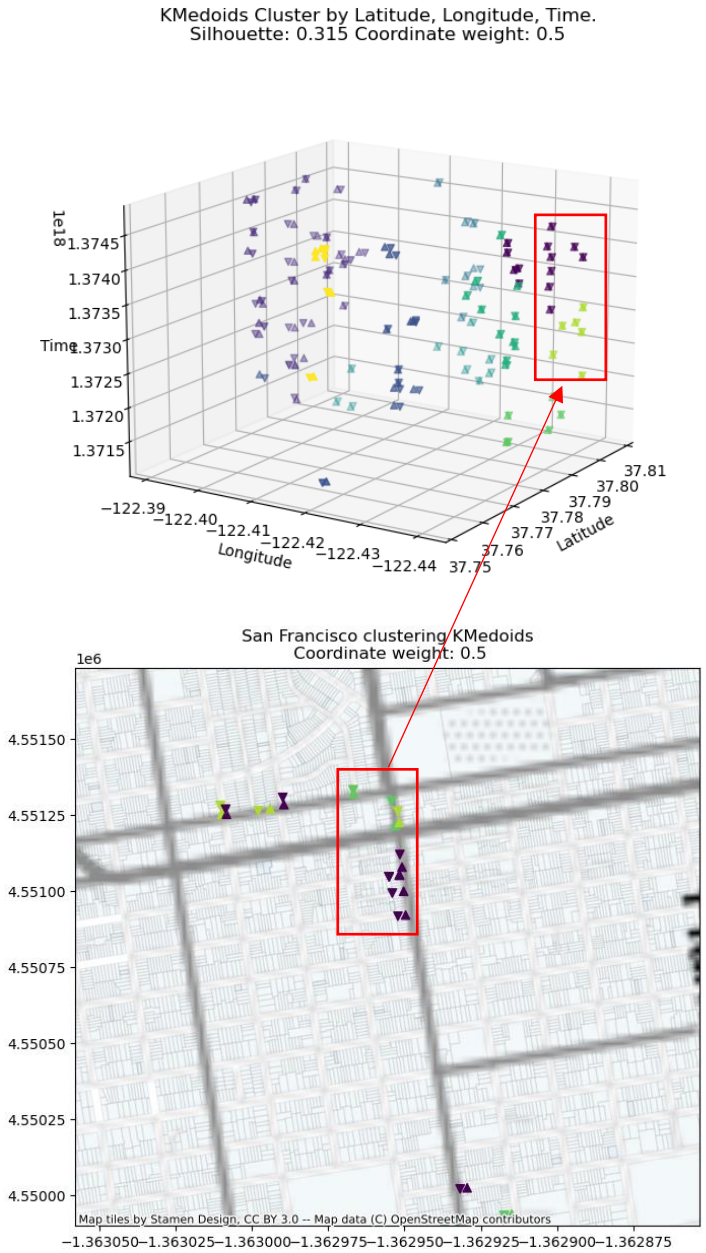


Figura 144

In conclusione, dal confronto tra l'*Agglomerative clustering* e il *K-Medoids* è possibile affermare che in questo caso specifico il *K-Medoids* tiene maggiormente conto della dimensione temporale, individuando un numero maggiore di *cluster* (9) con una maggiore similarità di spazio, tempo e tasso di occupazione. Con l'*Agglomerative* vengono individuati meno *cluster* (4) con una prevalente similarità della posizione geografica.

6.3.2.2. Risultati cluster con $w_1 = 0$; $w_2 = 1$

Al fine di identificare dei *cluster* solo sulla base di un tasso di occupazione simile in istanti temporali prossimi, senza considerare la localizzazione geografica, è stato impostato $\boldsymbol{w_1 = 0}$ e $\boldsymbol{w_2 = 1}$. Il numero di cluster è stato individuato automaticamente, in modo da massimizzare l'indice di *Silhouette*.

Agglomerative clustering – partizione SF Park con tasso di occupazione TRA 100% e 95%

L'algoritmo *Agglomerative clustering* con w_1=0 ha individuato *3 cluster* (1 in meno rispetto allo scenario precedente) con un indice di *Silhouette* pari a 0.666. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 145 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione rilevante accanto al tempo ai fini dell'individuazione dei cluster

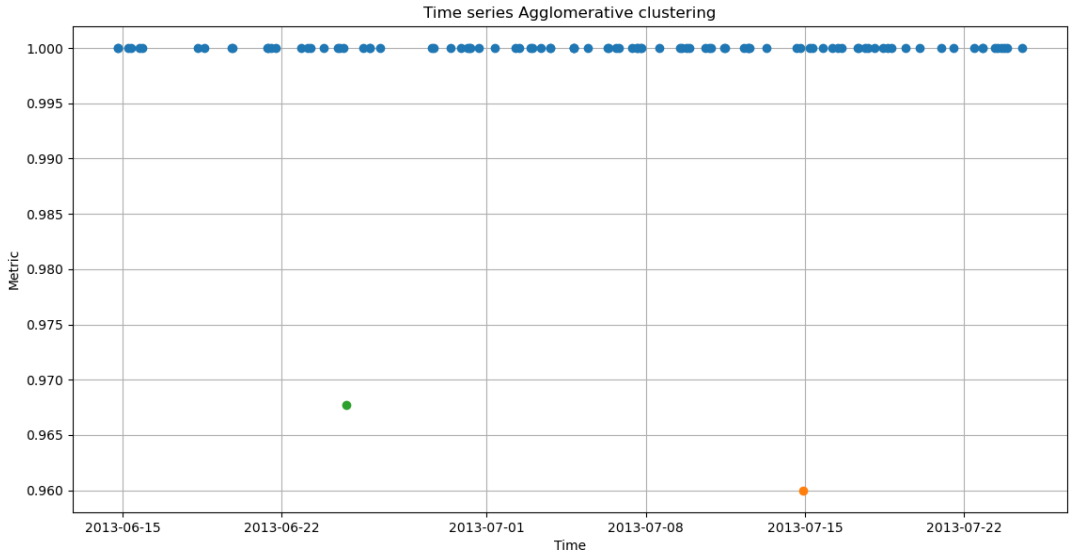


Figura 145

b) la Figura 146 mostra i grafici in 3 dimensioni della latitudine, longitudine e metrica/tempo:
si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1=0***), questa non viene considerata come discriminante per l'individuazione dei cluster. Al contrario c'è una differenziazione rispetto alla metrica. Rispetto alla coordinata temporale non sembra esserci una suddivisione, in quanto i dati sono distribuiti in maniera relativamente omogenea nel tempo.

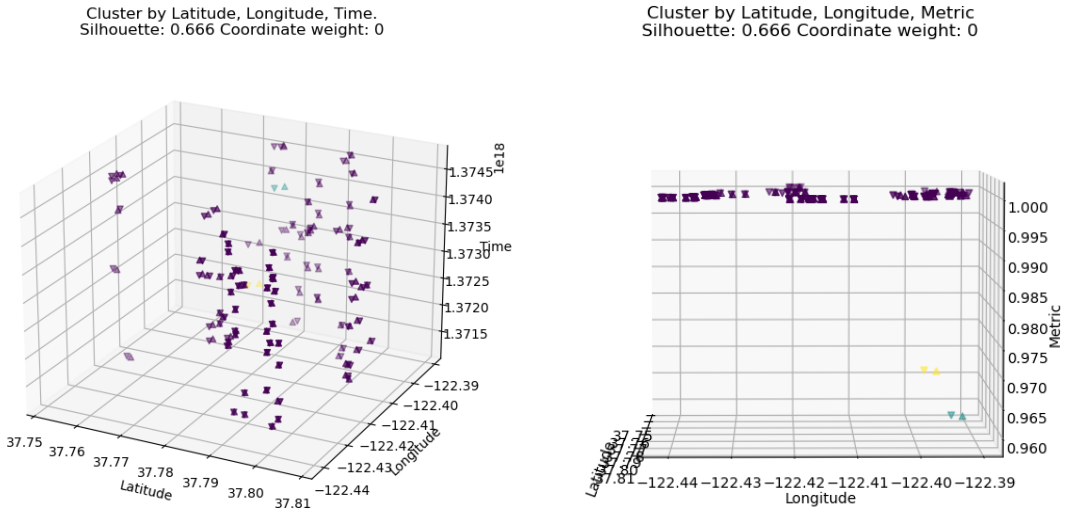


Figura 146

c) la Figura 147 la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1=***0), questa non viene considerata come discriminante per l'individuazione dei cluster.

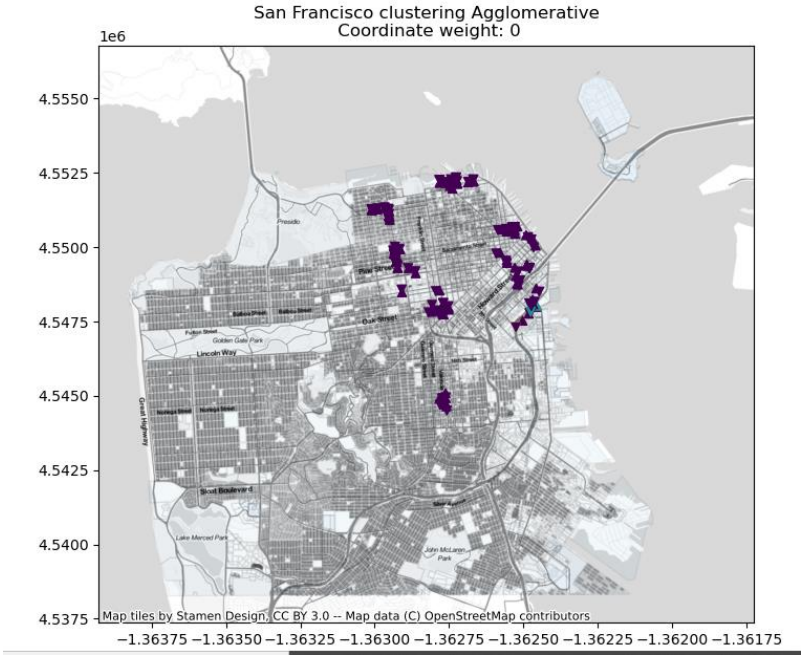


Figura 147

K-Medoids– partizione SF Park con tasso di occupazione tra 100% e 95%

L'algoritmo *K-Medoids* con w_1=0 ha individuato *3 cluster* (6 in meno rispetto allo scenario precedente) con un indice di *Silhouette* pari a 0,527. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 148 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:

si nota che l'istante temporale di misura del tasso di occupazione è una dimensione rilevante ai fini dell'individuazione dei *cluster*

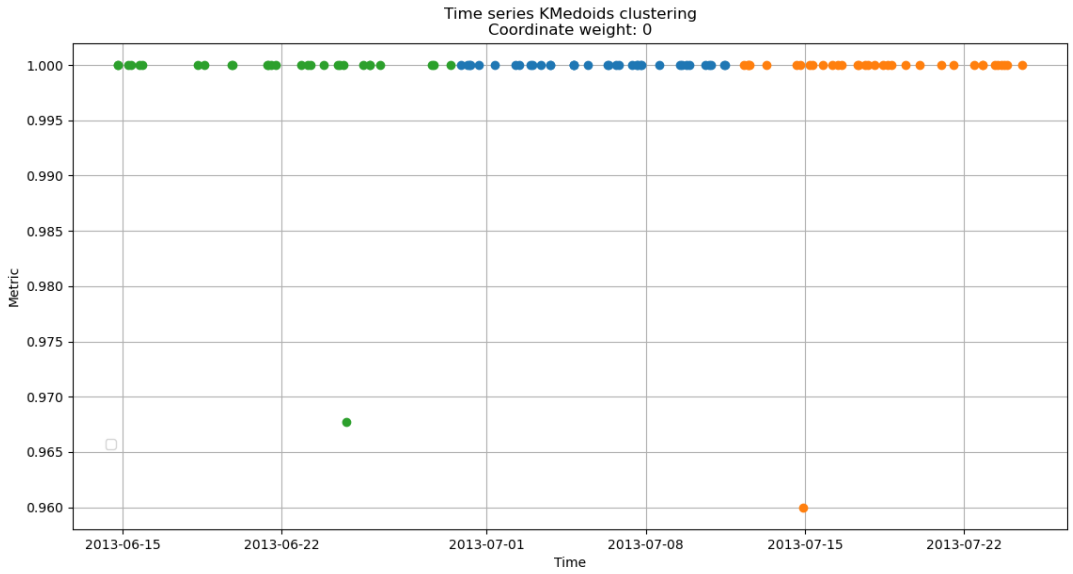


Figura 148

b) la Figura 149 mostra il grafico in 3 dimensioni della latitudine, longitudine e metrica:
si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1*=**0), questa non viene considerata come discriminante per l'individuazione dei cluster.

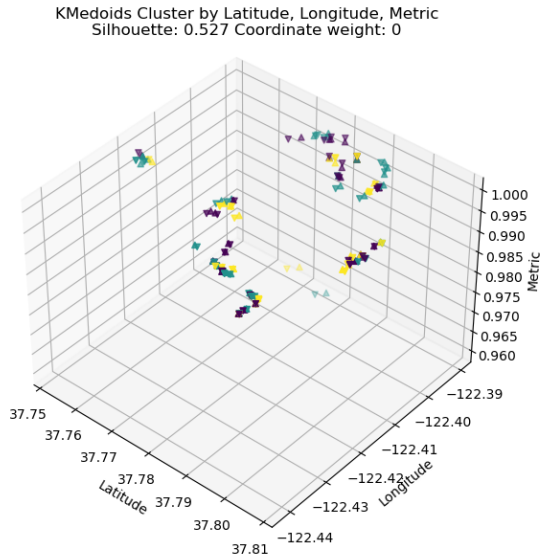


Figura 149

c) la Figura 150 mostra il grafico in 3 dimensioni della latitudine, longitudine e time:
si nota che avendo imposto un valore unitario (massimo) come peso della coordinata temporale (***w1*=**1), questa sembra essere la discriminante principale per l'individuazione dei cluster.

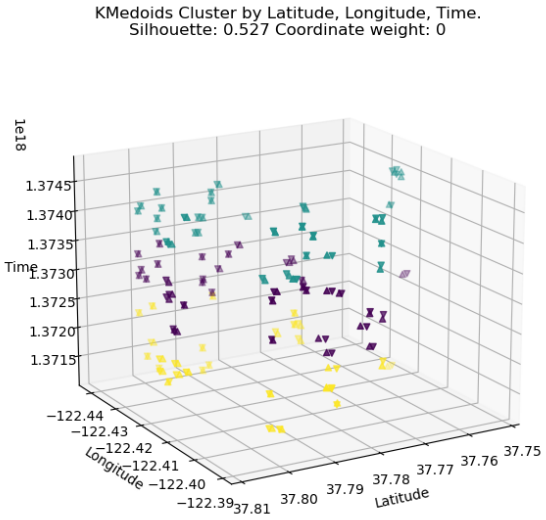


Figura 150

d) la Figura 151 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1=***0), questa non viene considerata come discriminante per l'individuazione dei cluster.

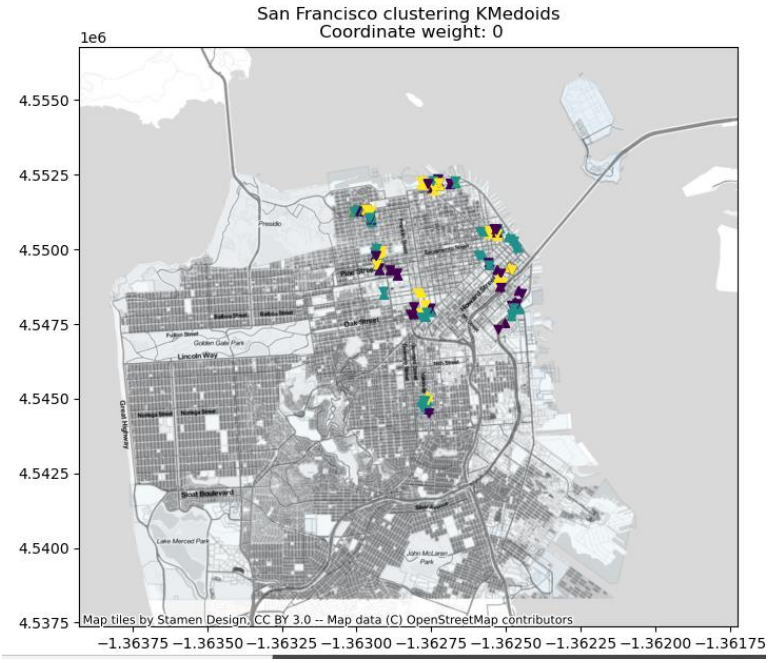


Figura 151

6.3.3. *Partizione con* capacity utilization rate compreso tra 80% e 60%

A partire dal dataset di partenza, sono stati isolati i punti con *capacity utilization rate* compreso tra 0.80 e 0.60**,** che sono risultati essere pari a 1.083.540 (pari al 21.3% del totale).

Sulla base di tale partizione, è stato estratto un campione casuale, stratificato rispetto al *segment ID,* costituito da 100 items riportato nella Figura 152.

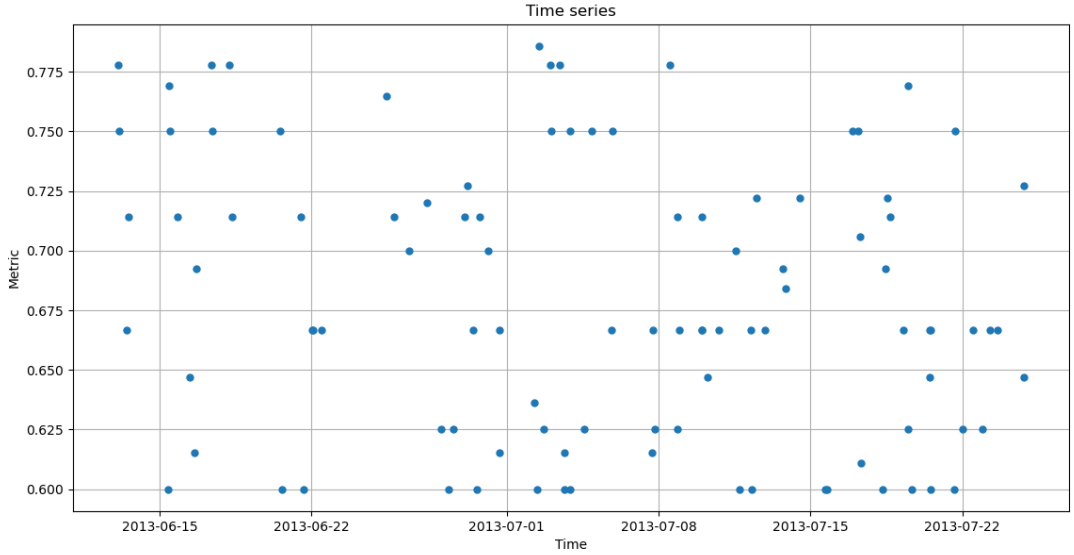


Figura 152

6.3.3.1. Risultati cluster con $w_1 = w_2 = 0,50$

Agglomerative clustering – partizione SF Park con tasso di occupazione tra 80% e 60%

L'algoritmo *Agglomerative clustering* con $w_1 = w_2 = 0.50$ ha individuato *9 cluster* con un indice di *Silhouette* pari a 0.333. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 153 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione discretamente ai fini dell'individuazione dei cluster. Ad esempio, si nota che il cluster in rosa è concentrato nella zona con tasso di occupazione maggiore del 75%; al contrario il cluster in rosso è concentrato nella zona con tasso di occupazione compreso tra il 60% e il 62,5%.

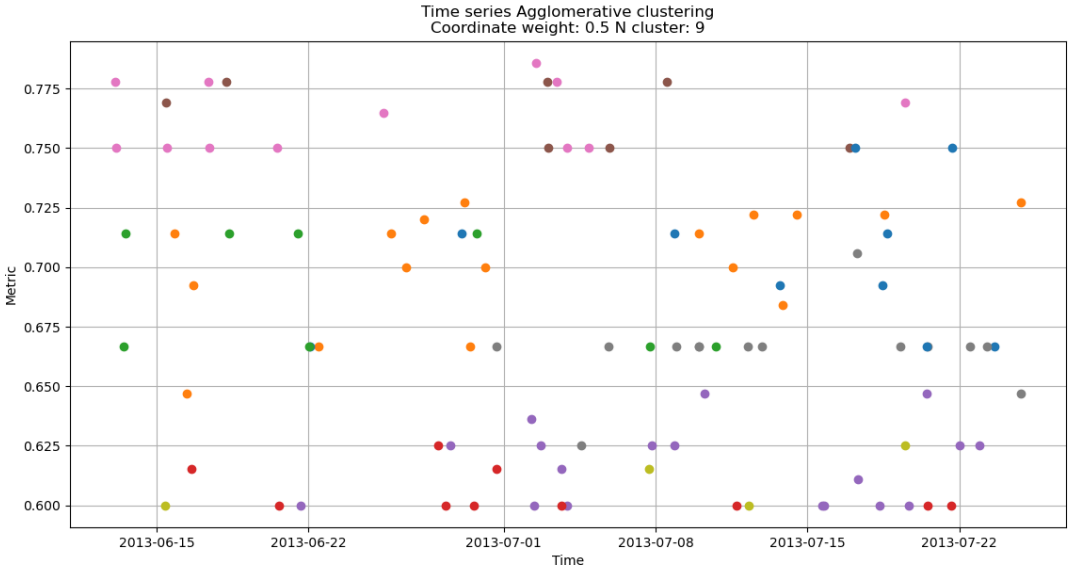


Figura 153

b) la Figura 154 mostra il grafico in 3 dimensioni della latitudine, longitudine e metrica:
come già sottolineato nel punto precedente, anche dal grafico di seguito si nota che la metrica è una discriminante per l'individuazione dei cluster. Ad esempio, il cluster in alto in turchese è caratterizzato da valori di metrica superiori al 75% circa.

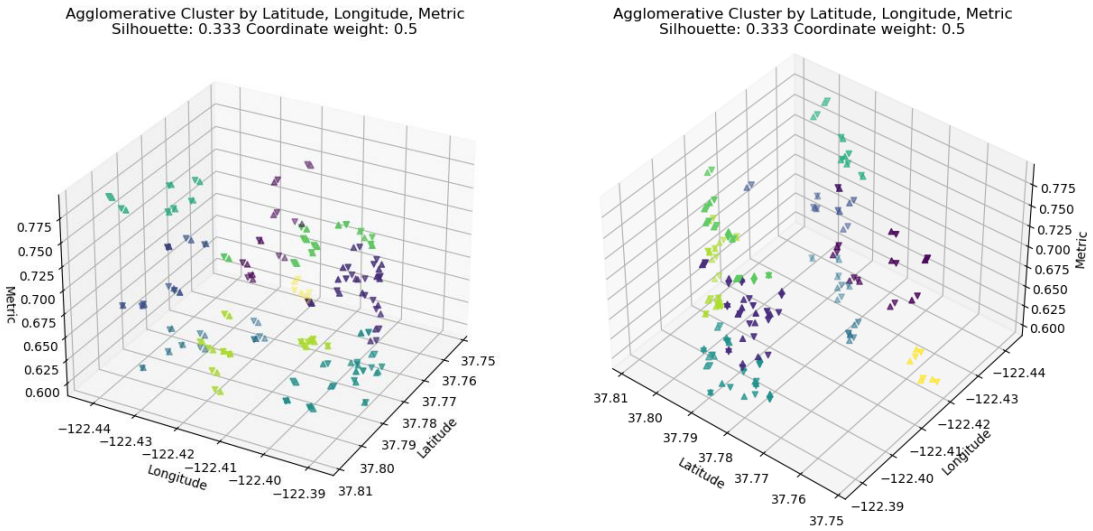


Figura 154

c) la Figura 155 mostra il grafico in 3 dimensioni della latitudine, longitudine e tempo:
anche se è stato impostato un peso della componente temporale pari a $\mathbf{w_2 = 0,50}$ non sembra essere una componente particolarmente rilevante, in quanto i dati sono distribuiti in modo relativamente uniforme.

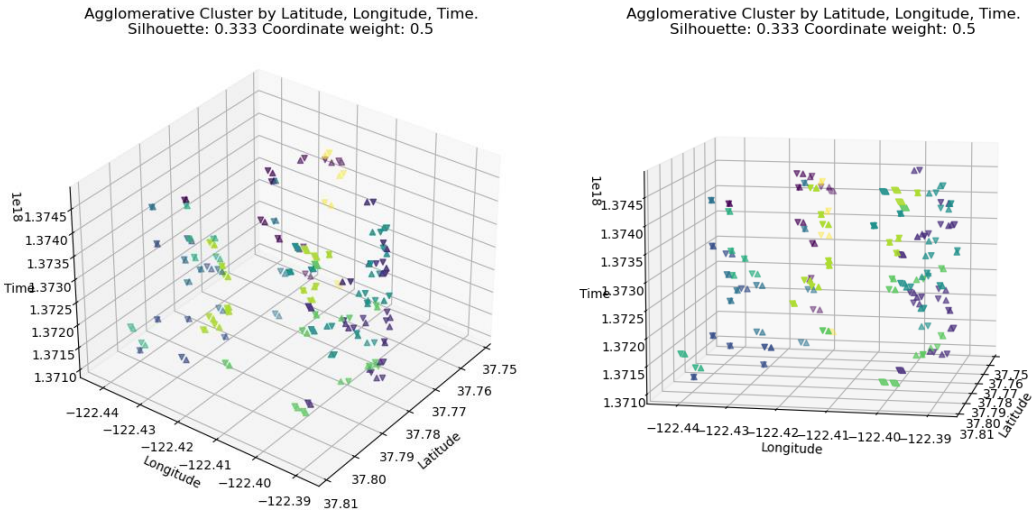


Figura 155

d) la Figura 156 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore di 0.50 come peso della coordinata spaziale (***w1*=**0.50), questa rappresenta una discriminante per l'individuazione dei cluster (accanto alla metrica e in misura inferiore al tempo). Ad esempio, si nota che il cluster in giallo è concentrato nella zona centrale della città mentre quello in verde è concentrato lungo la costa a nord.

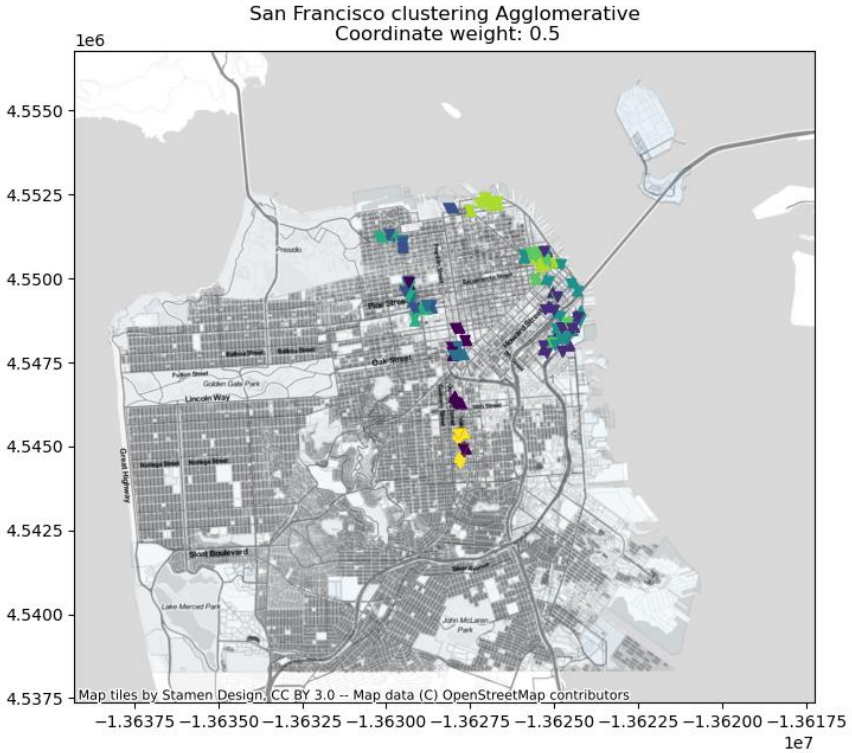


Figura 156

K-Medoids– partizione SF Park con tasso di occupazione compreso tra 80% e 60%

L'algoritmo *K-Medoids* con $w_1 = w_2 = 0.50$ ha individuato *3 cluster* con un indice di *Silhouette* pari a 0.265.

Il *K-Medoids* ha quindi individuato 6 cluster in meno rispetto a quelli intercettati dall'*Agglomerative,* con un indice di *Silhouette* simile.

Anche in tal caso, al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 157 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione discretamente rilevante ai fini dell'individuazione dei cluster. Ad esempio, si nota che il cluster in azzurro è concentrato nella zona con tasso di occupazione compreso maggiore del 70%.

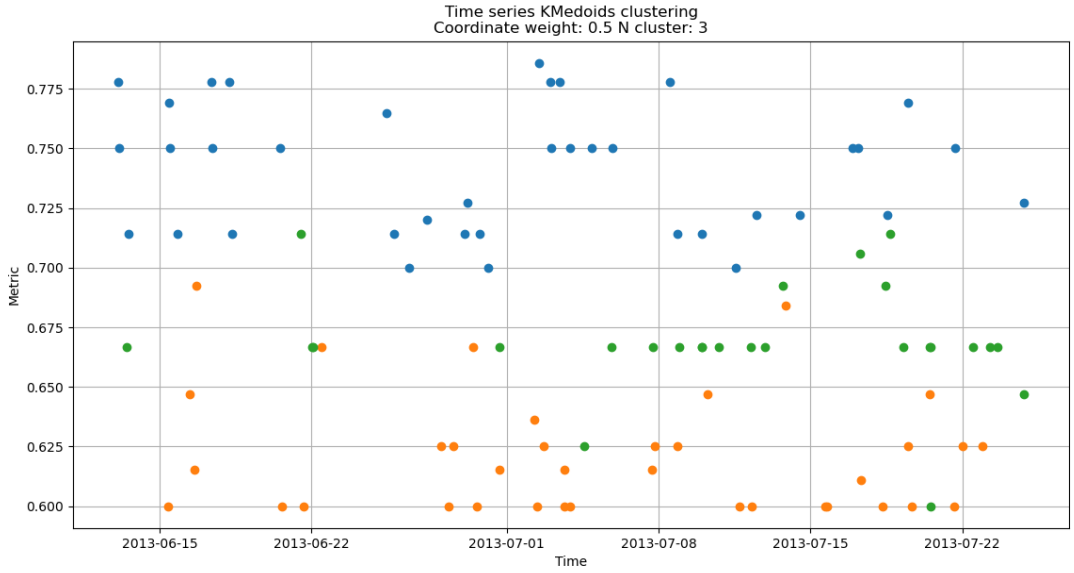


Figura 157

b) la Figura 158 mostra il grafico in 3 dimensioni della latitudine, longitudine e metrica:
come già sottolineato nel punto precedente, anche dal grafico di seguito si nota che la metrica è una discriminante rilevante per l'individuazione dei cluster. Ad esempio, il cluster in alto in viola è caratterizzato da valori di metrica superiori a circa il 75% circa.

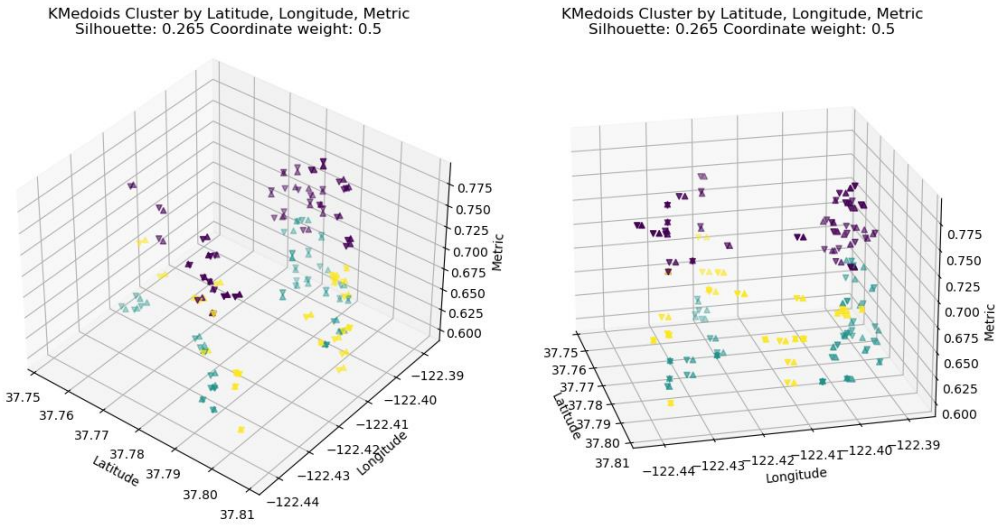


Figura 158

c) la Figura 159 mostra il grafico in 3 dimensioni della latitudine, longitudine e tempo:
anche se è stato impostato un peso della componente temporale pari a $\mathbf{w_2 = 0,50}$ non sembra essere una componente particolarmente rilevante, in quanto i dati sono distribuiti in modo relativamente uniforme.

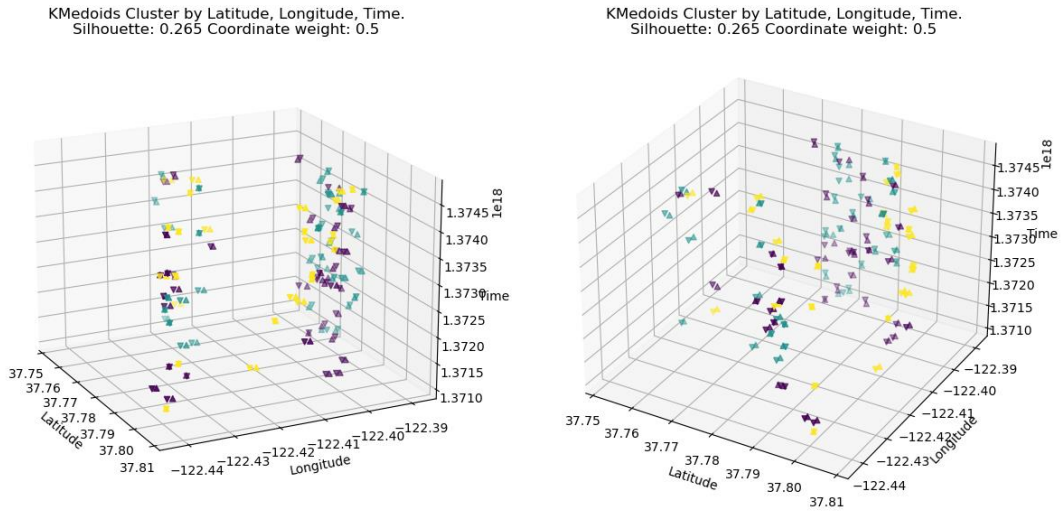


Figura 159

d) la Figura 160 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore di 0.50 come peso della coordinata spaziale (***w1*=**0.50), questa rappresenta una discriminante per l'individuazione dei cluster (accanto alla metrica e in misura inferiore al tempo). Ad esempio, si nota che il cluster in giallo è concentrato nella zona nord della città.

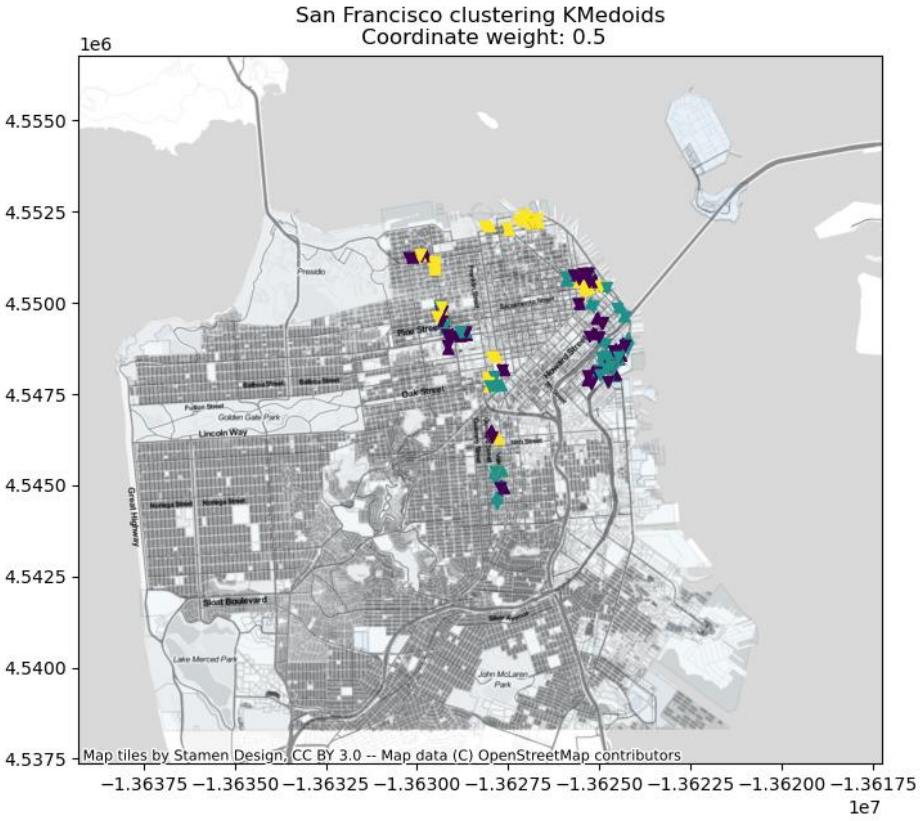


Figura 160

6.3.3.2. Risultati cluster con $w_1 = 0$; $w_2 = 1$

Agglomerative clustering – partizione SF Park con tasso di occupazione tra 80% e 60%

L'algoritmo *Agglomerative clustering* con w_1=0 ha individuato *9 cluster* (analogamente al numero di cluster individuato con w_1= w_2= 0.50) con un indice di *Silhouette* pari a 0.474. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 161 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione nettamente rilevante ai fini dell'individuazione dei cluster:

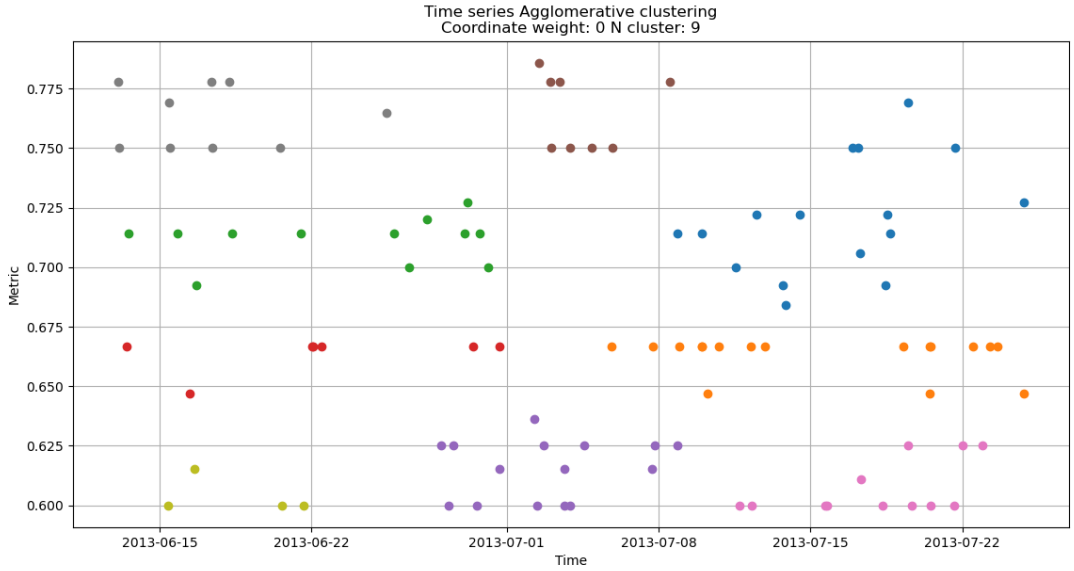


Figura 161

b) la Figura 162 e la Figura 163 mostrano i grafici in 3 dimensioni della latitudine, longitudine e metrica/tempo:
si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1*=**0), questa non viene considerata come discriminante per l'individuazione dei cluster. Al contrario c'è una differenziazione rispetto alla metrica e al tempo.

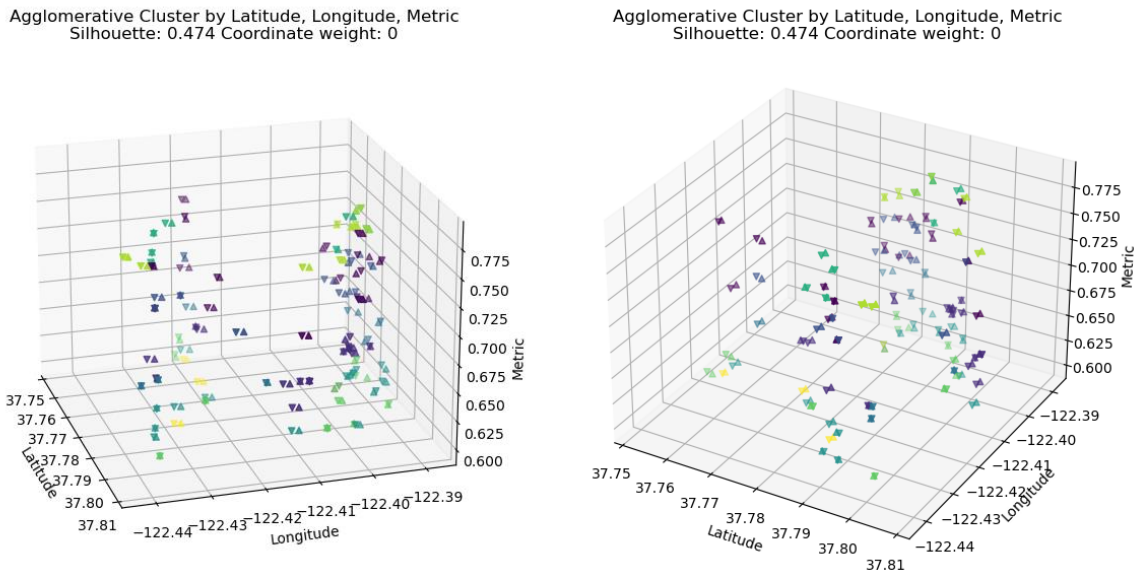


Figura 162

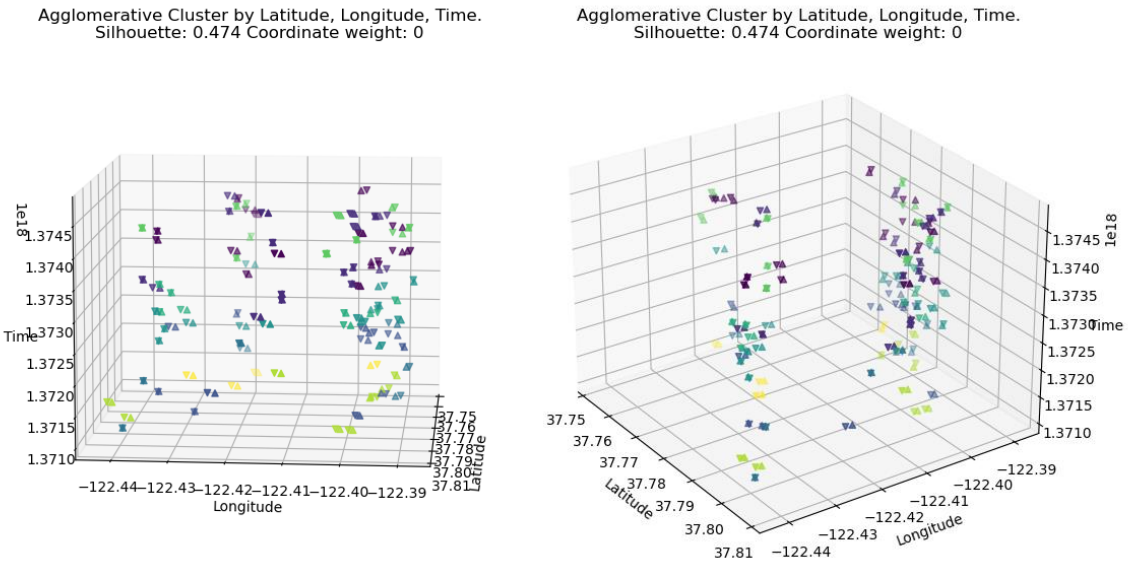


Figura 163

c) la Figura 164 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1=***0), questa non viene considerata come discriminante per l'individuazione dei cluster.

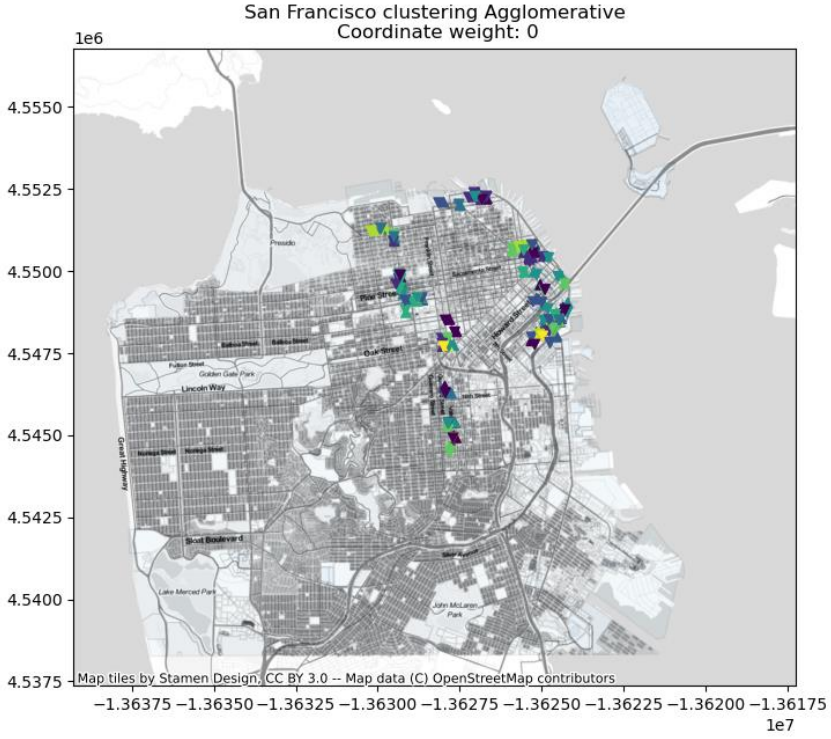


Figura 164

K-Medoids– partizione SF Park con tasso di occupazione compreso tra 80% e 60%

L'algoritmo *K-Medoids* con $w_1 = w_2 = 0.50$ ha individuato *3 cluster* con un indice di *Silhouette* pari a 0.384

Il *K-Medoids* ha quindi individuato 6 cluster in meno rispetto a quelli intercettati dall'*Agglomerative,* con un indice di *Silhouette* leggermente inferiore.

Anche in tal caso, al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 165 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione discretamente rilevante ai fini dell'individuazione dei cluster. Ad esempio, si nota che il cluster in arancione è concentrato nella zona con tasso di occupazione compreso tra il 60% e il 67.5%.

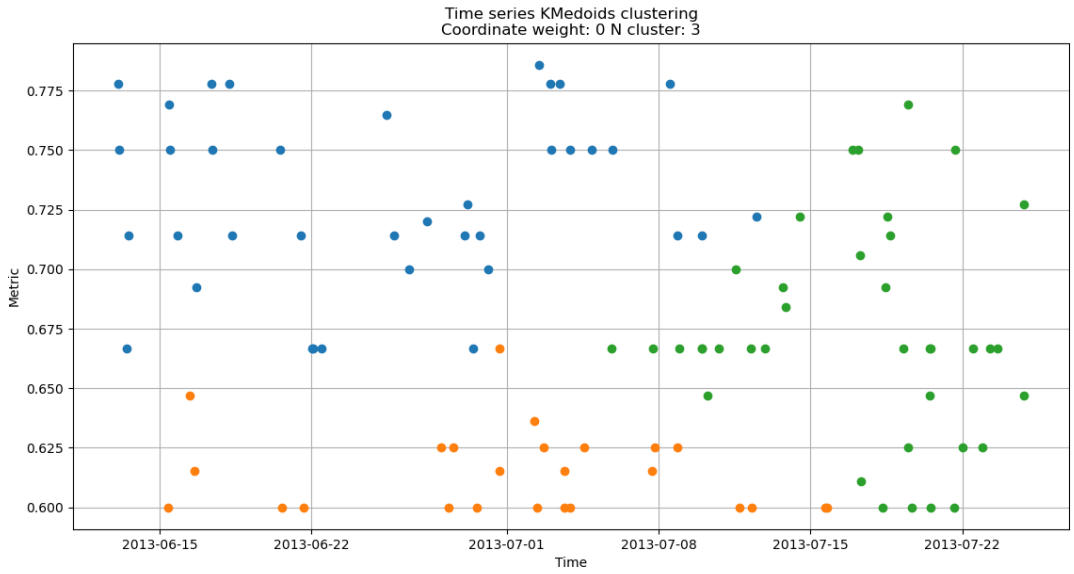


Figura 165

b) la Figura 166 e la Figura 167 mostrano i grafici in 3 dimensioni della latitudine, longitudine e metrica/tempo:
si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1*=**0), questa non viene considerata come discriminante per l'individuazione dei cluster. Al contrario c'è una differenziazione rispetto al tempo e alla metrica.

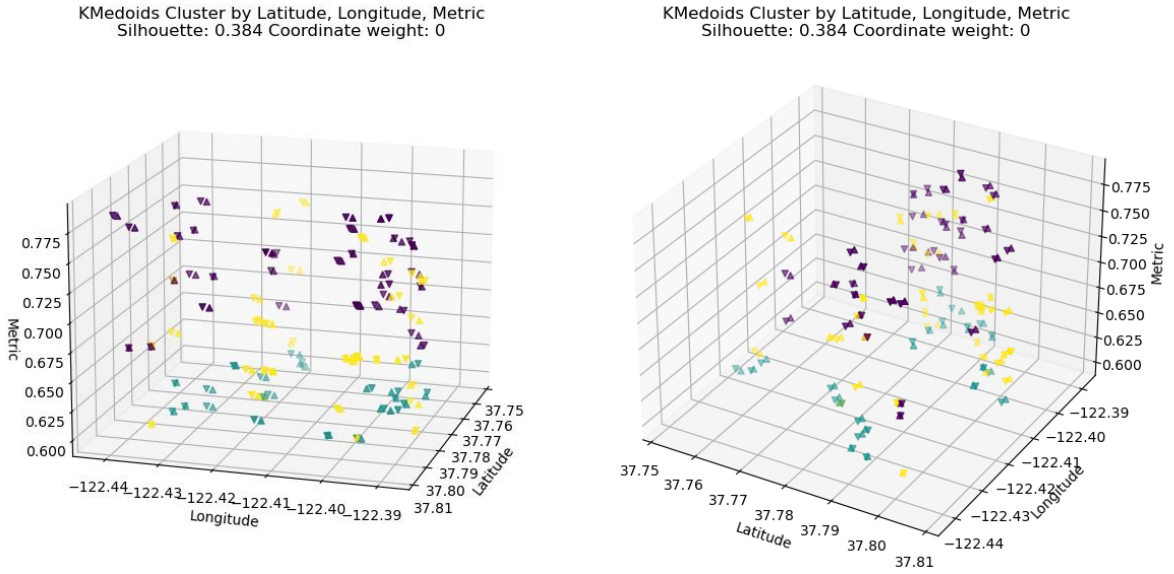


Figura 166

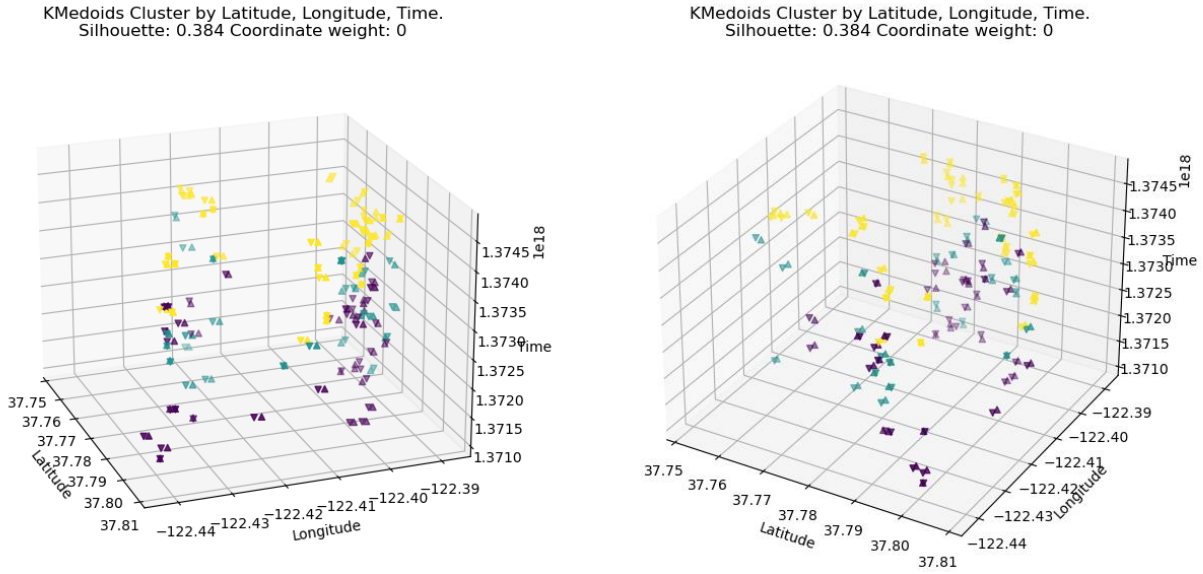


Figura 167

c) la Figura 168 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
anche in tal caso si nota che avendo imposto un valore nullo come peso della coordinata spaziale (***w1*=**0), questa non viene considerata come discriminante per l'individuazione dei cluster.

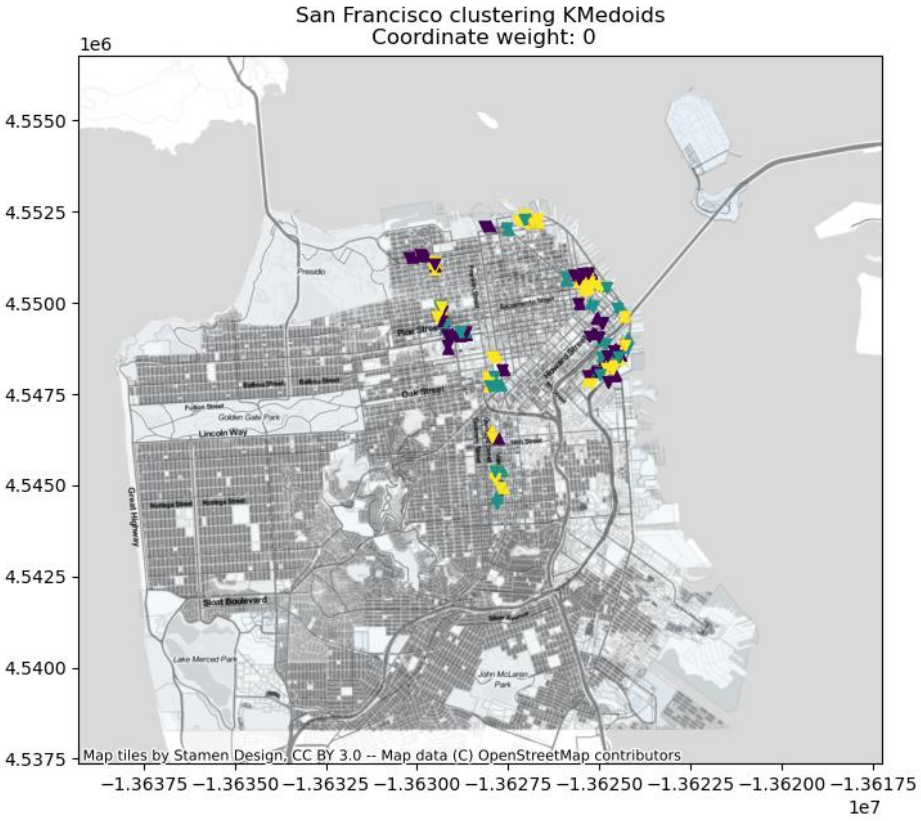


Figura 168

6.3.4. Clustering e prezzo dei parcheggi

Se si avesse potenza computazionale sufficiente ad analizzare e *clusterizzare* l'intero dataset, si potrebbe ottenere una informazione analoga rispetto a quella ottenuta nei paragrafi precedenti ma completa nel tempo, nel senso che per ogni parcheggio sarebbe nota l'appartenenza al *cluster* sulla base del tempo, della localizzazione e del tasso di occupazione. Tale informazione potrebbe essere utilizzata per variare il prezzo del parcheggio nel corso della giornata, con l'idea di assegnare il prezzo in modo proporzionale al tasso di occupazione rappresentativo del *cluster*, ovvero un prezzo maggiore ai cluster a cui è assegnato un tasso di occupazione più alto, poiché questo indica elevata domanda del parcheggio (e viceversa). La frequenza del cambiamento della tariffa è variabile per ogni parcheggio e avviene ogni qualvolta, nell'arco della giornata, il parcheggio cambia *cluster* di appartenenza.

Modificando il peso da assegnare alle coordinate spaziali e temporali sarebbe possibile dare più o meno importanza ai vari fattori (ad esempio, scegliendo di identificare *cluster* solo sulla base del tasso di occupazione nel tempo, scegliendo di non considerare la localizzazione geografica).

In tal modo di definisce un modello dei prezzi dinamico, che varia sulla base dell'andamento del tasso di occupazione nel tempo, ma anche in funzione del luogo. Infatti, modificando i pesi delle varie dimensioni si potrebbe facilmente modificare l'influenza di tempo, spazio e tasso di occupazione (e/o altre metriche).

Considerando il valore medio della metrica (in questo caso del *capacity utilization ratio*), viene fatto un *ranking* (crescente, dal valore più grande). Quindi $\boldsymbol{r_i}$ rappresenta la posizione del cluster i, con $\boldsymbol{r} \in [\boldsymbol{1}, \boldsymbol{n}]$ dove $\boldsymbol{n}$ è il numero di cluster individuati. Il prezzo $\boldsymbol{p_i}$ di tutti i parcheggi appartenenti al cluster i-esimo, sarà pari a:

Equazione 26: Prezzo in funzione del cluster

$$\boldsymbol{p_i = \frac{p_{max} - p_{min}}{n - 1} * (r_i - 1) + p_{min}}$$

dove:

$\boldsymbol{p_{max}}$: prezzo massimo per il parcheggio

$\boldsymbol{p_{min}}$: prezzo minimo per il parcheggio

$\boldsymbol{n}$: numero di cluster individuati

$\boldsymbol{r_i}$: posizione del cluster *i* nel *ranking* **crescente** sulla base dei valori medi dei cluster del *capacity utilization ratio.*

Il sistema dinamico dei prezzi del parcheggio si basa sul fatto che l 'elasticità della domanda di quest'ultimo- ovvero il grado in cui le variazioni di prezzo influiscono sull'occupazione dei parcheggi - varia a seconda delle diverse località e orari della giornata (a causa dei diversi scopi del viaggio ed esigenze del conducente) [42]. Tale teoria è proprio alla base del progetto pilota *SF-PARK,* in cui è stato sperimentato il sistema dinamico dei prezzi tra il 2011 e il 2013 monitorando circa il 25% dei parcheggi di San Francisco. Ogni giorno è stato diviso in 3 *pricing-time band* (9:00-12:00, 12:00-15:00, 15:00-18:00), in cui il prezzo poteva variare tra 25 *cents* a $6 all'ora. I prezzi dei giorni del weekend sono stati fissati separatamente da quelli lavorativi. Ciò significa che, ad esempio, il posto di uno stesso parcheggio poteva avere prezzi diversi tra mattina e pomeriggio. Per ottimizzare gli utilizzi dei parcheggi, il progetto pilota *SF-PARK* regolava periodicamente i prezzi in eccesso o in difetto, in modo tale che si raggiungesse il target di tasso di occupazione di tutti i parcheggi tra il 60% e l'80%. Le variazioni di prezzo incoraggiavano i conducenti a parcheggiare in aree sottoutilizzate, riducendo la domanda in aree sovrautilizzate [43].

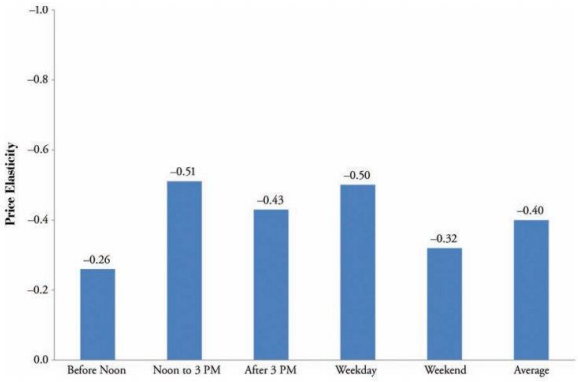


Figura 169 Elasticità del prezzo dei parcheggi in funzione della fascia oraria, relativamente al progetto pilota SF-PARK [42]

La Figura 169 mostra che i conducenti sono più sensibili alle variazioni di prezzo nella fascia oraria tra le 12:00 e le 15:00, mentre sono meno sensibili tra le 9:00 e le 12:00; in quest'ultima fascia oraria il prezzo può aumentare di più rispetto alla fascia 12:00-15:00, poiché si registrerà una riduzione della domanda più bassa. [42]

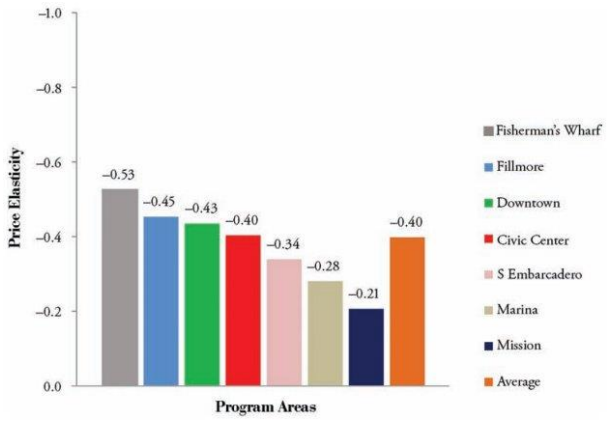


Figura 170 Elasticità del prezzo dei parcheggi in funzione dei quartieri, relativamente al progetto pilota SFpark [23]

La Figura 170 mostra che le variazioni di prezzo hanno un impatto sulla domanda più basso nelle zone residenziali, come *Mission* e *Marina,* rispetto ai *retail district* come *Fisherman's Wharf* e *Fillmore*. [42]

Per un approfondimento circa gli aspetti economici del progetto *SF-Park* fare riferimento al paragrafo 6.3.5.

Ad esempio, per i parcheggi che registrano un tasso di occupazione maggiore del target, quindi maggiore di 80%, è necessario aumentare il prezzo, al fine di ridurre la domanda. Al contrario per i parcheggi che registrano un tasso di occupazione minore del target, quindi minore di 60% è necessario ridurre il prezzo, al fine di aumentare la domanda. Se si volessero discriminare all'interno di questa fascia i parcheggi, al fine di fissare un prezzo diverso per ogni gruppo, si potrebbe applicare un algoritmo di *clustering* che tenga conto della posizione, del tasso di occupazione registrato e del tempo in cui tale tasso viene raggiunto.

Di seguito si riporta un esempio di applicazione di tale metodologia per l'identificazione dei prezzi dei parcheggi a partire dai *cluster* ottenuti.

Al fine di determinare i prezzi sulla base del cluster, è stata prima selezionata la partizione dei parcheggi con **tasso di occupazione compreso tra 80% e 95%**. Successivamente, è stato estratto un campione casuale (stratificato rispetto al segment ID) contenete 100 items. Di seguito nella Figura 171 il grafico della metrica (tasso di occupazione) nel tempo:

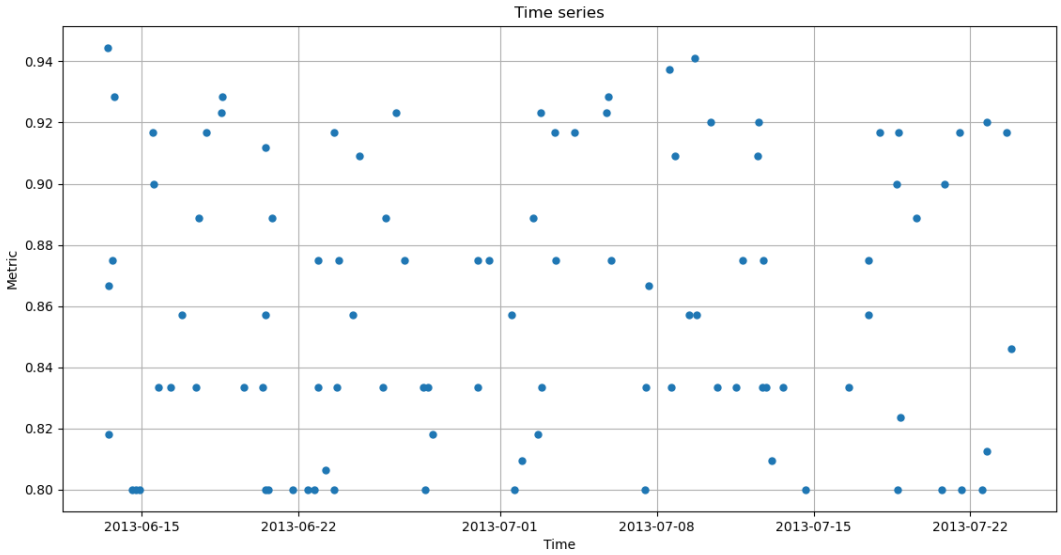


Figura 171

È stato applicato l'algoritmo di *clustering Agglomerative,* con l'obiettivo di individuare gruppi di parcheggi simili rispetto allo spazio e rispetto al tasso di occupazione nel tempo. Di conseguenza è stato fissato un peso della componente spaziale pari a quello della componente temporale ($\boldsymbol{w_1 = w_2 = 0,50}$). Per ogni *cluster* identificato è stato calcolato il valore medio del tasso di occupazione, sulla base del quale i *cluster* sono stati ordinati in modo crescente, per poi attribuire i vari prezzi sulla base dell'ordinamento. Ipotizzando che il prezzo massimo dei parcheggi coinvolti sia 1 $, l'obiettivo è aumentare il prezzo per ridurre il tasso di occupazione. Quindi si fissa un prezzo minimo maggiore di 1$, pari ad esempio a 1,5$. Si vuole attribuire tale prezzo minimo scelto a quel cluster che ha un tasso di occupazione medio più basso, poiché è sufficiente una minima riduzione di tasso di occupazione per raggiungere la fascia target. Analogamente, si fissa un prezzo massimo, pari ad esempio a 2.5$, da attribuire al parcheggio con un tasso di occupazione medio più alto, poiché è necessaria una massima riduzione del tasso di occupazione. I risultati ottenuti sono riassunti nella Tabella 12. Si ipotizza di avere:

1) una domanda con elasticità puntuale rispetto al prezzo pari a **-0.20 (ε)**
2) tutti i parcheggi hanno inizialmente uno stesso prezzo pari a **1$**

Si fissa un prezzo minimo pari a **1.5$** e un prezzo massimo pari a **2.5$.**

Tabella 12

cluster	***Tasso di occupazione medio (Q)***	***Rank***	*Price con Agglomerative* ***(p)***	***delta price (dp)***	***Tasso di occupazione aggiustato (Q')***
6	81,43%	1	1,50	0,50	73,29%
7	81,56%	2	1,64	0,64	71,07%
1	82,57%	3	1,79	0,79	69,59%
3	86,81%	4	1,93	0,93	70,69%
2	87,97%	5	2,07	1,07	69,12%
4	91,21%	6	2,21	1,21	69,06%

0	91,95%	7	2,36	1,36	66,99%
5	92,23%	8	2,50	1,50	64,56%

I valori di *"price con agglomerative (p)"* sono stati ottenuti applicando l'Equazione 26.

I valori del "tasso di occupazione aggiustato (Q')" sono stati ottenuti applicando la seguente equazione:

Equazione 27: Tasso di occupazione "aggiustato" in base alla correzione del prezzo

$$Q' = \left(1 + \varepsilon * \frac{dp}{p}\right) Q$$

Si nota che in questo modo il tasso di occupazione "aggiustato" rientra nella fascia target tra il 60% e l'80%.

Applicando l'*agglomerative clustering* sono stati identificati 8 *cluster,* con un indice di *Silhouette* pari a **0.315**. Al fine di rappresentare in maniera esaustiva i *cluster* ottenuti e le diverse dimensioni, di seguito si riporta:

a) la Figura 172 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione più o meno rilevante; ad esempio, si nota che il cluster in arancione è caratterizzato da un tasso di occupazione tra l'80% e l'86%.

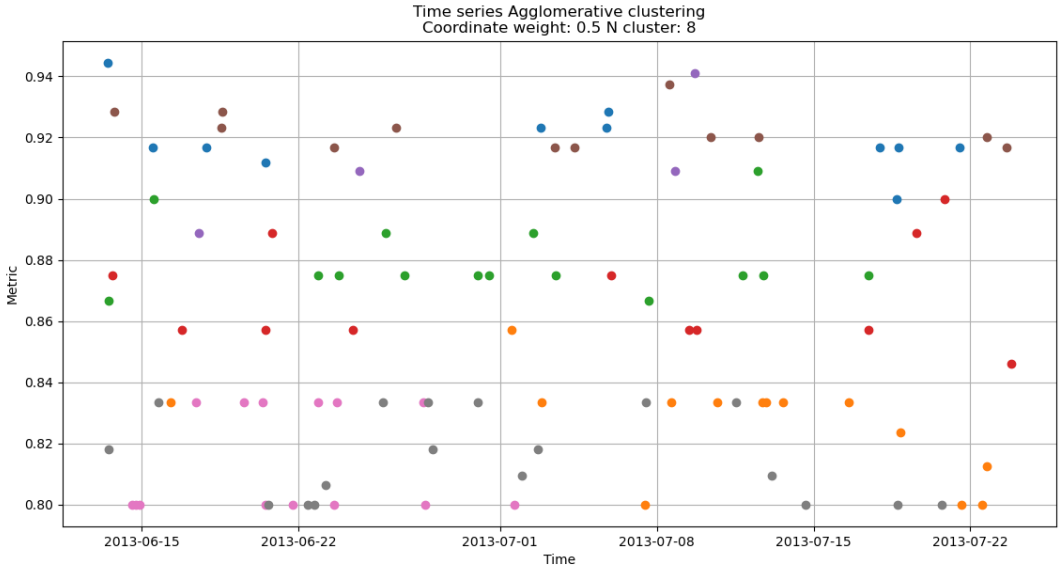


Figura 172

b) la Figura 173 e la Figura 174 mostrano i grafici in 3 dimensioni della latitudine, longitudine e metrica/tempo:
si nota che c'è una differenziazione maggiore rispetto alla metrica e una lieve differenziazione in base al tempo.

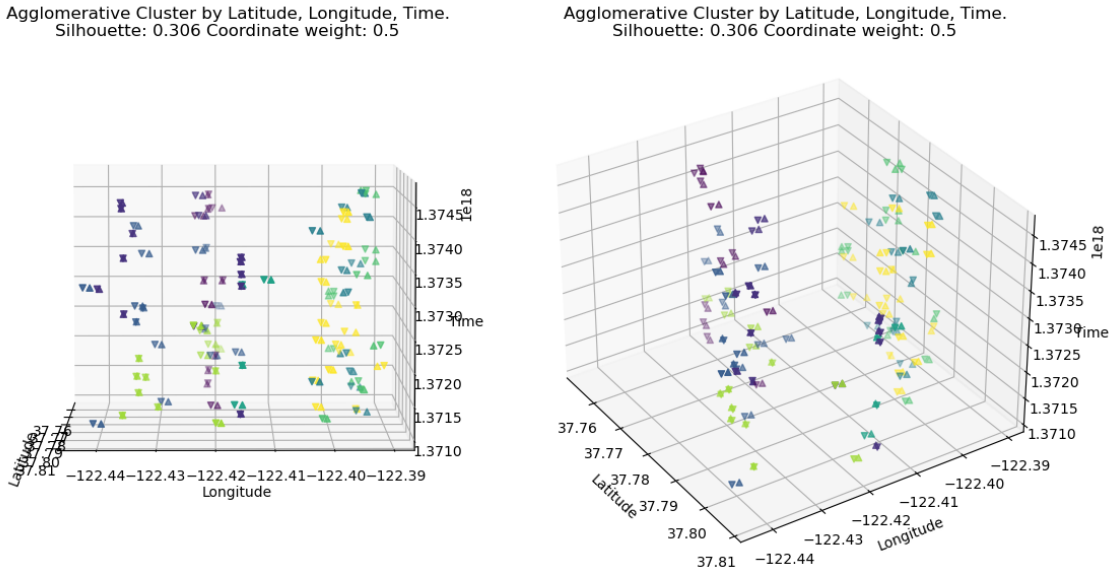


Figura 173

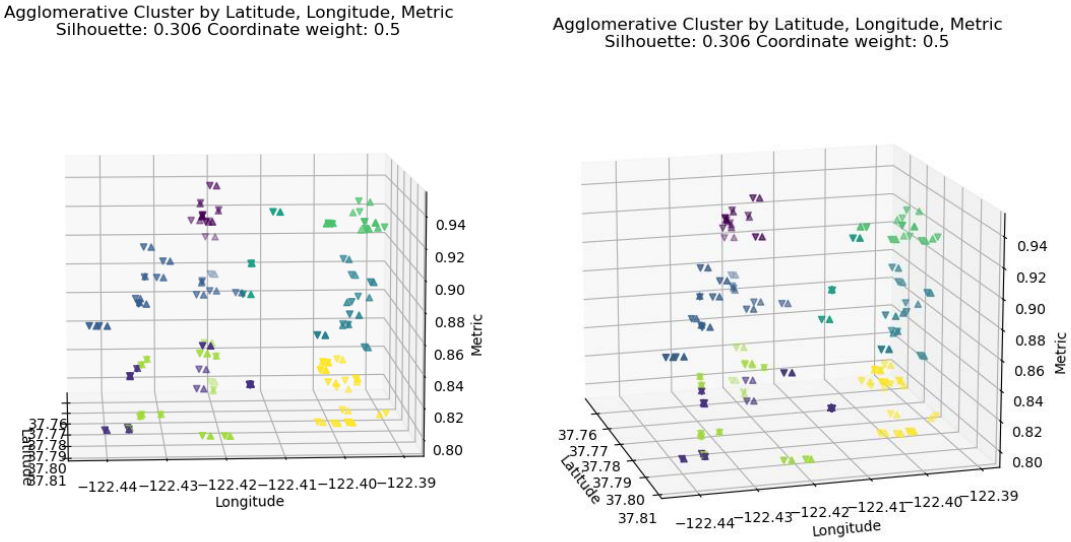


Figura 174

c) la Figura 175 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:*
si può notare una leggera discriminazione in base allo spazio al fine dell'identificazione dei cluster.

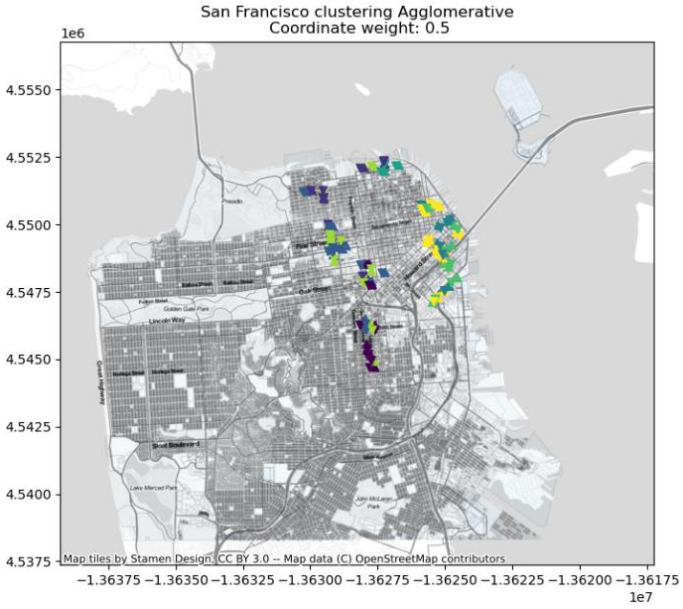


Figura 175

A parità di sample e parametri è stato eseguito il *K-Medoids.* I risultati sono riportati in tabella:

Tabella 13

cluster	***Tasso di occupazione medio (Q)***	***Rank***	*Price con K-Medoids* ***(p)***	***delta price dp***	***delta price dp %***	***Tasso di occupazione aggiustato (Q')***	***Delta Tasso di occupazione (%)***
3	81,42%	1	1,50	0,50	50%	73,28%	-10%
0	82,40%	2	1,83	0,83	83%	68,67%	-17%
2	85,49%	3	2,17	1,17	117%	65,54%	-23%
1	90,79%	4	2,50	1,50	150%	63,55%	-30%

I valori di *"price con K-Medoids (p)"* sono stati ottenuti applicando l'Equazione 26.

I valori del "tasso di occupazione aggiustato (Q')" sono stati ottenuti applicando la seguente equazione:

Equazione 28: Tasso di occupazione "aggiustato" in base alla correzione del prezzo

$$\boldsymbol{Q' = \left(1 + \varepsilon * \frac{dp}{p}\right) Q}$$

Si nota che in questo modo il tasso di occupazione "aggiustato" rientra nella fascia target tra il 60% e l'80%.

Al fine di visualizzare i risultati ottenuti di seguito si riporta:

a) la Figura 176 mostra il grafico della metrica (ovvero del tasso di occupazione) nel tempo:
si nota che la metrica è una dimensione più o meno rilevante; ad esempio, si nota che il cluster in arancione è caratterizzato da un tasso di occupazione maggiore di circa l'88%.

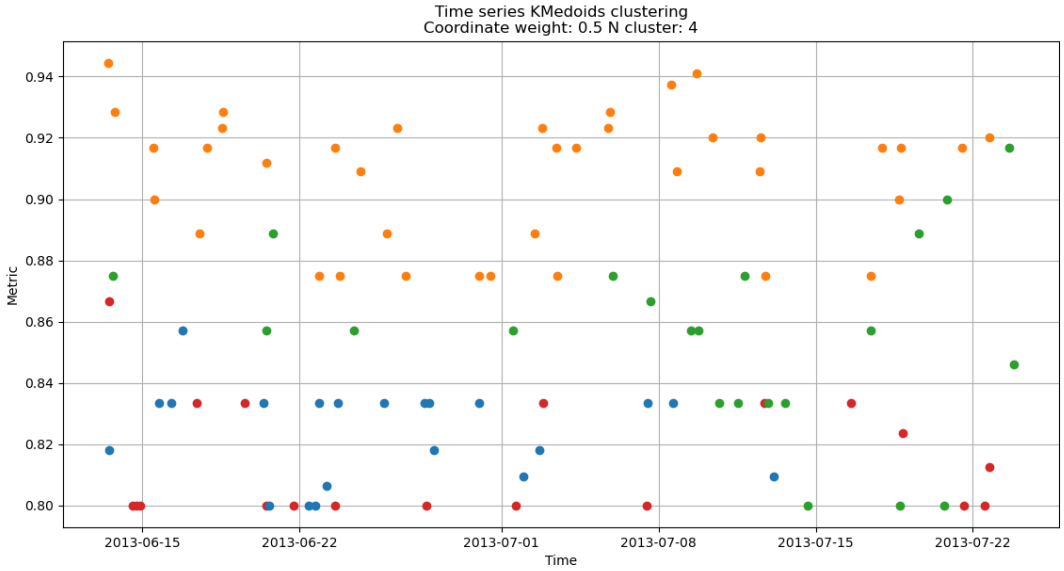


Figura 176

b) la Figura 177e la Figura 178 mostrano i grafici in 3 dimensioni della latitudine, longitudine e metrica/tempo:
si nota che c'è una differenziazione maggiore rispetto alla metrica e una lieve differenziazione in base al tempo.

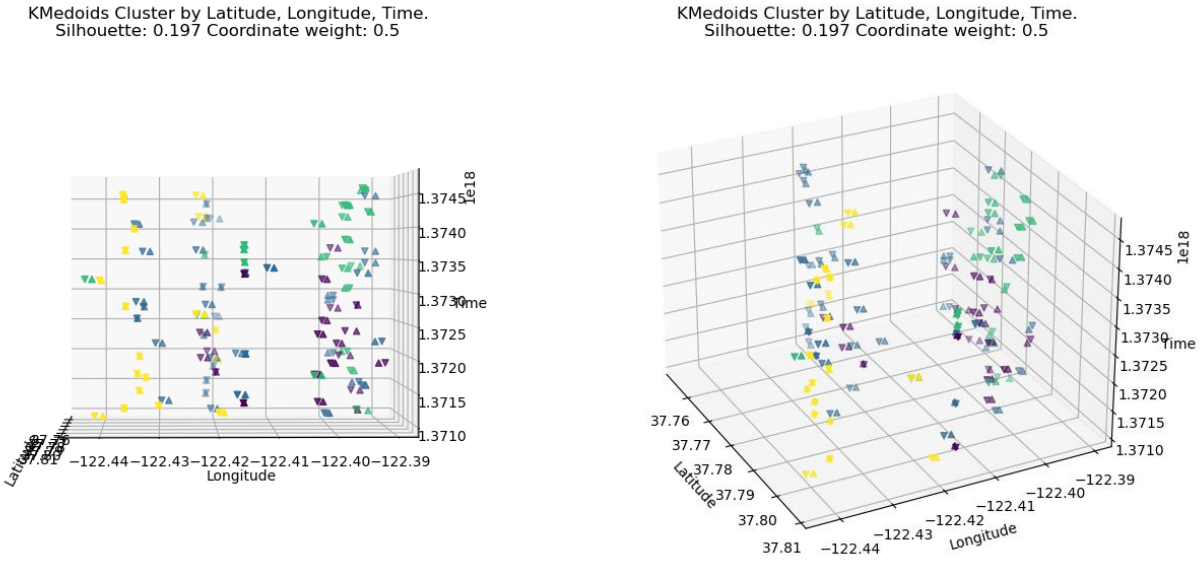


Figura 177

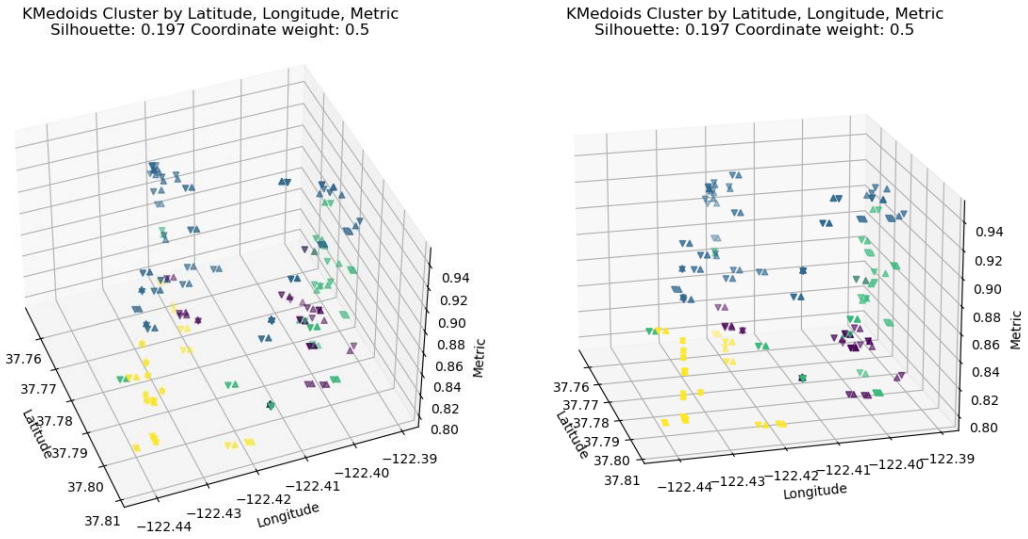


Figura 178

c) la Figura 179 mostra la mappa di San Francisco, da cui è visibile la geolocalizzazione dei vari parcheggi *clusterizzati:* si può notare una leggera discriminazione in base allo spazio al fine dell'identificazione dei cluster.

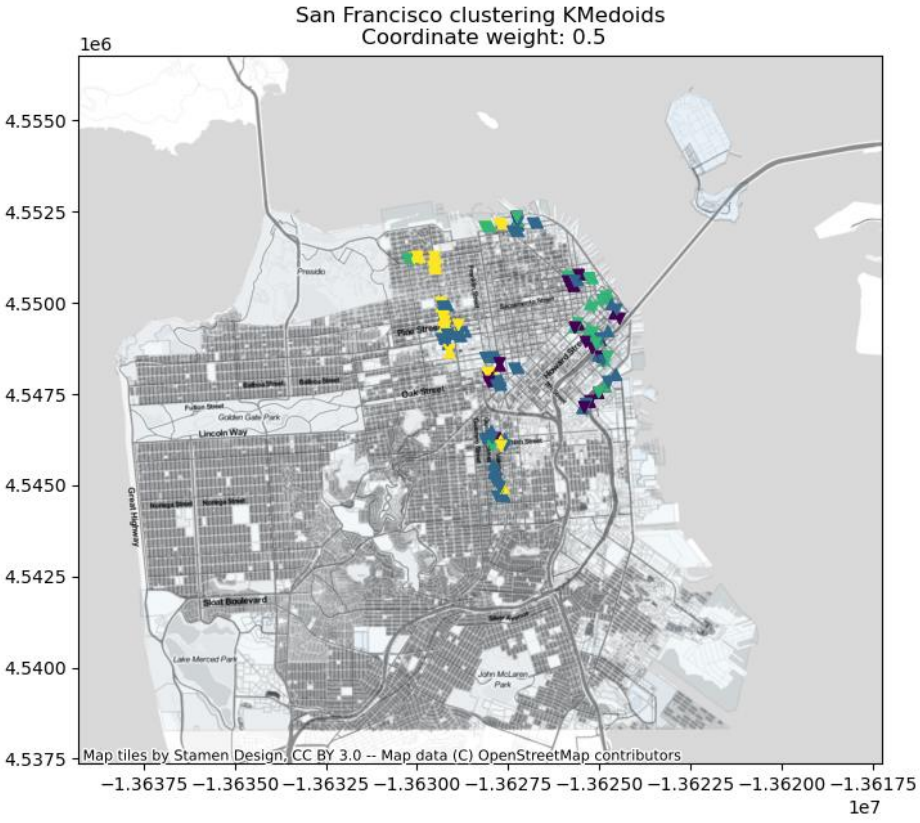


Figura 179

Il grafico riportato nella Figura 180 mostra il prezzo da assegnare ad ogni cluster, separatamente per *K-Medoids* e *Agglomerative.*

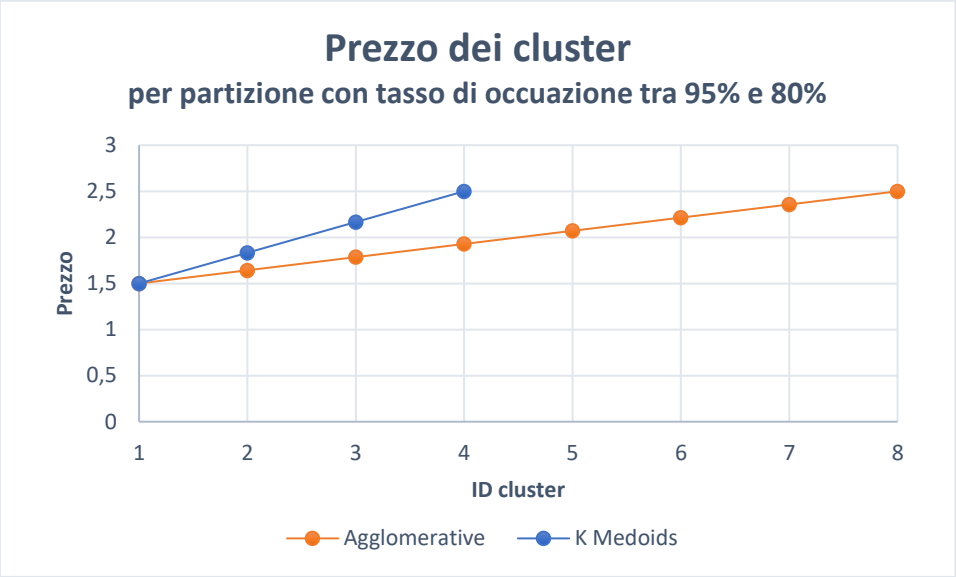


Figura 180

6.3.5. APPROFONDIMENTO: SF-PARK

In questo paragrafo viene contestualizzato il progetto *SF-Park* da un punto di vista economico. In particolare, il *clustering* dei dati spazio-temporali, oggetto della tesi, potrebbe essere utilizzato come algoritmo di *datamining* alla base del modello di determinazione dei prezzi dinamici dei parcheggi.

Il progetto *SF-Park,* introdotto da SFMTA (*San Francisco Municipal Transportation Agency*) nel 2011 a San Francisco [44], si inserisce all'interno del processo di adeguamento dei prezzi in base al tasso di occupazione dei parcheggi, detto anche "*demand-based pricing*" o "*performance-based pricing*".

Il Dipartimento dei trasporti degli Stati Uniti ha investito in tale progetto circa 18 milioni di dollari per testare i prezzi modificati sulla base della domanda; infatti, il progetto *SF-Park* nasce con l'obiettivo di fissare il "giusto" prezzo del parcheggio.

Una politica dei prezzi dinamici può migliorare le prestazioni sia dei parcheggi sia delle strade adiacenti [45]. Infatti, un costo del parcheggio troppo basso crea un eccesso di domanda, che si traduce in elevati costi sociali per tutti tranne per quei pochi fortunati conducenti che trovano un parcheggio a basso costo. Allo stesso modo, un costo del parcheggio troppo elevato genera un eccesso di offerta, a cui seguono problemi come ad esempio, posti auto vuoti, a cui seguono delle perdite di potenziali clienti per i negozi vicini, a cui può seguire il fatto che i dipendenti dei negozi potrebbero perdere il lavoro e i governi le entrate fiscali. Per evitare i problemi causati da prezzi errati dei parcheggi, alcune città, tra cui San Francisco, Seattle e Washington DC, hanno iniziato a adeguare i prezzi dei parcheggi in base alla località, all'ora del giorno e al tasso di occupazione.

In particolare, relativamente al progetto *SF-Park,* San Francisco ha installato dei sensori che rilevano l'occupazione di ogni parcheggio e dei parchimetri che addebitano un costo del parcheggio calcolato considerando i prezzi variabili in base all'ora del giorno. Utilizzando questa nuova tecnologia, la città adegua circa una volta ogni sei settimane i prezzi dei parcheggi sulla base dei tassi di occupazione Questo processo di *try-and-error* mira a creare una struttura di prezzi che variano in base al tempo e al tasso di occupazione per produrre un tasso di occupazione medio compreso tra il 60% e l'80% per ogni parcheggio.

La Figura 181 considera i prezzi del parcheggio nei giorni feriali presso la famosa destinazione turistica, Fisherman's Wharf, a maggio 2012 [45].

Figura 181

Ogni blocco ha prezzi diversi durante tre periodi della giornata (prima di mezzogiorno, da mezzogiorno alle 15:00 e dopo le 15:00). Prima delle prime modifiche nell'agosto 2011, il prezzo era di $ 3 l'ora, senza variazioni durante la giornata. A maggio 2012, i prezzi sono stati diminuiti per il periodo prima di mezzogiorno, mentre la maggior parte dei prezzi è stata aumentata tra mezzogiorno e le 15:00. La maggior parte dei prezzi dopo le 15:00 erano inferiori rispetto a metà giornata, ma superiori rispetto al mattino. Il prezzo del parcheggio sul blocco all'estrema sinistra delle mappe nella Figura 181, ad esempio, era $ 1,50 un'ora prima di mezzogiorno, $ 3 un'ora da mezzogiorno alle 15:00 e $ 1,75 un'ora dopo le 15:00. Ad esempio, un autista che arriva alle 11 e parcheggia per due ore paga $ 1,50 per la prima ora e $ 3 per la seconda ora.

Il sistema di adeguamento dei prezzi SF-PARK si basa esclusivamente sull'occupazione osservata nel periodo precedente, utilizzando un semplice processo di *try-and-error* per regolare i prezzi in risposta ai tassi di occupazione. La Figura 182 di seguito illustra come l'aumento del prezzo nel blocco A (affollato) e la riduzione dei prezzi nel blocco B (sottooccupato) può spostare un'auto per migliorare le prestazioni di entrambi i blocchi. Quindi tale sistema si basa sull'adeguamento dei prezzi dei posti auto per modificare il comportamento dei parcheggiatori. L'obiettivo di tale progetto è portare tutti i parcheggi ad un tasso di occupazione compreso tra il 60% e l'80%.

Aumentare il prezzo di un parcheggio sovraffolato offre diversi importanti vantaggi come mostrato nella Figura 182;

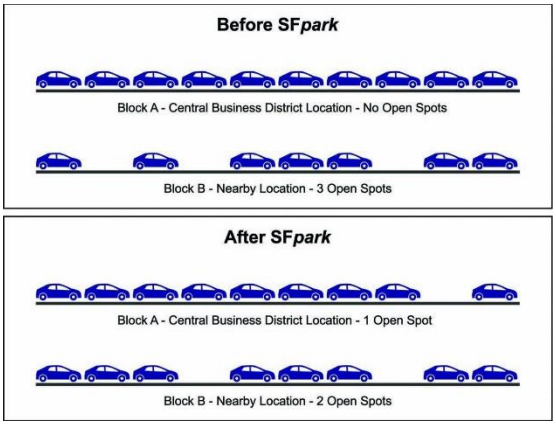


Figura 182

1. Innanzitutto, un parcheggiatore con una *willingness to pay* più bassa del prezzo imposto non sarà disposto a parcheggiare in quel box; di conseguenza lascerà libero un posto per quei conducenti con una più alta *willingness to pay* e allo stesso tempo evitando che questi ultimi impieghino molto tempo nella ricerca di tale parcheggio.
2. In secondo luogo, riducendo i tempi per la ricerca di un parcheggio, si avrà un beneficio sul traffico meno congestionato.
3. In terzo luogo, se i prezzi di un parcheggio sono più alti, tendenzialmente i conducenti parcheggeranno per un tempo più breve, aumenteranno il tasso di *turnover* e quindi consentiranno a più auto di utilizzare i posti auto.
4. In quarto luogo, per risparmiare i costi di parcheggio più elevati, tendenzialmente sempre più persone utilizzeranno il *carpool* o trasporti pubblici, riducendo il numero di auto in circolazione e di conseguenza l'inquinamento.

L'utilizzo dei prezzi per modificare il comportamento di pochi parcheggiatori può quindi migliorare i trasporti, l'economia e l'ambiente.

Inoltre, in un'ottica più generale, SF-PARK aiuta a "*depoliticizzare*", ovvero a fissare delle regole trasparenti basate sui dati per la determinazione del prezzo del parcheggio, piuttosto che determinare il prezzo del parcheggio sulla base di decisioni poco trasparenti della politica in forza. Infatti, viene applicato un prezzo basso laddove c'è un parcheggio poco affollato, in modo da aumentarne l'occupazione. Al contrario, applica un prezzo alto, laddove c'è un parcheggio molto affollato, in modo da spingere i conducenti a scegliere posti

meno costosi, quindi poco affollati. In questo modo, i prezzi vengono fissati sulla base della domanda e non vengono stabiliti per generare maggiori entrate nelle casse pubbliche.

La tabella di seguito definisce i criteri di determinazione dei prezzi sulla base del tasso di occupazione che sono stati utilizzati durante parte del progetto pilota SF-Park nel 2011:

Tabella 14

Occupancy rate	Price change
Below 30%	−50 cents per hour
30%–60%	−25 cents per hour
60%–80%	no change
Above 80%	+25 cents per hour

Se i prezzi più alti incoraggiano alcuni conducenti ad allontanarsi dai parcheggi più affollati, è più probabile che tre tipi di conducenti possono essere definiti *first mover,* ovvero saranno i primi a parcheggino più lontano:

1. parcheggiatori a lungo termine
2. conducenti solitari
3. coloro che attribuiscono un valore basso al risparmio di tempo di viaggio.

1. Parcheggiatori a lungo termine

L'economista canadese Vickrey [46], premio Nobel per l'economia nel 1996, ha osservato che i parcheggiatori a lungo termine hanno più da guadagnare se scelgono parcheggi più economici. Un conducente che parcheggia per quattro ore in uno spazio distante che costa $ 1 l'ora in meno risparmierà $ 4, mentre un conducente che parcheggia per 15 minuti risparmierebbe solo 25 centesimi. Sembra quindi probabile che i conducenti che parcheggiano per più tempo siano tra i primi a spostarsi negli spazi più economici ma meno convenienti. Se un conducente intenzionato a parcheggiare per quattro ore si sposta in uno spazio lontano, più conducenti che parcheggiano per un tempo più breve possono utilizzare lo spazio più conveniente e risparmiare tempo a piedi.

2. Conducenti solitari

Un autista solitario che parcheggia per un'ora e si sposta in uno spazio più lontano che costa $ 1 l'ora in meno risparmierà $ 1, mentre un *carpool* per quattro persone risparmierà solo 25 centesimi a persona. Pertanto, sembra probabile che i conducenti solitari saranno tra i primi a spostarsi negli spazi più economici ma meno costosi, mentre i *carpooler* parcheggeranno negli spazi più convenienti ma più costosi.

3. Coloro che attribuiscono un valore basso al risparmio di tempo di viaggio

I conducenti che amano camminare o che attribuiscono un valore basso al risparmio di tempo speso a camminare si sposteranno anche verso gli spazi più economici, anche se più lontani dal luogo desiderato. Ad esempio, i conducenti che arrivano in anticipo e hanno tempo libero parcheggeranno più lontano, mentre i conducenti che arrivano in ritardo parcheggeranno più vicino. Anche i conducenti a basso reddito che attribuiscono un valore inferiore al risparmio di tempo hanno maggiori probabilità di parcheggiare più lontano. Se i prezzi dei parcheggi rimangono gli stessi ovunque, i conducenti a basso reddito non possono risparmiare denaro spostando le loro posizioni di parcheggio e camminando più lontano.

SF-PARK assegna quindi i prezzi dei parcheggi in modo più efficiente di quanto non possano fare i prezzi uniformi. I parcheggiatori di breve durata, i carpooler, coloro che hanno difficoltà a camminare e coloro che danno molta importanza al risparmio di tempo si sposteranno verso i parcheggi più vicini. Al contrario, i parcheggiatori di lunga data, i conducenti solitari, coloro che amano camminare e coloro che attribuiscono un valore basso al risparmio di tempo si sposteranno verso i parcheggi più distanti. SF-PARK consente a tutti i conducenti di sfruttare una nuova opportunità per risparmiare denaro o tempo.

Dopo diversi anni di pianificazione, *San Francisco Municipal Transportation Authority* (**SFMTA**) ha lanciato il progetto *SF-PARK* nell'aprile 2011. Il programma pilota copre sette zone che contengono 7.000 parcheggi lungo la carreggiata e 14 garage pubblici. SF-park ha apportato le prime modifiche ai prezzi a livello di blocco nell'agosto 2011.

La maggior parte dei sensori funziona tutti i giorni dalle 9:00 alle 18:00, con prezzi che variano in base all'ora del giorno e tra i giorni feriali e i fine settimana. SFMTA ha stabilito il tasso di occupazione target desiderato per i blocchi SFpark tra il 60% e l'80%.

1. Se l'occupazione media per un dato periodo rientra in questo intervallo, il prezzo non cambierà nel periodo successivo.
2. In caso contrario, i prezzi cambiano in base ai tassi di occupazione nel periodo precedente secondo il programma nella Tabella 14 riportata precedentemente. Il prezzo minimo per ora su ogni blocco è di 25 centesimi e il massimo è di $ 6.

Se SFpark funziona come previsto, i prezzi sposteranno i tassi di occupazione verso la fascia target.

Di seguito viene analizzato come i prezzi hanno influenzato le occupazioni durante il primo anno del programma, in cui ci sono stati 5.294 cambiamenti di prezzo.

SFpark ha effettuato 6 aggiustamenti di prezzo durante il primo anno, riportati nella tabella di seguito.

Tabella 15

	August 2011	**August 2012**				
Neighborhood	**All day($)**	**Before noon($)**	**Noon to 3 p.m.($)**	**After 3 p.m.($)**	**Average($)**	**Change(%)**
Downtown	3.50	3.92	4.51	4.40	4.28	22
South Embarcadero	3.50	2.47	3.16	2.83	2.82	−19
Civic Center	3.00	1.87	2.87	2.56	2.43	−19
Fisherman's Wharf	3.00	1.51	2.82	2.59	2.31	−23
Fillmore	2.00	1.88	2.44	2.36	2.23	11
Marina	2.00	1.91	2.72	2.68	2.44	22
Mission	2.00	1.73	2.50	2.63	2.29	14

In generale, nel cambiamento dei prezzi sulla base del tasso di occupazione, si può intercettare una dipendenza spaziale.

In tutte le fasce temporali, i prezzi sono aumentati per l'area del centro (*Downtown*), mentre sono diminuiti nel Civic Center, Fisherman's Wharf e South Embarcadero.

Nelle altre aree, i prezzi sono aumentati in alcune fasce temporali e sono diminuiti in altre. In media, i prezzi sono diminuiti al mattino e sono aumentati a mezzogiorno e al pomeriggio.

Prima di ogni variazione di prezzo, SFpark pubblica i dati sull'occupazione e sui prezzi per tutti i parcheggi nelle zone pilota. L'elasticità della domanda rispetto al prezzo misura il modo in cui queste variazioni di prezzo hanno influenzato i tassi di occupazione.

Figura 183

La Figura 183 mostra i cambiamenti di prezzo e di occupazione in due località: il blocco 600 di Beach Street a Fisherman's Wharf e il blocco 200 di Drumm Street nel centro. Su Beach Street, il prezzo iniziale nell'agosto 2011 era di $ 3 l'ora e l'occupazione iniziale solo del 27%. A febbraio 2012, il prezzo era sceso a 1,75 $/ora, mentre l'occupazione era aumentata al 56%. Applicando l' Equazione 30, l'elasticità media della domanda rispetto al prezzo risulta essere pari a **-1,33**. La domanda è dunque elastica: una maggiore occupazione ha più che compensato il prezzo più basso. Su Drumm Street, il prezzo iniziale era di $ 3,50 l'ora e l'occupazione iniziale era del 98%. Dopo che il prezzo è aumentato a $ 4,50 l'ora, l'occupazione è scesa all'86%. Applicando l' Equazione 30, l'elasticità media della domanda rispetto al prezzo risulta essere pari a **-0,5.** La domanda era anelastica: l'occupazione è diminuita meno dell'aumento del prezzo.

Le variazioni di prezzo hanno modificato il tasso di occupazione verso l'obiettivo desiderato. Tuttavia, dato il range di occupazione target compreso tra il 60% e l'80%, il prezzo è rimasto troppo basso su Beach (dove l'occupazione era solo del 56%) e troppo alta su Drumm (dove l'occupazione era dell'86%). Quindi è stata necessaria una successiva modifica.

6.3.6. Elasticità della domanda rispetto al prezzo

In microeconomia, con il concetto di elasticità della domanda rispetto al prezzo si indica quella misura che mette in relazione la variazione percentuale della quantità domandata con la variazione percentuale del prezzo. Si definisce come la "reattività" della quantità domandata di un bene in seguito ad una variazione del prezzo di tale bene.

L' elasticità della domanda rispetto al prezzo venne elaborata dall'economista Alfred Marshall (1890) e indica la variazione percentuale attesa della domanda di un dato prodotto/servizio (espressa in termini di quantità venduta Q) rispetto ad una variazione percentuale del prezzo dello stesso prodotto:

Equazione 29 Elasticità della domanda rispetto al prezzo

$$\varepsilon(p) = \frac{\partial Q/Q}{\partial p/p} = \frac{\partial Q}{\partial p}\frac{p}{Q}$$

In particolare, si può rilevare che:

- $\varepsilon(p) = 0$ se la domanda è perfettamente rigida
- $0 < |\varepsilon(p)| < 1$ se la domanda è inelastica
- $1 < |\varepsilon(p)| < \infty$ se la domanda è elastica
- $|\varepsilon(p)| = \infty$ se la domanda è perfettamente elastica

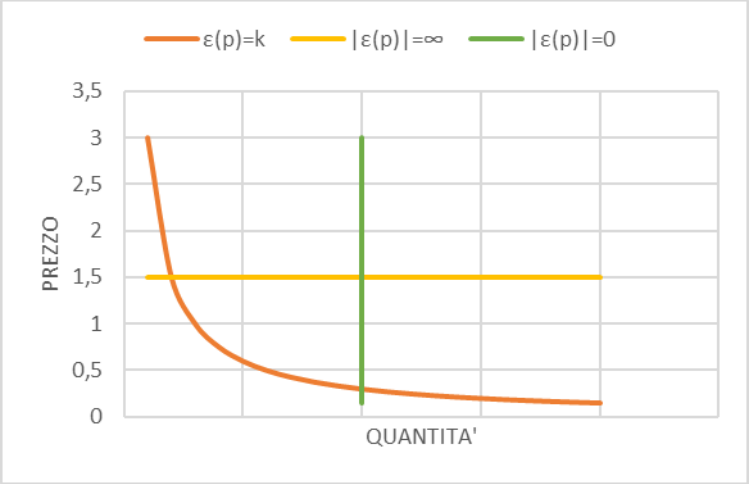
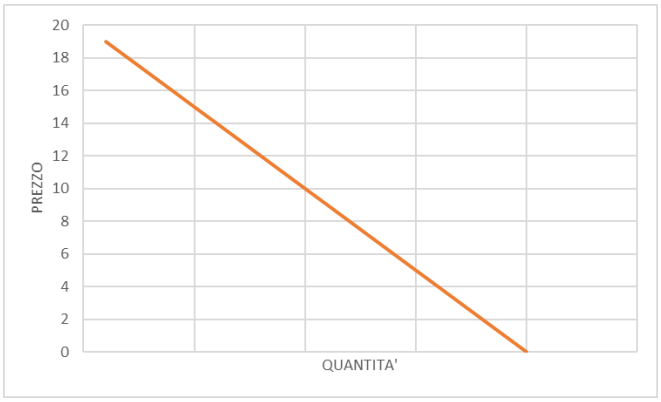


Figura 184 Curve di domanda– casi limite

La Figura 184 descrive alcuni casi limite della curva di domanda; in particolare, la curva lineare verticale (ovvero parallela all'asse del prezzo) descrive una domanda perfettamente rigida, in quanto variazioni del prezzo non comportano nessuna variazione della quantità $\varepsilon(p) = 0$. La curva lineare orizzontale (parallela all'asse delle quantità) implica invece una domanda perfettamente elastica, in quanto piccole variazioni di prezzo (infinitesime variazioni) comportano una variazione infinita della quantità $\varepsilon(p) = \infty$. La curva iperbolica invece definisce una domanda ad elasticità costante in ogni suo punto $|\varepsilon(p)| = k$.



La Figura 185 mostra la curva di domanda lineare. Si nota che l'elasticità della domanda rispetto al prezzo varia nei diversi punti della retta in quanto resta costante il rapporto $\frac{\partial Q}{\partial p}$ mentre cambia il rapporto $\frac{p}{Q}$.

Figura 185 Curva di domanda lineare

Per variazioni finite di prezzo e quantità non si fa riferimento all'elasticità puntuale ($\varepsilon(p) = \frac{\partial Q/Q}{\partial p/p}$), ma all'*elasticità media*, così definita [47]:

$$\varepsilon_M(p) = \frac{\Delta Q}{\Delta p}\frac{p_0 + p_1}{Q_0 + Q_1}$$

Equazione 30 Elasticità media della domanda

In genere, la domanda di un bene è correlata negativamente al prezzo, nel senso che più questo cresce, minore sarà la quantità richiesta dai consumatori di questo bene. Al contrario, quando il prezzo si abbassa, il consumatore tende ad aumentare la quantità richiesta dello stesso bene. In realtà, il concetto di elasticità della domanda rispetto al prezzo da solo non basta per capire la ragione per cui la richiesta sul mercato può variare anche repentinamente. Infatti, la domanda di un bene dipende anche dai prezzi dei possibili beni sostituti, oltre che di quelli dei beni complementari. Nel caso specifico della domanda di posti auto, almeno in prima analisi si può trascurare l'influenza di beni sostituti o complementari validi. Tali beni, infatti, potrebbero essere collegati in senso più generico alla sfera del trasporto; dunque, un bene complementare potrebbe essere la benzina, necessaria per l'utilizzo dell'auto da cui nasce la necessità di trovare un parcheggio; un bene sostituito potrebbe essere un generico trasporto pubblico, come il pullman.

6.4. BIKESHARING DATASET

6.4.1. Descrizione dataset

Il dataset reale utilizzato nell'analisi di seguito riportata è relativo al *bike sharing* di alcune città della California (tra cui San Francisco, Los Angeles e Santa Clara).

L'analisi di tale dataset può essere inserita all'interno della *data analysis* relativa alla *sharing economy.* Dal *bikesharing* è possibile ottenere dati di tipo *spazio-temporali* grazie all'utilizzo di tecnologie come dispositivi dotati di GPS o app su smartphone geolocalizzabili, che tracciano lo spostamento (in termini di latitudine e longitudine) nel tempo, consentendo di studiare le abitudini di viaggio e la mobilità degli utilizzatori. Inoltre, dall'analisi di tali dati è possibile estrarre informazioni rilevanti per la pianificazione e la gestione dei trasporti. Tuttavia, la dimensionalità dei dataset relativi al *bike sharing* è molto elevata e da qui nasce la difficoltà di visualizzare le diverse traiettorie, spesso sovrapposte, al fine di comunicare i risultati ottenuti o estrarre nuovi *pattern*. In particolare, il *dataset* considerato contiene 519.700 records, ognuno dei quali corrisponde ad un percorso eseguito da un rider, tra il 28/06/2017 ore 09:47:36 e il 31/12/2017 ore 23:59:01 (l'arco temporale considerato è di circa 6 mesi). Di seguito nella Figura 186 si riporta il grafico della durata del percorso, espressa in secondi, in funzione del tempo.

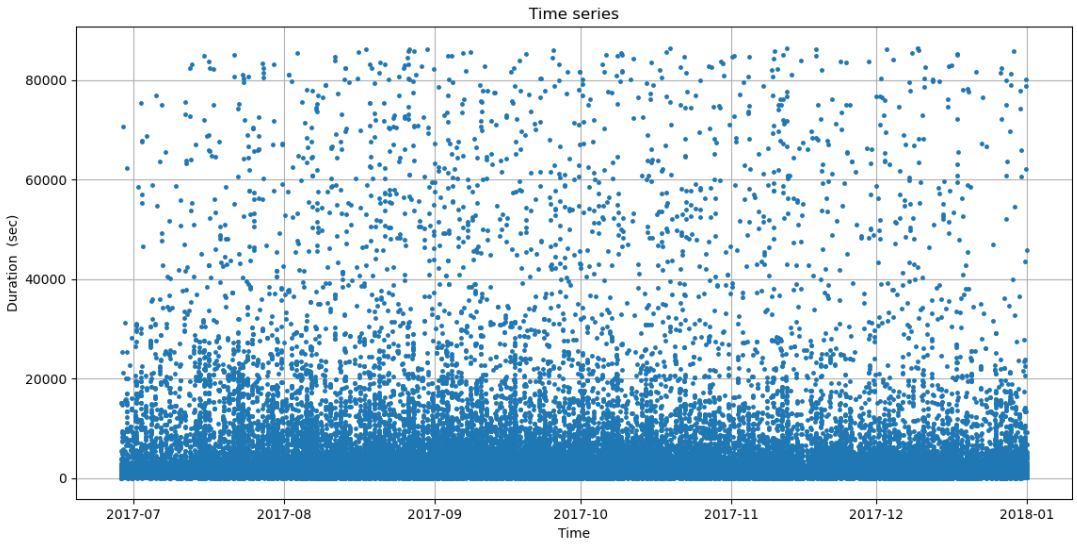


Figura 186

Il dataset *bikesharing* contiene i seguenti attributi:

- *duration_sec:* durata in secondi del percorso
- *start_time:* data e ora dell'inizio dell'affitto della biciletta
- *end_time:* data e ora della fine dell'affitto della biciletta
- *start_station_name:* stazione in cui è avvenuto l'inizio dell'affitto della biciletta
- *start_station_latitude:* latitudine della stazione in cui è avvenuto l'inizio dell'affitto della biciletta
- *start_station_longitude:* longitudine della stazione in cui è avvenuto l'inizio dell'affitto della biciletta
- *end_station_name:* stazione in cui è avvenuta la fine dell'affitto della biciletta
- *end_station_latitude:* latitudine della stazione in cui è avvenuta la fine dell'affitto della biciletta
- *end_station_longitude:* longitudine della stazione in cui è avvenuta la fine dell'affitto della biciletta
- *bike_id*: id della bicicletta affittata
- *user_type:* tipologia di utente
- *member_birth_year*: data di nascita dell'utente

- *member_gender*: sesso dell'utente

Utilizzando lo stesso modello del *SF-PARK dataset,* gli attributi considerati sono:

1. *start_time,* che corrisponde alla dimensione temporale
2. *start_station_latitude – start_station_longitude,* che corrispondono alle coordinate geografiche di partenza
3. *end_station_latitude – end_station_longitude,* che corrispondono alle coordinate geografiche di arrivo
4. *duration_sec,* che corrisponde alla metrica.

Prima di eseguire il *clustering* utilizzando sia l'algoritmo *Agglomerative* che *K-Medoids,* è stata analizzata la distribuzione dei dati. La tabella di seguito riassume quanto rilevato:

Tabella 16

ID	Durata dei percorsi (metrica)	#records	% TOT
1	4 ore=<x< 24 ore	3940	1%
2	2 ore=<x<4 ore	4457	1%
3	1 ora=<x< 2 ore	7789	1%
4	45 min =<x< 1 ora	4558	1%
5	30 min =<x<45 min	15184	3%
6	15 min =<x< 30 min	103930	20%
7	**5 min =<x<15 min**	**302426**	**58%**
8	4 min=<x< 5 min	32290	6%
9	3 min =<x<4 min	25478	5%
10	2 min=<x<3 min	14705	3%
11	60 s =<x<2 min	4943	1%
	Totale	519.700	

La durata dei percorsi con bike sharing, come mostra la Figura 187, sembra seguire una distribuzione normale con valore medio di un viaggio compreso tra i 5 minuti e i 15 minuti.

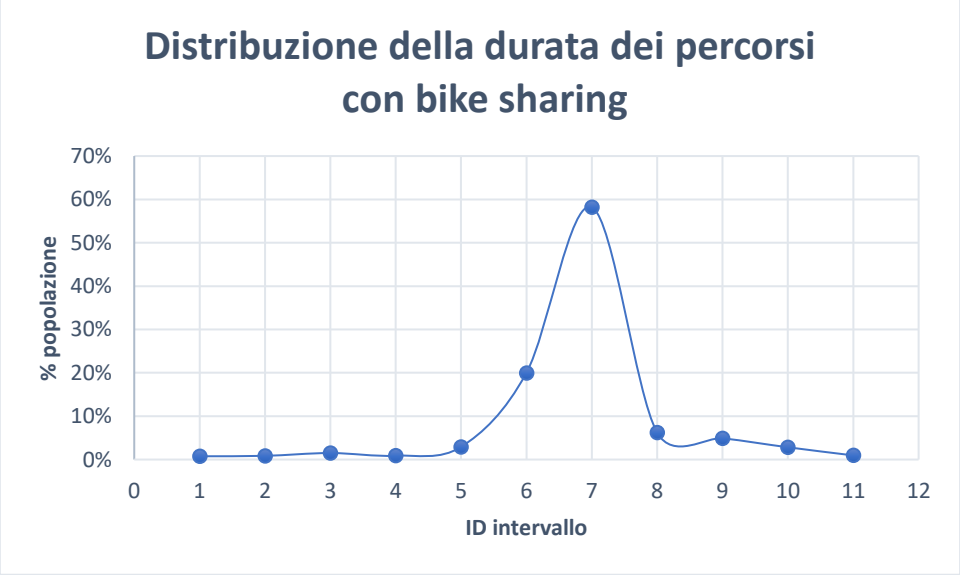


Figura 187

Analogamente a quanto fatto per l'analisi del *datataset SF-Park,* descritta nel paragrafo precedente (6.3), al fine di ridurre la dimensionalità del dataset è stata eseguita prima una partizione secondo gli intervalli

dell'attributo "*durata*" (ovvero la metrica) riportati nella suddetta tabella; successivamente è stato effettuato un campione il quale è stato clusterizzato. Di seguito si riportano i risultati relativi l campione effettuato sulle partizioni:

1) ID 1 caratterizzata da una durata del viaggio compresa tra le 4 ore e le 24 ore
2) ID 7 caratterizzata da una durata del viaggio compresa tra i 5 min e i 15 min

6.4.2. Partizione con durata compresa tra 4 ore e 24 ore

6.4.2.1. Agglomerative clustering

Sono stati selezionati tutti i percorsi con durata compresa tra le 4 e le 24 ore. Risultano essere presenti 3940 tragitti appartenenti a tale categoria, corrispondenti circa all'1% della popolazione. Sulla base di tale popolazione è stato estratto un campione casuale di 100 items.

Tale campione è stato clusterizzato con l'algoritmo l'*agglomerative,* fissando un peso della dimensione temporale pari a quello della componente spaziale ($\boldsymbol{w_1 = w_2 = 0.50}$).

Il grafico di seguito (Figura 188Figura 182) mostra lo *scatterplot* delle durate (in secondi) di tali percorsi, in funzione del tempo. Sono stati identificati 3 cluster, i quali sembrano discriminati dalla metrica, in particolare:

1) il cluster in arancione è costituito da viaggi con durata intorno a 80.000 secondi (circa 24 ore)
2) il cluster in blu è caratterizzato da viaggi con durata compresa tra 40'000 secondi (circa 11 ore) e 80'000 secondi (circa 24 ore)
3) il cluster in verde è costituito da viaggi con durata compresa tra circa 20'000 secondi (circa 5 ore) e 40'000 secondi (circa 11 ore)

La maggior parte dei dati è concentrata nella fascia identificata dal cluster in verde (Figura 188).

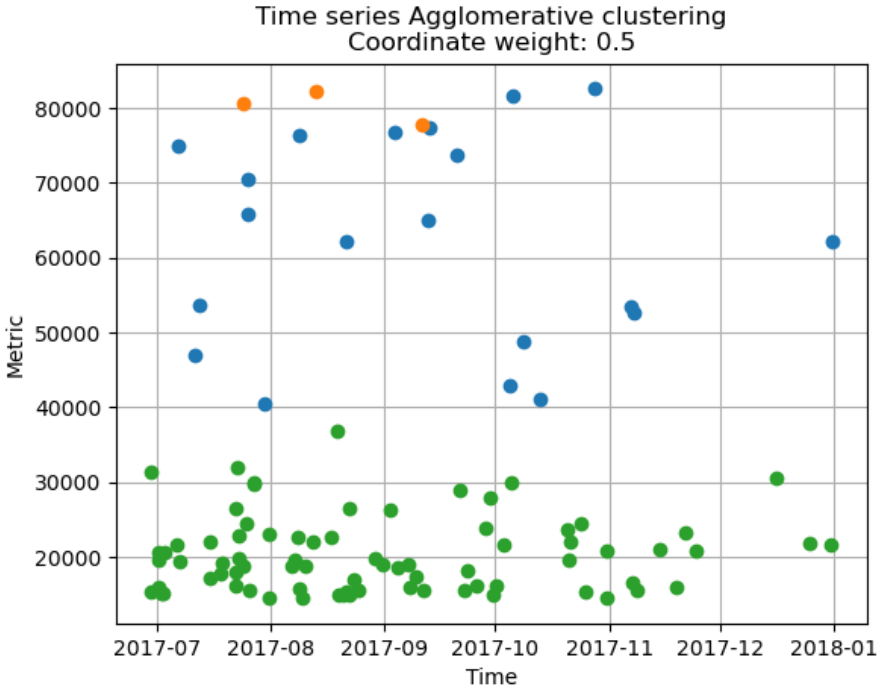


Figura 188

Di seguito nella Figura 189 si riporta il *radar chart* del cluster ottenuto, che mostra le informazioni circa la metrica massima, minima e media dei 3 cluster.

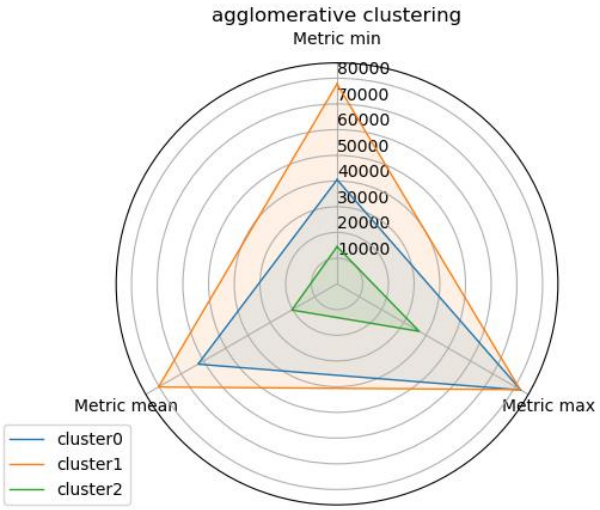


Figura 189

A sinistra si riporta (Figura 190) il grafico 3D della latitudine, longitudine e tempo. I diversi colori identificano la suddivisione in *cluster.* Si nota che il cluster in azzurro identifica dei tragitti localizzati in un'area molto più lontana rispetto ai tragitti appartenenti agli altri 2 cluster (in giallo e in viola).

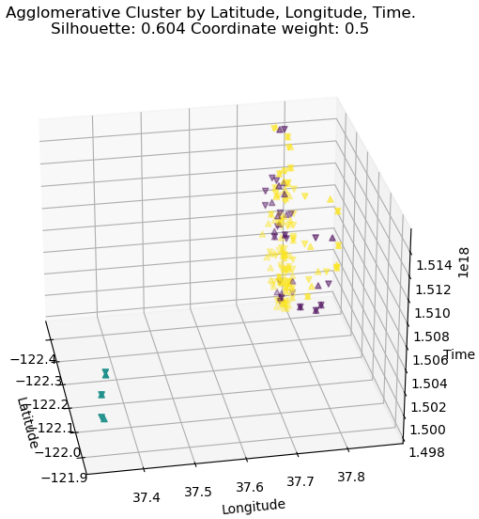


Figura 190

Al contrario, il grafico 3D (Figura 191) della latitudine, longitudine e metrica mostra i 3 diversi cluster ben distinti rispetto alla metrica. Il cluster in azzurro è caratterizzato da dei punti associati a *timestamp* bassi e da valori alti di metrica (durata dei percorsi). Quindi il cluster in azzurro intercetta tragitti più isolati rispetto agli altri con una durata maggiore. Potrebbero rappresentare una anomalia.

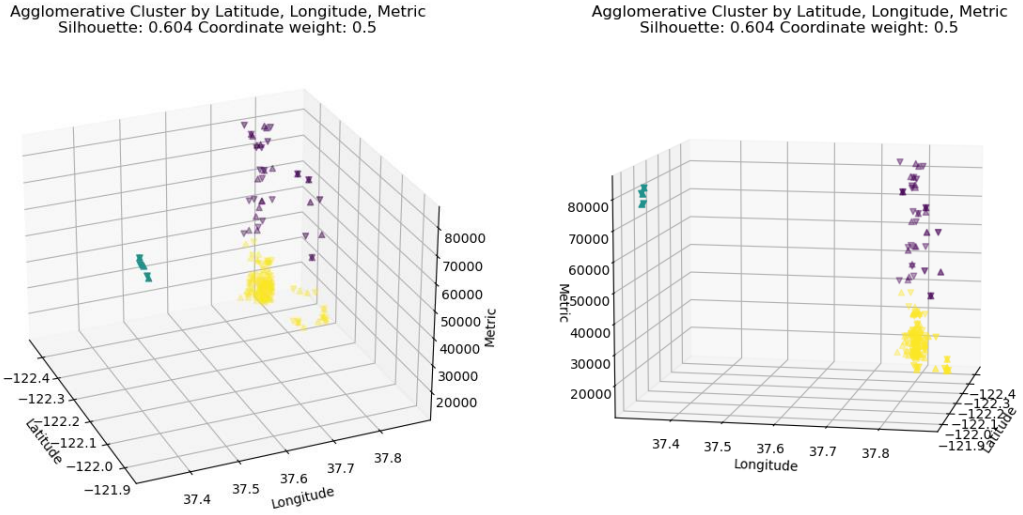


Figura 191

6.4.2.2. K-Medoids

I risultati ottenuti con l'applicazione *dell'agglomerative* clustering sono stati confrontati con quelli ottenuti con il *K-Medoids.* Per questo motivo, lo stesso campione è stato clusterizzato con l'algoritmo *K-Medoids,* fissando un peso della dimensione temporale pari a quello della componente spaziale($\boldsymbol{w_1 = w_2 = 0.50}$).

La Figura 192 mostra lo *scatterplot* delle durate (in secondi) di tali percorsi, in funzione del tempo. Sono stati identificati 4 cluster, i quali sembrano essere discriminati in base alla metrica e in base al tempo. In particolare:

1) il cluster in arancione è costituito da viaggi con durata superiore ai 50.000 secondi e si estende per tutto l'arco temporale considerato
2) il cluster in blu è caratterizzato da viaggi con bassa durata (meno di 20.000 secondi) mentre il periodo di tempo è limitato dal 07/2017 al 09/2017
3) Il cluster in verde è costituito da viaggi con durata compresa tra circa 20'000 secondi (5 ore) e 50'000 secondi mentre il periodo di tempo è limitato dal 07/2017 al 09/2017
4) il cluster in rosso è costituito da viaggi con durata compresa tra circa 20'000 secondi e 40'000 secondi mentre il periodo di tempo è limitato dal 08/2017 al 01/2018.

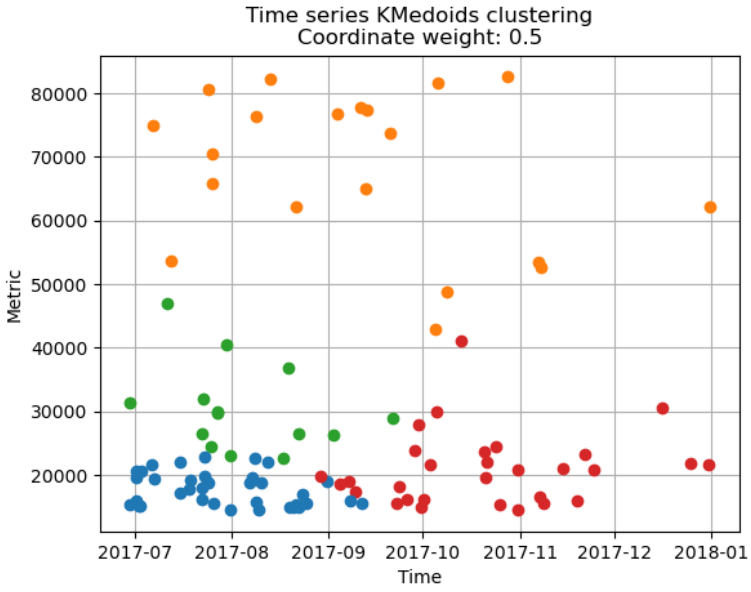


Figura 192

Si riporta nella Figura 193 il grafico 3D della latitudine, longitudine e metrica che mostra i 4 diversi cluster ben distinti rispetto alla metrica. Il cluster in azzurro è caratterizzato da valori elevati di metrica (durata dei percorsi).

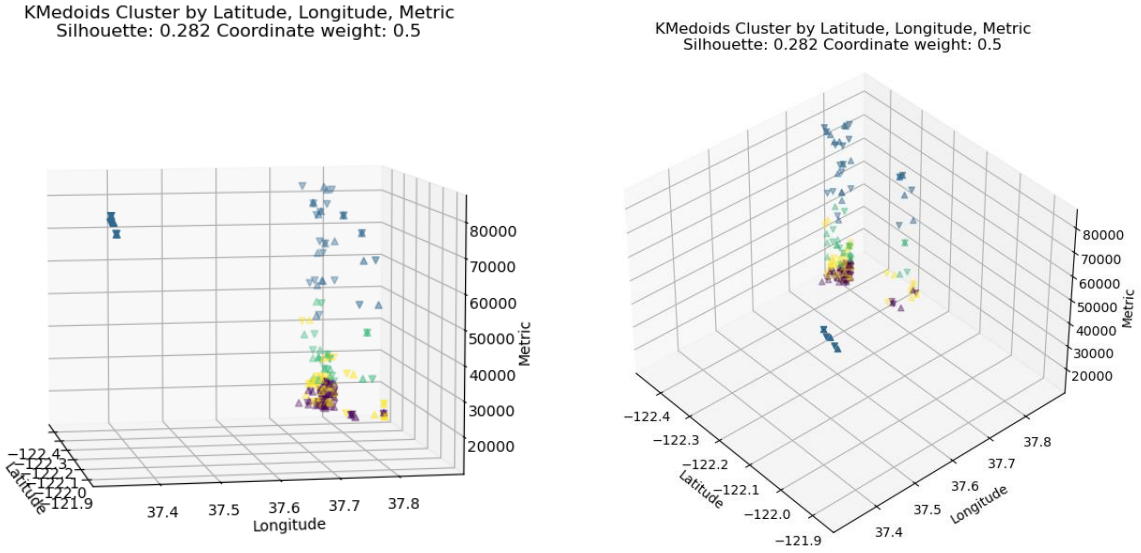


Figura 193

Si riporta nella Figura 194 il grafico 3D della latitudine, longitudine e tempo che mostra i 4 diversi cluster discretamente distinti anche rispetto alla coordinata temporale.

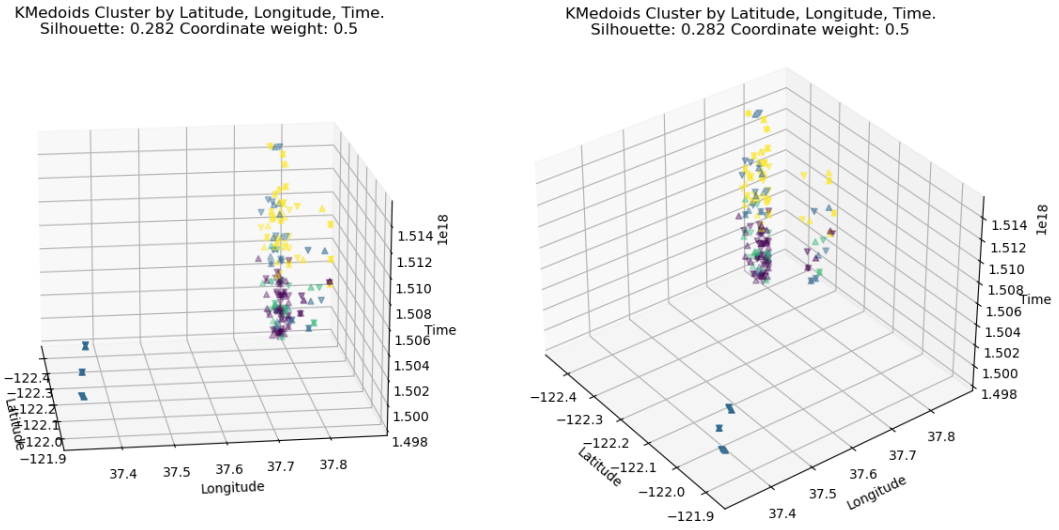


Figura 194

Dal confronto tra i risultati del cluster identificati dai due algoritmi, emerge che:

1) rispetto all'*agglomerative clustering,* il *K-Medoids* sembra discriminare maggiormente rispetto al tempo, a parità di altri parametri, identificando 1 cluster in più
2) al contrario l'*agglomertive clustering* discrimina maggiormente rispetto allo spazio; infatti, la differenza fondamentale tra i due risultati è che il *K-Medoids* non isola in un cluster diverso i tragitti localizzati in una zona molto distante dalla maggior parte degli altri tragitti, ma li include nel cluster che ha valore di metrica simile.

6.4.3. Partizione con durata compresa tra 5 minuti e 15 minuti

6.4.3.1. Agglomerative clustering

Sono stati selezionati tutti i percorsi con durata compresa tra i 5 minuti e i 15 minuti.

Risultano essere presenti 302.426 tragitti che soddisfano tale condizione, corrispondenti circa al 58% della popolazione. Sulla base di tale popolazione è stato estratto un campione casuale di 100 items.

Tale campione è stato clusterizzato con l'algoritmo l'*agglomerative,* fissando un peso della dimensione temporale pari a quello della dimensione spaziale($\boldsymbol{w_1 = w_2 = 0.50}$).

La Figura 195 mostra lo *scatterplot* delle durate (in secondi) di tali percorsi, in funzione del tempo. Sono stati identificati 3 cluster, i quali sembrano essere discriminati in base metrica, in particolare:

1) il cluster in verde è costituito da viaggi con durata compresa tra circa 300 secondi (5 minuti) e 500 secondi (circa 8 minuti)
2) il cluster in blu è costituito da viaggi con durata compresa tra circa 500 secondi (8 minuti) e 900 secondi (15 minuti)

Si nota che il cluster in arancione contiene due punti che corrispondono a tragitti con durata di circa 700 secondi e 450 secondi, i quali sono stati isolati in un terzo cluster distinto poiché associati ad un punto di partenza e arrivo isolati rispetto agli altri tragitti.

La maggior parte dei dati è concentrata nella fascia identificata dal cluster in blu.

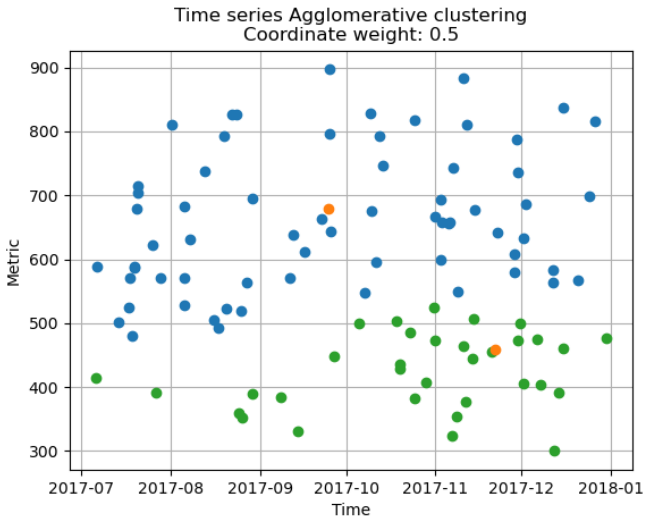


Figura 195

Di seguito (Figura 196) si riporta il *radar chart* del cluster ottenuto, che contiene informazioni circa la metrica massima, minima e media dei 3 cluster.

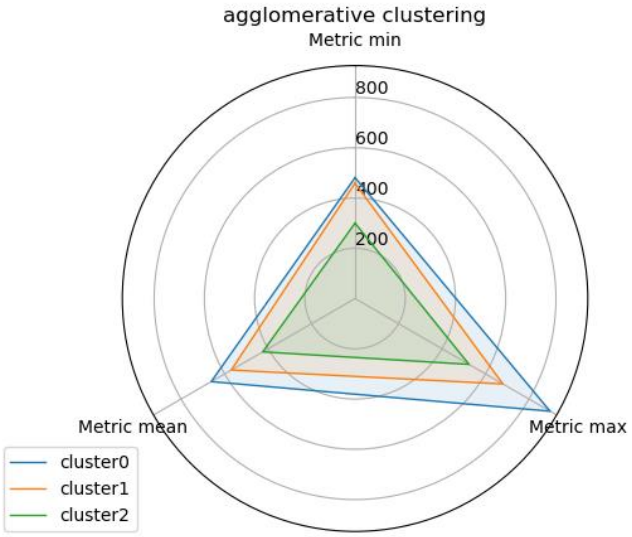


Figura 196

Si riporta il grafico 3D (Figura 197) della latitudine, longitudine e tempo che mostra i 3 diversi cluster. In particolare, si nota che, analogamente alla partizione dei tragitti con durata tra 4 ore e 24 ore (paragrafo 6.4.2), anche in tal caso è stato isolato un cluster caratterizzato da tragitti con punto di partenza e arrivo isolati e molto distanti rispetto agli altri. Al contrario, la componente temporale non sembra essere una discriminante particolare per l'identificazione dei *cluster*.

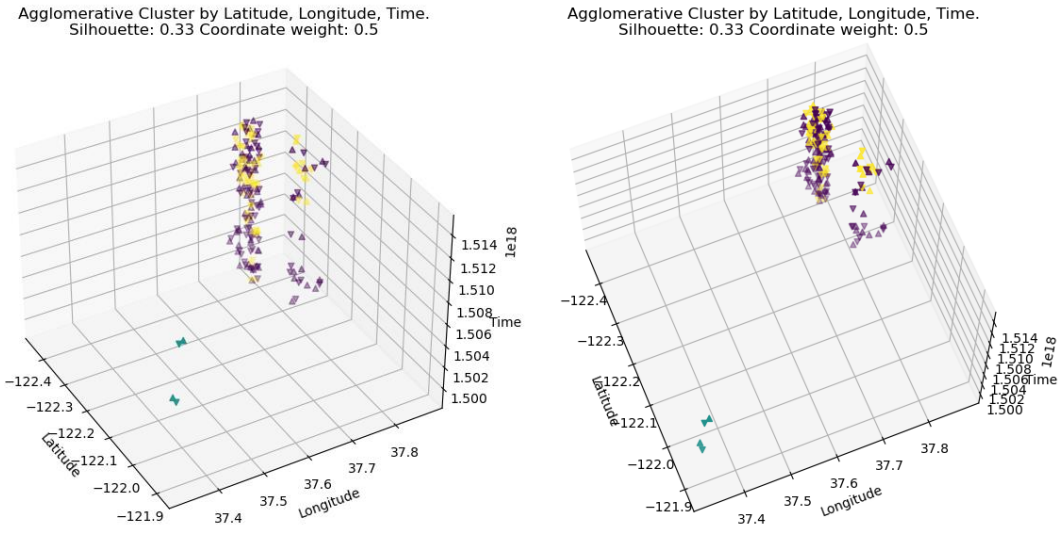


Figura 197

Al contrario, il grafico 3D (Figura 198) della latitudine, longitudine e metrica mostra i 3 diversi cluster ben distinti rispetto alla metrica. Il cluster in azzurro è caratterizzato da dei punti associati a *timestamp* bassi e da valori bassi di metrica (durata dei percorsi). Quindi il cluster in azzurro intercetta tragitti più isolati rispetto agli altri con una durata inferiore.

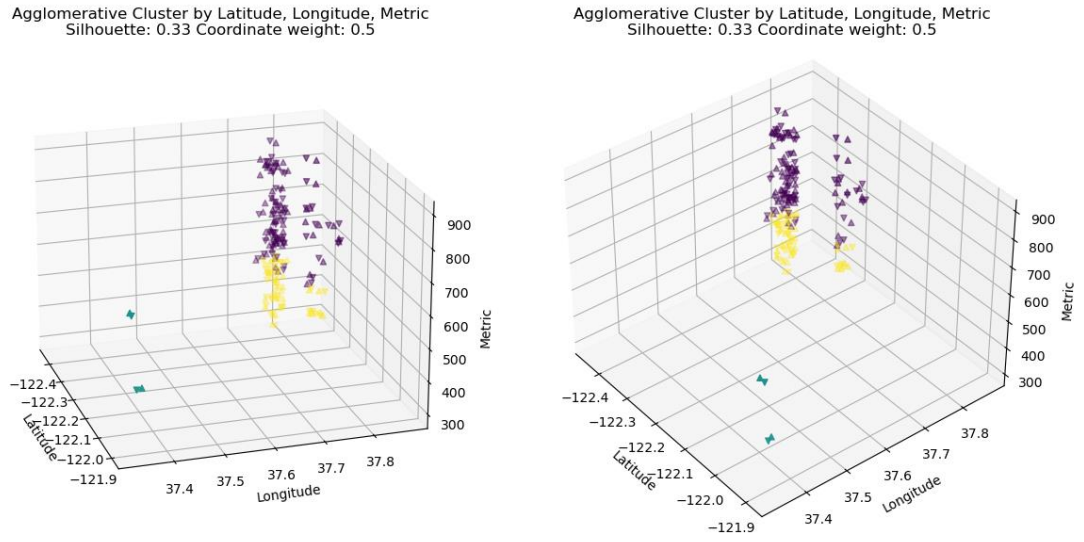


Figura 198

6.4.3.2. K-Medoids

I risultati ottenuti con l'applicazione *dell'agglomerative* clustering sono stati confrontati con quelli ottenuti con il *K-Medoids.* Per questo motivo, lo stesso campione è stato clusterizzato con l'algoritmo *K-Medoids,* fissando un peso della dimensione temporale pari a quello della componente spaziale($\boldsymbol{w_1 = w_2 = 0.50}$).

La Figura 199 mostra lo *scatterplot* delle durate (in secondi) di tali percorsi, in funzione del tempo. Sono stati identificati 3 cluster, i quali sembrano essere discriminati in base alla metrica e in base al tempo. In particolare:

1) il cluster in arancione è costituito da viaggi con durata superiore ai 500 secondi e si estende per un arco temporale è compreso tra 07/2017 e 10/2017
2) il cluster in blu è caratterizzato da viaggi con più bassa durata (meno di 500 secondi) mentre il periodo di tempo è esteso a tutto il periodo considerato
3) Il cluster in verde è costituito da viaggi con durata maggiore, superiore a circa 600 secondi e si estende per un periodo temporale compreso tra il10/2017 e 01/2018

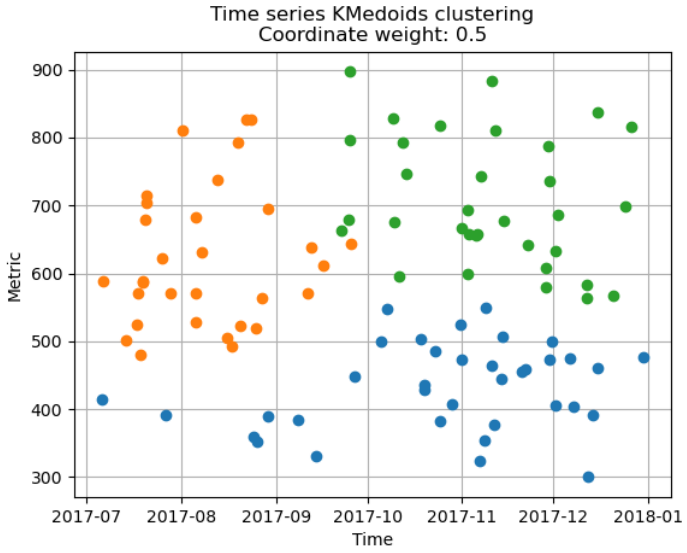


Figura 199

Si riporta il grafico 3D (Figura 200) della latitudine, longitudine e tempo che mostra i 3 diversi cluster. Si nota che in particolare in azzurro è stato intercettato un *cluster* caratterizzato da timestamp bassi (quindi da viaggi effettuati durante il primo periodo di tempo in analisi).

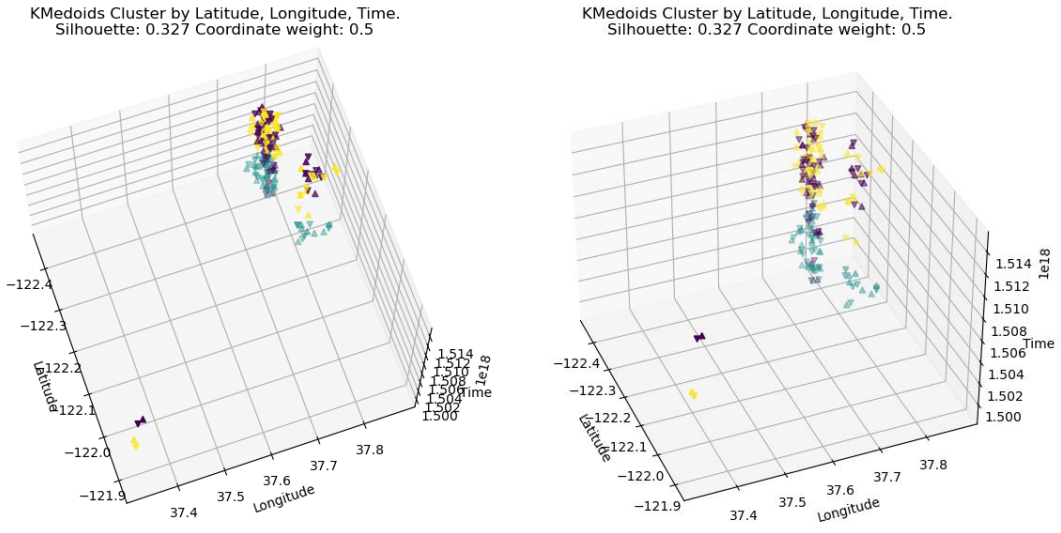


Figura 200

Di seguito si riporta il grafico 3D (Figura 201) della latitudine, longitudine e metrica che mostra i 3 diversi cluster ben distinti rispetto alla metrica. Analogamente a quanto rilevato per il *K-Medoids* applicato sul campione della partizione 4 ore – 24 ore (paragrafo 6.4.2), anche in tal caso il *K-Medoids* non isola in un *cluster* distinto quei tragitti che hanno posizione di partenza e arrivo particolarmente distanti.

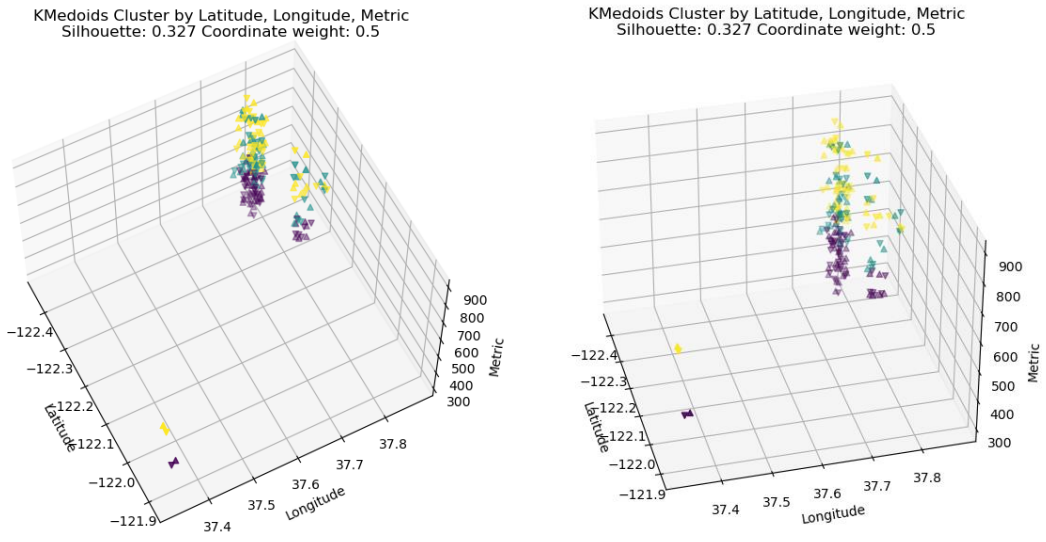


Figura 201

In generale, dal confronto tra i risultati del cluster identificati dai due algoritmi (*Agglomerative* e *K-Medoids*) rispetto alla partizione 5 minuti – 15 minuti, può essere confermato quanto è emerso sulla base del confronto dei due risultati ottenuti dal *clustering* rispetto alla partizione 4 ore – 24 ore.

In particolare, si conferma che:

1) rispetto all'*agglomerative clustering,* il *K-Medoids* sembra discriminare maggiormente rispetto al tempo, a parità di altri parametri, identificando 1 cluster in più

2) al contrario l'*agglomerative clustering* discrimina maggiormente rispetto allo spazio; infatti, la differenza fondamentale tra i due risultati è che il *K-Medoids* non isola in un cluster diverso i tragitti localizzati in una zona molto distante dalla maggior parte degli altri tragitti, ma li include nel cluster che ha valore di metrica simile.

In generale, il dataset sembra essere costituiti da tragitti concentrati in 3 macroaree, due delle quali sono tra loro più vicine, mentre la terza è molto più distante.

7. Conclusioni

L'obiettivo principale del presente lavoro di tesi è la presentazione di diverse metodologie di *clustering* per l'analisi dei dati spazio-temporali. Le proprietà geografiche e temporali sono un aspetto chiave di molti problemi di *data analysis* nel dominio del *business* in generale, della politica e dell'economia, ma anche della scienza.

Le analisi condotte nel presente lavoro di tesi si basano principalmente sull'applicazione di algoritmi di *clustering distance-based.* In particolare, al fine di testare la nuova metrica di distanza introdotta per i dati spazio-temporali, sono stati simulati una serie di dataset con attributi spaziali e temporali; tali dataset sono stati *clusterizzati* con gli algoritmi *K-Medoids, Agglomerative* e *ST-DBSCAN*.

I risultati ottenuti hanno mostrato che:

- il peso delle componenti spaziali e temporali che massimizzano un indice (*Silhouette* o *Dunn* Index) variano a seconda della distribuzione dei dati rispetto allo spazio e al tempo; quindi, non sarebbe del tutto corretto fissare a priori tali parametri, indipendentemente dal dataset. Al contrario, è preferibile fissare tali parametri sulla base di una prima analisi della distribuzione dei dati (in modo tale che, ad esempio, si attribuisca un basso peso alla componente temporale se i dati sono distribuiti in modo più o meno uniforme nel tempo, in quanto ciò potrebbe indicare una scarsa incidenza del tempo).
  Di conseguenza, per il cluster dei datatset reali *SF Park* e *Bikesharing* il peso delle componenti spaziali e temporali non è stato fissato a priori ma è stato considerato come un input modificabile, sulla base della distribuzione;

- massimizzando la *Silhouette* e il *Dunn Index* come criterio per la scelta del numero di cluster si individua la maggior parte delle volte lo stesso numero di cluster. Dunque, il *Dunn Index* e la *Silhouette* possono essere considerati indici "sostituibili".
  Di conseguenza, per il cluster dei datatset reali *SF Park* e *Bikesharing il numero di* cluster è stato determinato in modo da massimizzare il solo indice di *Silhouette,* maggiormente utilizzato in letteratura (non considerando il *Dunn Index*).

Dopo aver valutato l'impatto dei vari parametri della metrica di distanza introdotta mediante clustering dei dataset sintetizzati, sono stati *clusterizzati 4* dataset reali (non simulati), anch'essi caratterizzati da attributi spaziali e temporali; tali dataset sono:

- **GPS ST-EVENTS**: dataset di un subset di dati raccolti da Microsoft Research Asia durante il progetto *Geolife,* che riporta le traiettorie raccolte con il GPS degli smartphone

- **SF-PARK:** dataset relativo al progetto *SF-Park* di San Francisco che riporta la disponibilità di alcuni parcheggi di San Francisco misurata grazie a sensori posti sotto l'asfalto

- **BIKESHARING**: dataset contenete le informazioni relative al servizio di bike sharing di alcune città della California (tra cui San Francisco, Los Angeles, Santa Clara)

- **GEOREFERENCED TIME SERIES**: dataset contenete i valori orari della temperatura atmosferica, misurati nell'arco di 5 anni e relativi a 50 città degli Stati Uniti e del Canada e 6 città di Israele.

Dall'analisi del dataset **GPS ST-EVENTS** si possono inferire considerazioni circa i diversi luoghi visitati dagli utenti nel corso del tempo.

Dall'analisi del dataset **SF-PARK** è possibile individuare un modello di clustering spazio-temporale utilizzabile per la determinazione *near-real* time del prezzo dinamico dei parcheggi. Tale prezzo, infatti, può essere determinato per ogni parcheggio sulla base del cluster di appartenenza identificato, con l'idea di assegnare il prezzo in modo proporzionale al tasso di occupazione rappresentativo del *cluster*. La frequenza del cambiamento della tariffa è variabile per ogni parcheggio e avviene ogni qualvolta, nell'arco della giornata, il parcheggio cambia *cluster* di appartenenza. Modificando il peso da assegnare alle coordinate spaziali e temporali sarebbe possibile dare più o meno importanza ai vari fattori (ad esempio, scegliendo di identificare *cluster* solo sulla base del tasso di occupazione nel tempo, e dunque non considerando la localizzazione geografica).

Dall'analisi del dataset *bikesharing* è possibile studiare le abitudini di viaggio, la mobilità degli utilizzatori ed estrarre informazioni rilevanti per la pianificazione e la gestione dei trasporti. In particolare, dall'analisi del dataset ***BIKESHARING***, è emerso che:

- la durata dei percorsi con bikesharing sembra seguire una distribuzione normale con valore medio di un viaggio compreso tra i 5 minuti e i 15 minuti;

- in generale, il dataset bikesharing sembra essere costituito da tragitti concentrati in 3 macroaree, due delle quali sono tra loro più vicine (San Francisco e Santa Clara), mentre la terza è molto più distante (Los Angeles)

- rispetto all'*agglomerative clustering,* il *K-Medoids* sembra discriminare maggiormente rispetto al tempo, a parità di altri parametri, identificando 1 cluster in più;

- al contrario l'*agglomerative clustering* discrimina maggiormente rispetto allo spazio; infatti, la differenza fondamentale tra i due risultati è che il *K-Medoids* non isola in un cluster diverso i tragitti localizzati in una zona molto distante dalla maggior parte degli altri tragitti, ma li include nel cluster che ha valore di metrica simile (ovvero la durata del tragitto);

Il clustering del dataset **GEOREFERENCED TIME SERIES** che riporta i valori orari della temperatura atmosferica, misurati nell'arco di 5 anni e relativi a 50 città degli Stati Uniti e del Canada e 6 città di Israele, è stato eseguito considerando solo l'andamento della temperatura nel tempo, senza considerare la localizzazione geografica (ovvero le coordinate latitudine e longitudine). Dai risultati ottenuti si nota come i *cluster* individuati rispecchiano in un certo qual modo la distribuzione geografica (quindi città vicine vengono raggruppate in uno stesso *cluster*) e le caratteristiche climatiche (ad esempio, la temperatura oceanica è diversa da quella continentale, per cui vengono individuati cluster diversi).

Questa peculiarità potrebbe non essere una caratteristica generale delle *timeseries,* ma presumibilmente è relativa a dati che riguardano condizioni ambientali, come *timeseries* di pressioni, temperatura, umidità, livelli di precipitazioni, caratteristiche dei venti o misure legate all'inquinamento.

In altri contesti, invece, non considerando anche la posizione, si potrebbe rischiare di perdere delle informazioni, ottenendo modelli di *cluster* meno precisi.

In generale, il processo di *clustering* dipende fortemente dalle caratteristiche specifiche dei dati considerati; tali differenze si enfatizzano nel contesto dei dati spazio-temporali. Una delle limitazioni incontrate nel corso dell'analisi è stata l'interpretazione di alcuni risultati di clustering; tuttavia, l'uso di una misura di centralità (come la media dei valori della metrica appartenenti ad un cluster) o di un indice di performance (come l'indice di Silhouette o il Dunn Index) potrebbero essere utilizzati per identificare cluster importanti e aiutare nell'interpretazione dei risultati. Una seconda limitazione incontrata è stata analizzare e *clusterizzare* dataset di grandi dimensioni (i dataset spazio-temporali per loro natura sono *high dimensional dataset*). Per ridurre la dimensionalità, il *dataset* è stato partizionato rispetto all'attributo *metrica;* dopodichè, per ogni intervallo è stato eseguito un campionamento stratificato su cui sono stati performati i suddetti algoritmi di *clustering*.

Il *next step* principale è approfondire i criteri di ponderazione delle componenti utilizzate per la metrica di distanza e comprendere come le informazioni sugli attributi influenzerebbe il clustering utilizzando ulteriori dataset reali. Inoltre, sarebbe necessario adattare il metodo di cluster proposto per rimuovere potenzialmente *noise* o eventuali *outliers* prima dello sviluppo dei cluster.

Bibliografia

[1] F. M. M. N. a. S. R. S. Kisilevich, «Spatiotemporal clustering,» in *Data mining and Knowledge Discovery Handbook*, New York, Springer, 2010, pp. 855-874.

[2] [Online]. Available: https://www.kaggle.com/naiku007/gps-data.

[3] L. Q. W. J. &. S. Y. DENG Min, «A general method of spatio-temporal clustering analysis».

[4] S. G. S. S. P. P. K.P. Agrawal, «Development and validation of OPTICS based spatio-temporal clustering technique,» *Information Sciences,* vol. 369, pp. 388-401, 2016.

[5] C. C. Aggarwal, Datamining: The textbook, Springer, 2015.

[6] S. G. a. P. Smyth, «Trajectory Clustering with Mixtures of Regression Models».

[7] D. Birant e Kut, «A. ST-DBSCAN: An algorithm for clustering spatial–temporal data.,» 2007.

[8] A. K. Derya Birant, «ST-DBSCAN: An algorithm for clustering spatial–temporal data,» *Data & Knowledge Engineering,* pp. 208-221, 2007.

[9] S. M. I. W. P. F. I. a. I. J. Hesam Izakian, «Clustering Spatiotemporal Data: An Augmented Fuzzy C-Means,» *IEEE TRANSACTIONS ON FUZZY SYSTEMS,* vol. 21, n. 5, pp. 855-868, 2013.

[10] Wikipedia, «Lp space,» [Online]. Available: https://en.wikipedia.org/wiki/Lp_space.

[11] D. J. B. J. Clifford, « Using Dynamic Time Warping to FindPatterns in Time Series,» 1994.

[12] «Dynamic programming example,» [Online]. Available: http://www.phon.ox.ac.uk/jcoleman/old_SLP/Lecture_5/DTW_explanation.html.

[13] A. Carrozza, *Analisi dei modelli di diffusione delle certificazioni ISO 9000 in Europa. Tesi di Laurea Magistrale,* 2018.

[14] J. D. a. S. R. David S. Lamb, «Space-Time Hierarchical Clustering for Identifying Clusters in Spatiotemporal Point Data,» *International Journal of Geo-information.*

[15] D. G. Jeremy Mennis, «Spatial data mining and geographic knowledge discovery—An introduction,» *Computers, Environment and Urban Systems,* n. 33, pp. 403-408, 2009.

[16] K. M., «A spatial scan statistic. Communications in Statistics: Theory and Methods,» p. 1481–1496, 1997.

[17] I. VS, «On detecting space-time clusters. In: Proceedings of the 10th International,» 2004.

[18] H.-P. K. J. S. X. X. Martin Ester, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.,» 1996.

[19] «ST-DBSCAN,» [Online]. Available: https://github.com/eren-ck/st_dbscan.

[20] C. P. a. G. &. N. Andrienko, «Finding arbitrary shaped clusters with related extents in space and time,» in *International Symposium on Visual Analytics Science and Technology* , 2010.

[21] *. A. T. M. B. G. McArdlea, «SPATIO-TEMPORAL CLUSTERING OF MOVEMENT DATA: AN APPLICATION TO TRAJECTORIES GENERATED BY HUMAN-COMPUTER INTERACTION,» in *XXII ISPRS Congress*, Melbourne, Australia, 25 August – 01 September 2012.

[22] D. C. X. X. J. C. Y. Z. N. C. B. H. B. G. Ziyue Chen, «Spatial self-aggregation effects and national division of city-level PM2.5 concentrations in China based on spatio-temporal clustering,» *Journal of Cleaner Production,* n. 207, pp. 875-881, 2019.

[23] L. Y. C. J. L. E. Hwang SY, «Mining mobile group patterns: A trajectory based approach,» 2005.

[24] P. D. Nanni M, «Time-focused clustering of trajectories of moving objects.,» *Journal of Intelligent Information Systems,* pp. 267-289, 2006.

[25] W. W. S. B. B. G. Kang JH, «Extracting places from traces of locations.,» 2004.

[26] I. W. P. F. I. a. I. J. Hesam Izakian Student Member, «Clustering Spatiotemporal Data: An Augmented Fuzzy C-Means,» *IEEE TRANSACTIONS ON FUZZY SYSTEMS,* pp. 855-867, 2013.

[27] K. S. Atsuyuki Okabe, Spatial Analysis Along Networks: Statistical and Computational Methods.

[28] M. Y. S. J. M.-P. Ricardo Oliveira, «4D+SNN: A Spatio-temporal Density-based Clustering Approach with 4D Similarity,» in *2013 IEEE 13th International Conference on Data Mining Workshops*, 2013.

[29] A. A. S. C. T. C. Xin Xiao, «Twitter data laid almost bare: An insightful exploratory analyser,» *Expert Systems With Applications,* pp. 501-517, 2017.

[30] D. Guo e H. H. J. E. Miller, «Multivariate Spatial Clustering and Geovisualization,» in *Geographic Data Mining and Knowledge Discovery*, CRC Press , 2009, pp. 325-345.

[31] A. R. a. J. Hirschberg, «V-Measure: A conditional entropy-based external cluster evaluation measure».

[32] W. L. R. Ka Yee Yeung, «Details of the Adjusted Rand index and Clustering algorithms,» 2001.

[33] B. Desgraupes, «Clustering Indices,» [Online]. Available: https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf.

[34] «sklearn.datasets.make_blobs,» [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html.

[35] «sklearn.cluster.AgglomerativeClustering,» scikit-learn developers (BSD License), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.

[36] s.-l. d. (. License), «sklearn.datasets.make_blobs,» [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html.

[37] s.-l.-e. developpers, «sklearn_extra.cluster.KMedoids,» [Online]. Available: https://scikit-learn-extra.readthedocs.io/en/latest/generated/sklearn_extra.cluster.KMedoids.html.

[38] s.-l. d. (. License)., «sklearn.cluster.AgglomerativeClustering,» [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.

[39] «st-dbscan 0.1.5,» [Online]. Available: https://pypi.org/project/st-dbscan/.

[40] G. R. F. Eren Cakmak from the Data Analysis and Visualization Group and the Department of Collective Behaviour at the University Konstanz funded by the Deutsche Forschungsgemeinschaft (DFG, «st-dbscan 0.1.5,» [Online]. Available: https://pypi.org/project/st-dbscan/.

[41] S.-l. d. (. License), «Nearest Neighbors,» [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html.

[42] «SF's success with dynamic, demand-responsive meter pricing,» [Online]. Available: https://www.aei.org/carpe-diem/sfs-success-with-dynamic-demand-responsive-meter-pricing/.

[43] [Online]. Available: https://www.sfmta.com/projects/sfpark-pilot-program.

[44] S. P. Program. [Online]. Available: https://www.sfmta.com/projects/sfpark-pilot-program.

[45] G. P. &. D. Shoup, «Getting the Prices Right. An Evaluation of Pricing Parking by Demand in San Francisco,» *Journal of the American Planning Association ,* vol. 79, 2013.

[46] W. Vickrey, «My innovative failures in economics,» *Atlantic Economic Journal,* p. 21(1): 1–9, 1993.

[47] P. Ravazzi, Il sistema economico, Carocci.

[48] «sklearn.cluster.KMeans,» [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.

[49] R. T. a. J. F. a. G. V. a. F. D. a. G. A. a. C. H. a. M. P. a. R. Y. a. M. R. a. K. K. a. E. Woods, «Tslearn, A Machine Learning Toolkit for Time Series Data,» *Journal of Machine Learning Research,* vol. 21, n. 118, pp. 1-6, 2020.