

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

---

SCUOLA DI SCIENZE  
Corso di Laurea in Informatica

**Analisi e studio dell'utilità  
dei servizi basati sulla localizzazione  
al variare dei dati personali condivisi**

**Relatore:**  
Dott. Andrea Melis

**Presentata da:**  
Francesco Apollonio

**Co-Relatore:**  
Prof. Luca Bedogni

**Sessione 1  
Anno Accademico 2022/2023**

## **Abstract**

*La protezione dei dati sensibili è una priorità nell'informatica odierna. Abbiamo condotto uno studio in cui analizziamo la variazione della qualità di un Location Based Service all'aumentare della distorsione della posizione inviata.*

*Vogliamo in questo modo studiare il Trade-Off tra questi due parametri con l'obiettivo di ottenere un valore di distorsione ottimale che migliori significativamente la privacy senza compromettere la qualità del servizio.*

*E' stato, inoltre, analizzato il bilanciamento di questo equilibrio al variare di differenti città in cui è stato studiato: dai piccoli paesi alle grandi metropoli.*

*Abbiamo in questo modo scoperto come tale Trade-Off risulti favorire le città con una piccola densità di popolazione nelle quali pur applicando una distorsione maggiore la qualità del servizio rimane eccellente. Lo studio si conclude con interessanti spunti per sviluppi futuri riguardo l'argomento.*

## GLOSSARIO

- **API:** Application Programming Interface è un insieme di regole, protocolli e strumenti specificamente progettati per consentire l'interazione e la comunicazione tra due o più software. Essa definisce i metodi standardizzati con i quali le diverse componenti di un sistema software possono richiamare e scambiare informazioni tra loro.

L'API funge da interfaccia tra le diverse applicazioni, consentendo loro di accedere e utilizzare le funzionalità offerte da un determinato servizio o piattaforma senza dover conoscere i dettagli interni dell'implementazione.

- **Point of Interest (P.O.I.):** è un termine utilizzato nell'ambito della geolocalizzazione e della navigazione per descrivere una posizione specifica di interesse o rilevanza in un determinato contesto. Si tratta di un punto fisico o virtuale che rappresenta un luogo, un'entità o un'attrazione significativa, come ad esempio un ristorante, un museo, un'azienda o un punto di interesse turistico.

Formalmente, un Point of Interest è definito come una coppia di coordinate geografiche (latitudine e longitudine) che identifica un punto sulla superficie terrestre. Oltre alle coordinate spaziali, un P.O.I. può includere anche informazioni aggiuntive, come il nome, la descrizione, le categorie di appartenenza e altri attributi che forniscono ulteriori dettagli sul luogo o sull'entità rappresentata.

- **Query:** Una query di ricerca è una richiesta o una domanda formulata dall'utente per ottenere informazioni specifiche o per trovare risultati rilevanti all'interno di un sistema di ricerca, come un motore di ricerca o un database. Formalmente, una query di ricerca è una sequenza di parole chiave, frasi o criteri di ricerca che vengono inseriti in un'apposita interfaccia o campo ed inviati al servizio specifico al fine di recuperare informazioni pertinenti.
- **Location Based Service:** Si tratta di un servizio in grado di fornire funzionalità, informazioni o contenuti basati sulla geolocalizzazione dell'utente che ne fa richiesta.

- **Scraping:** Processo nel quale un'applicazione estrae informazioni utili a partire da una grossa quantità di dati.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Stato dell'arte</b>	<b>7</b>
2.1	La situazione attuale . . . . .	7
2.2	Il problema in analisi . . . . .	9
2.3	Strategia . . . . .	10
<b>3</b>	<b>Analisi Progettuale</b>	<b>12</b>
3.1	Gli strumenti utilizzati . . . . .	12
3.1.1	Google Maps . . . . .	12
3.1.2	Python . . . . .	13
3.1.3	File CSV . . . . .	13
3.2	Approccio metodologico . . . . .	14
3.3	Ottenere Point of Interest a partire da una posizione precisa . . . . .	15
3.3.1	Un esempio di richiesta ed i risultati ottenuti . . . . .	17
3.4	Ottenere Point of Interest a partire da una posizione distorta . . . . .	18
3.4.1	La distorsione della posizione . . . . .	18
3.4.2	Il codice per il calcolo della circonferenza . . . . .	22
3.4.3	Un esempio di richiesta ed i risultati ottenuti . . . . .	23
3.5	Metodi di valutazione . . . . .	24
3.5.1	Esempio di valutazione di un caso d'uso . . . . .	25
<b>4</b>	<b>Risultati</b>	<b>27</b>
4.1	Definizione delle variabili . . . . .	27
4.1.1	Le città definite . . . . .	27

---

4.1.2	I parametri definiti . . . . .	28
4.2	Analisi dei risultati . . . . .	29
4.2.1	Il raggio di ricerca . . . . .	29
4.2.2	Il numero di punti generati casualmente . . . . .	36
4.2.3	I P.O.I. al variare della distorsione applicata . . . . .	41
4.3	L'efficienza in termini di utilizzo di risorse . . . . .	49
4.3.1	La banda di rete utilizzata . . . . .	49
4.3.2	L'impiego di risorse hardware . . . . .	50
<b>5</b>	<b>Conclusioni</b>	<b>51</b>
5.1	Il valore di distorsione ottimale . . . . .	51
5.1.1	Il valore ottimale nelle grandi metropoli . . . . .	51
5.1.2	Il valore ottimale nei piccoli paesi . . . . .	52
5.2	Sviluppi futuri . . . . .	53

# Capitolo 1

## Introduzione

La protezione dei dati personali degli utenti è, da sempre, uno dei temi più importanti nel mondo della sicurezza informatica.

Sapere come i dati personali vengono utilizzati dagli applicativi con cui ci interfacciamo quotidianamente può risultare fondamentale al fine di poterli preservare e proteggere.

Sebbene numerose iniziative legislative siano state create nel tempo [1] per migliorare la gestione e proteggere la privacy degli utenti, dal punto di vista tecnologico le sfide da affrontare sono all'ordine del giorno.

Inseriamo dunque questo lavoro all'interno di tale contesto, studiando la possibilità di manipolare e quindi approssimare i nostri dati prima di condividerli col servizio che ne fa richiesta.

In questi termini ci chiediamo in particolare quanto possa variare la qualità di un servizio basato sulla localizzazione (*Location Based Service*) ricevendo una posizione particolarmente meno precisa del normale.

Conduciamo pertanto uno studio sul Trade-Off tra il livello di distorsione della posizione, che garantisce un miglioramento della privacy dell'utente, e la possibile e diretta riduzione della qualità del servizio in uso.

Effettuiamo lo studio di questo Trade-Off utilizzando uno dei Location Based Service più famosi al mondo: Google Maps.

Utilizzando un applicativo fornito da tale servizio conduciamo dunque degli esperimenti

di ricerca di Point Of Interest utilizzando sia una posizione estremamente precisa che una posizione da noi distorta, fino ad un massimo di 2500 di metri.

Svolgiamo tali esperimenti in diversi ambienti con l'obiettivo di analizzarne le differenze: dalle grandi città, caratterizzate da una densa popolazione, ai piccoli paesini di provincia.

I risultati ottenuti sono interessanti. Si identifica un valore di distorsione ottimale che offre all'utente un significativo miglioramento della privacy, mantenendo allo stesso tempo un alto livello di qualità del servizio utilizzato.

Nell'analisi di questo equilibrio, inoltre, emerge che il valore di distorsione risulta notevolmente più elevato in ambienti meno densamente popolati (e quindi con una minore densità di P.O.I.), consentendo così di preservare un livello ancora maggiore di privacy rispetto alle grandi metropoli.

Il resto di questo lavoro è suddiviso come segue:

*Capitolo 2* descrive l'attuale stato del progresso scientifico in questo ambito di ricerca e definisce il nostro modo rivoluzionario (rispetto agli studi precedenti) di affrontarlo;

*Capitolo 3* descrive dal punto di vista tecnico il nostro approccio al problema, quindi gli strumenti utilizzati;

*Capitolo 4* sintetizza servendosi di più grafici l'interesse dei risultati ottenuti;

*Capitolo 5* termina il nostro lavoro, analizzando e presentando gli esiti dei nostri studi, per poi concludere con alcune considerazioni riguardanti le potenziali direzioni di sviluppi futuri di questa ricerca.



# Capitolo 2

## Stato dell'arte

### 2.1 La situazione attuale

Quotidianamente affrontiamo tutti, probabilmente in maniera inconsapevole, un problema: la diffusione dei nostri dati personali ad aziende o enti che, per il servizio che ci offrono, potrebbero farne a meno o comunque richiederli in maniera **meno precisa**.

Il problema della diffusione incontrollata dei nostri dati personali, dunque, esiste ed è estremamente concreto.

Occorre, infatti, aver chiaro che anche a partire da pochissimi dati è possibile ottenere informazioni molto rilevanti riguardo la vita delle persone.

Lo dimostrano diversi studi, in particolare uno londinese [3] che indaga sulle vulnerabilità umane di fronte all'ingegneria sociale nel contesto dei social network.

Per far fronte a questo genere di problematiche, in tempi recenti sono state applicate sempre di più nuove tecniche, presumibilmente più sicure, per il raccoglimento di grandi quantità di dati sensibili, come la **K-Anonymity** [11].

La K-Anonymity può, in breve, essere definita come segue: Un certo dataset viene definito "k-anonymous" se ogni persona nel dataset è indistinguibile da k-1 utenti.

Nonostante queste tecniche siano il prodotto di ampie ricerche, spesso risultano vulnerabili ad attacchi, come dimostrato dettagliatamente da uno studio del 2006 [8].

Aldilà di questo, si presume, comunque, che ogni utente ad oggi sia in grado di valutare attentamente le autorizzazioni (in termini di privacy) richieste da un'applicazione e di determinare se si sente a proprio agio nel concederle.

Tuttavia, ricerche condotte da Kelley et al. [7] e da altri ricercatori [6] hanno evidenziato come questo approccio presenti diverse lacune: la maggior parte degli utenti non comprende in modo esaustivo (o comunque come ci si aspetterebbe) le implicazioni delle proprie decisioni in quest'ambito.

Un'ulteriore analisi, ancora più approfondita, riguardo al rapporto tra condivisione dei propri dati personali e consapevolezza di ciò è stata condotta in uno studio della Carnegie Mellon University [9].

In primis, lo studio espone ancora una volta come, nonostante i diversi sforzi nel rendere il quanto più possibile chiare le applicazioni della condivisione dei propri dati, l'utente medio dimostra ancora scarsa comprensione di ciò.

E' interessante però notare i risultati di un successivo sondaggio condotto dalla stessa università: questo riporta come gli utenti si sentirebbero maggiormente a proprio agio nella condivisione di informazioni personali se dovessero poter avere un maggior controllo sulle stesse.

In particolare, si evidenzia come l'utente preferisca l'accettazione di determinate condizioni di condivisione dei dati purché queste siano estremamente chiare, limpide, ed il quanto più possibile gestibili (ad esempio potendo decidere di modificarle quando si vuole).

A tal proposito riteniamo importante sottolineare come la comunità scientifica stia lavorando e facendo progressi a riguardo attraverso progetti di **Usable Privacy** [10]. Questo campo di ricerca pone i suoi obiettivi nella progettazione e nell'implementazione di misure e sistemi di privacy che siano facili da usare e comprensibili, consentendo alle persone di gestire efficacemente le proprie informazioni personali e mantenere la loro privacy in vari contesti. Tali studi si concentrano in particolare nel colmare il divario tra politiche sulla privacy, tecnologie ed esperienza utente, rendendo la protezione dei dati un qualcosa di più accessibile e manipolabile per le persone.

In supporto di questo genere di problematiche, sono state inoltre create vere e proprie

applicazioni [5] volte ad aiutare l'utente a prendere consapevolmente decisioni nell'ambito della condivisione dei propri dati personali.

## 2.2 Il problema in analisi

Il problema che questa tesi intende affrontare è quindi il seguente: sarebbe possibile gestire i dati che diffondiamo in base a quanto ognuno di noi lo ritenga necessario?

Nell'analisi di questo problema, una maggiore attenzione è rivolta ai servizi basati sulla condivisione della propria posizione (*Location Based Service*); questi rappresentano infatti una categoria di applicazioni in cui la 'qualità' dei dati diffusi è maggiormente controllabile e risulta, inoltre, particolarmente rilevante ai fini dell'utilizzo del servizio stesso.

Come spiegato da Bedogni et al. [2], è inoltre chiaro che la posizione di un utente potrebbe rivelare aspetti privati della sua stessa vita.

Esiste, dunque, chiaramente un **Trade-Off** tra l'utilità di determinati dati, in termini di qualità del servizio ricevuto poi dall'utente, e la privacy dello stesso, con riferimenti a possibili attacchi o informazioni che utenti malintenzionati potrebbero comprendere basandosi sui dati ai quali hanno accesso (come la posizione esatta di un individuo).

Sarebbe dunque importante se la condivisione di questi dati fosse valutata direttamente dall'utente interessato, il quale dovrebbe essere in grado di decidere (e quindi comprendere al meglio) quali dati condividere e quali mantenere privati [4].

L'obiettivo principale di questo studio è dunque quello di analizzare proprio tale Trade-Off tra la qualità del servizio ricevuto e la privacy garantita dal servizio stesso, basando il tutto sulla manipolazione della precisione della posizione inviata ad un *Location Based Service*.

Definiamo alcune *research questions* che ci hanno aiutato nella definizione dei confini del nostro problema:

E' veramente necessario condividere la nostra posizione in maniera estremamente precisa con l'App che, in fin dei conti, ci deve solo mostrare una serie di risultati generici?

Cosa accadrebbe se, invece che inviare una posizione precisa al metro, ne inviassimo una

molto più approssimata, ad esempio con una precisione di più o meno 100 metri? Riusciremmo comunque ad usufruire allo stesso modo di quel servizio? Se sì, a discapito di qualcosa?

In questo lavoro andiamo in particolare ad analizzare quello che potrebbe essere il comportamento di un *Location Based Service* nel momento in cui gli viene dato in input una posizione meno precisa rispetto a quella che normalmente un utente invierebbe. Possiamo in questo modo studiare le differenze tra la condivisione di una posizione accurata e la condivisione di una posizione approssimata. Vogliamo in particolare riportare le differenze di una qualità del servizio presumibilmente migliore che si otterrebbe nel primo caso ed una peggiore che si otterrebbe nel secondo, considerando però come in quest'ultimo si vada a ottenere un guadagno in termini di privacy.

Analizziamo quindi se, ed eventualmente fino a che punto, converrebbe distorcere la precisione della nostra posizione con l'obiettivo di ottenere una minore diffusione dei nostri dati sensibili, pur mantenendo la qualità del servizio di cui usufruiamo il quanto più accettabile possibile.

## 2.3 Strategia

Decidiamo di prendere in analisi uno dei *Location Based Service* più utilizzati: **Google Maps**.

Questo con l'obiettivo di lavorare su un qualcosa di noto e diffuso e poter quindi analizzare nel dettaglio i guadagni o le perdite nel Trade-Off studiato.

Utilizziamo, dunque, un API di **Google Maps** che permette di ottenere una lista di *Point of Interest* a partire da una posizione (data da coordinate di latitudine e longitudine) ed una *parola chiave*.

Analizziamo anzitutto la differenza tra i *P.O.I.* ottenuti a partire dalla posizione esatta, nonché molto precisa, e quelli ottenuti da una posizione da noi distorta (quindi approssimata), in modo tale da nascondere la posizione corrente.

In secondo luogo studiamo poi l'utilizzo delle risorse hardware e quindi energetiche del-

---

l'eventuale dispositivo che, volendo guadagnare maggiore privacy, dovrà in qualche modo lavorare **in locale** alla distorsione della posizione reale prima di inoltrarla al fornitore del servizio in uso.

Questi studi vengono ovviamente fatti sulla base della stessa *parola chiave*, facendo variare invece il luogo in cui essi sono compiuti (grande città, piccola città, paesino di campagna, ecc.) e le variabili (discusse opportunamente in seguito) che definiscono i parametri di ricerca.

In generale analizziamo quindi la variazione della qualità del servizio e l'utilizzo delle risorse hardware a discapito di un potenziale guadagno in termini di privacy.

# Capitolo 3

## Analisi Progettuale

### 3.1 Gli strumenti utilizzati

#### 3.1.1 Google Maps

Dunque, partendo da questa premessa, l'applicativo scelto per eseguire i nostri test è **Google Maps**<sup>1</sup>.

Google Maps è infatti uno dei servizi più utilizzati per la navigazione satellitare e la ricerca di *Point of Interest* attraverso una posizione.

Google offre, inoltre, una serie molto interessante di *API* estremamente utili per permettere ai programmatori di accedere a singole funzionalità di applicativi come Maps.

Tra le tante API disponibili quella da noi scelta è **Find Place**<sup>2</sup>.

**Find Place** consente di eseguire ricerche di luoghi in base a criteri come la tipologia della località cercata, l'indirizzo, il numero di telefono o il nome stesso.

L'API scelta restituisce i risultati corrispondenti a tali criteri sotto forma di oggetti JSON contenenti informazioni sui *P.O.I.* trovati come il nome, le coordinate geografiche, indirizzo, numero di telefono e altro ancora.

Ciò che interessa a noi, tra tutte le informazioni disponibili, è il nome dei P.O.I. trovati (purché sia già appurato che la loro posizione sia nel range specificato).

---

<sup>1</sup><https://www.google.it/maps/>

<sup>2</sup><https://developers.google.com/maps/documentation/places/web-service/search-find-place>

Attraverso il nome riusciamo quindi a capire quanti differenti P.O.I. sono stati trovati e possiamo inoltre effettuare confronti tra i risultati ottenuti a partire da una posizione distorta e non.

Importante sottolineare una limitazione nell'utilizzo dell'API Find Place che potrebbe essere rilevante al fine dell'analisi dei risultati finali: il numero massimo di risultati con i quali l'API risponde ad una singola query è di 20.

### 3.1.2 Python

Per portare a termine questo studio è necessaria la scelta di un linguaggio di programmazione. Risulta infatti essenziale avere un modo per:

- Utilizzare i servizi offerti dall'API *Find Place*
- Distorcere la posizione di partenza. (Non permesso nativamente da *Find Place*)
- Automatizzare la ricerca di *P.O.I.*
- Automatizzare l'analisi delle differenze tra ricerca con posizione distorta e non.
- Automatizzare e organizzare la raccolta di tutte le suddette informazioni in un singolo file.

Dunque, tra i diversi linguaggi disponibili, scegliamo Python in quanto semplice nell'utilizzo.

Nelle successive sezioni vi sono delle parti dedicate alla spiegazione del codice Python utilizzato; queste non risultano, tuttavia, esaustive essendo il nostro lavoro non incentrato su di esse.

Per approfondimenti legati alla parte di programmazione fare riferimento alla repository Open Source del progetto disponibile su [Github](#).

### 3.1.3 File CSV

Per poter raccogliere le grandi quantità di dati ottenuti e poterli poi studiare decidiamo di utilizzare i file CSV.

Il formato CSV rappresenta, infatti, uno dei modi più semplici per rappresentare dati in forma tabellare all'interno di un semplice file di testo e poterli poi manipolare al meglio per sviluppare, ad esempio, dei grafici.

## 3.2 Approccio metodologico

Quello che facciamo è dunque effettuare una query a Maps (tramite l'API *Find Place*) utilizzando una parola chiave e delle coordinate precise; otteniamo così una serie di *Point of Interest*.

Successivamente, eseguiamo la stessa query, con la stessa parola chiave, ma fornendo a *Find Place* una posizione in qualche modo imprecisa, lavorando su un parametro di distorsione della posizione che possa variare a nostro piacimento; otteniamo ancora una volta una serie di *Point of Interest*.

Infine, ottenuti i due risultati, non ci resta che analizzarli, rapportarli, e capire quanto la distorsione della nostra posizione iniziale abbia influito nell'ottenimento dei risultati. Ci chiediamo in particolare:

- Come variano i *Point of Interest* ottenuti in base alla distorsione della posizione?
- La qualità del servizio offerto da Maps fino a che parametro di distorsione rimane accettabile?
- La distorsione della posizione operata in locale quanto influisce sul dispositivo che la esegue in termini di utilizzo di risorse hardware?

Analizzando il problema e il Trade-Off che vogliamo studiare capiamo di dover lavorare su 3 cose in particolare:

- Ottenere *Point of Interest* a partire da una posizione precisa.
- Ottenere *Point of Interest* a partire da una posizione distorta.
- Valutazione dei risultati ottenuti.



### 3.3 Ottenere Point of Interest a partire da una posizione precisa

L'ottenimento dei *P.O.I.* a partire da una posizione precisa trova risoluzione nella semplice applicazione dell'API *Find Place* tramite il codice Python mostrato in figura 3.1.

E', infatti, sufficiente generare un particolare URL (definito dalle linee guida dell'API) ed eseguire una richiesta HTTP con una delle tante librerie Python che lo consentono; quella da noi scelta è **Request**.

Definiamo dunque le coordinate di Latitudine e Longitudine, la parola chiave e il range entro il quale vogliamo avvenga la ricerca dei luoghi.

Creiamo ora l'URL, come possiamo vedere tra le righe 13 e 18 del codice in 3.1 e inviamo la richiesta HTTP come mostrato a riga 22.

Decidiamo poi di salvare i risultati ottenuti all'interno di un file per una migliore manipolazione successiva.

```
1 def search_nearby_places(place_to_search, x_coordinate, y_coordinate,
2 search_range = 500):
3     """Function that implement the Google API to search for a place
4     around coordinates using keywords.
5     Args:
6     place_to_search (string): keywords to search
7     x_coordinate (float): longitude of the coordinates
8     y_coordinate (float): latitude of the coordinates
9     search_range (int, optional): meters around we want to search
10    places. Defaults to 500.
11    Returns:
12    string: the name of the file containing the research results
13    """
14    url = ("https://maps.googleapis.com/maps/api/place/nearbysearch/
15    json?location=" +
16          str(x_coordinate) + "%2C" +
17          str(y_coordinate) + "&radius=" +
18          str(search_range) + "&keyword=" +
19          place_to_search + "&key=" +
20          GOOGLE_API_KEY)
21    payload = {}
22    headers = {}
23
24    response = requests.request("GET", url, headers=headers, data=
25    payload)
26
27    # creation of the file containing the research results
28    filename = str(place_to_search + "_in_" + str(x_coordinate)[:8] +
29    "_" + str(y_coordinate)[:8] + ".txt")
30    with open(filename, "a") as file:
31        for line in response.text.split('\n'):
32            if 'name' in line:
33                file.writelines(line + '\n')
34    print("Research of " + place_to_search + " in " + str(x_coordinate)
35    + "_" + str(y_coordinate) + " completed!\n")
36    return filename
```

Listing 3.1: Ottenimento di P.O.I. a partire da una posizione precisa.

### 3.3.1 Un esempio di richiesta ed i risultati ottenuti

A scopo illustrativo supponiamo di voler ottenere i *Point Of Interest* rappresentanti i "Bar" che si trovano in un range di 500 metri dalle 'Due Torri' di Bologna.

Definiamo dunque le nostre variabili:

- `place_to_search = "Bar"`
- `x_coordinate = 44.494214008658446` (Longitudine delle Due Torri)
- `y_coordinate = 11.346444195740402` (Latitudine delle Due Torri)
- `search_range = 500`

Eseguendo dunque il codice mostrato in 3.1 con queste variabili definite otteniamo un JSON contenente 20 Point Of Interest.

Attraverso un breve processo di scraping volto a filtrare, tra tutte le informazioni, solo quelle riguardanti il nome dei P.O.I. trovati, otteniamo i seguenti risultati.

```
1     "name" : "Lime Bar",
2     "name" : "Bar Maurizio",
3     "name" : "Bar Vittorio Emanuele",
4     "name" : "NU Lounge Bar",
5     "name" : "Bar Caffè delle Drapperie",
6     "name" : "Brek Cocktail Bar",
7     "name" : "Green Bar S.N.C.",
8     "name" : "Roxy Bar",
9     "name" : "Bar Tabaccheria Santunione G.",
10    "name" : "Bar 48 Il Quarantotto L'arte Del Caffè",
11    "name" : "Bar Al Bricco D'oro",
12    "name" : "Bar della Pioggia",
13    "name" : "Bar Zecca",
14    "name" : "Bar Due Torri",
15    "name" : "Le Mercanzie Lounge Bar",
16    "name" : "Bar Otello",
17    "name" : "Bar Tabacchi",
18    "name" : "Bar",
19    "name" : "Cafe' bar",
```

20

```
"name" : "Black & White - Vintage American bar"
```

## 3.4 Ottenere Point of Interest a partire da una posizione distorta

Ottenere *P.O.I.* utilizzando l'API *Find Place* ma fornendo una posizione imprecisa, distorta, non è un qualcosa di nativamente possibile tramite il suddetto applicativo. Dunque, è necessario trovare un modo che possa simulare, il quanto più fedelmente possibile, la ricerca di *P.O.I.* utilizzando una posizione imprecisa rispetto a quella reale.

### 3.4.1 La distorsione della posizione

Per fare ciò partiamo da un parametro variabile rappresentante, in metri, la distorsione della nostra reale posizione. Chiamiamo questo parametro *D*.

Supponiamo ora di voler distorcere una posizione reale avente coordinate geografiche *X* e *Y* per poter poi calcolare dei *P.O.I.* a partire da essa.

Definiamo una circonferenza di raggio *D* intorno alle coordinate di partenza e generiamo casualmente all'interno della stessa una serie di nuove posizioni rappresentate ciascuna da coppie di coordinate (Longitudine e Latitudine).

A questo punto eseguiamo la normale ricerca di *P.O.I.* per ognuna di queste coordinate, esattamente come spiegato nel paragrafo precedente.

Otteniamo così, per ognuno dei punti generati, dei risultati differenti (analizziamo successivamente quanto differenti in base alla variazione di alcuni parametri).

**L'intersezione** di tutti i risultati ottenuti rappresenta esattamente l'esito della nostra ricerca di *P.O.I.* attorno ad *X* ed *Y* con una distorsione della posizione iniziale di *D* metri.

## Il calcolo della circonferenza e i problemi affrontati

Il calcolo di una circonferenza esatta intorno ad un punto rappresentato da coordinate geografiche non è affatto immediato.

I problemi nascono nel momento in cui consideriamo che la superficie terrestre non è una sfera perfetta. Andando, infatti, ad ignorare questa caratteristica forma del pianeta e calcolando, dunque, la circonferenza come se stessimo lavorando su una sfera, i risultati ottenuti non sono decisamente soddisfacenti, se pur comunque interessanti.

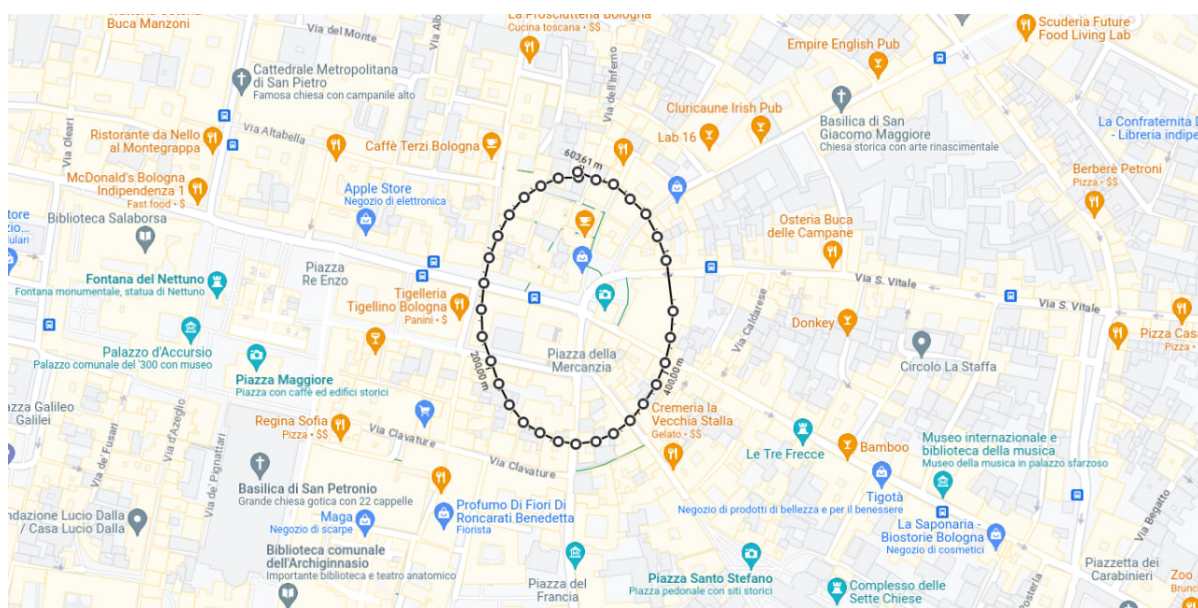


Figura 3.1: Circonferenza ottenuta supponendo la terra una sfera.

Come possiamo notare in 3.1, approssimando la terra ad una sfera perfetta, otteniamo un ovale piuttosto che una circonferenza intorno al punto interessato.

Andare poi a lavorare con la posizione distorta rappresentata da quest'ovale potrebbe generare importanti imprecisioni, soprattutto considerando che il parametro di distorsione inizialmente definito è il raggio di un'ipotetica circonferenza.

Immaginiamo, ad esempio, il caso in cui, decidendo di generare dei punti all'interno della circonferenza per il calcolo dei P.O.I. con posizione distorta, ne venga generato uno

nelle immediate prossimità dell'intersezione della diagonale maggiore; in questo caso i risultati ottenuti a partire da quel punto sarebbero assolutamente fuorvianti. Questo banalmente perché si trovano al di fuori dello spazio intorno al punto centrale in cui ci si aspettava di ricercare risultati.

### **Proiezione azimutale equidistante: la soluzione al nostro problema**

I risultati ottenuti con la precedente strategia non risultano soddisfacenti. Decidiamo perciò di seguire un altro approccio valido per ottenere una circonferenza intorno a delle coordinate geografiche considerando la reale forma della terra.

La soluzione al problema è l'applicazione della **proiezione azimutale equidistante** nel calcolo della circonferenza.

La proiezione azimutale equidistante, come spiega Wikipedia<sup>3</sup> è una proiezione cartografica equidistante. Ha la peculiarità che, rispetto ad un determinato punto geografico scelto come centro, conserva inalterati, rispetto alle altre zone, sia l'angolo (azimut) che la distanza, e quindi tutti gli altri punti sulla mappa sono proporzionatamente ricollocati dalla loro relativa posizione rispetto a questo centro. Questo si realizza proiettando tutti i punti della terra su una superficie piana, ad essa tangente.

Il risultato di quest'applicazione si può banalmente riassumere nel riuscire a calcolare una circonferenza esatta attorno al punto geografico interessato.

---

<sup>3</sup>[https://it.wikipedia.org/wiki/Proiezione\\_azimutale\\_equidistante](https://it.wikipedia.org/wiki/Proiezione_azimutale_equidistante)

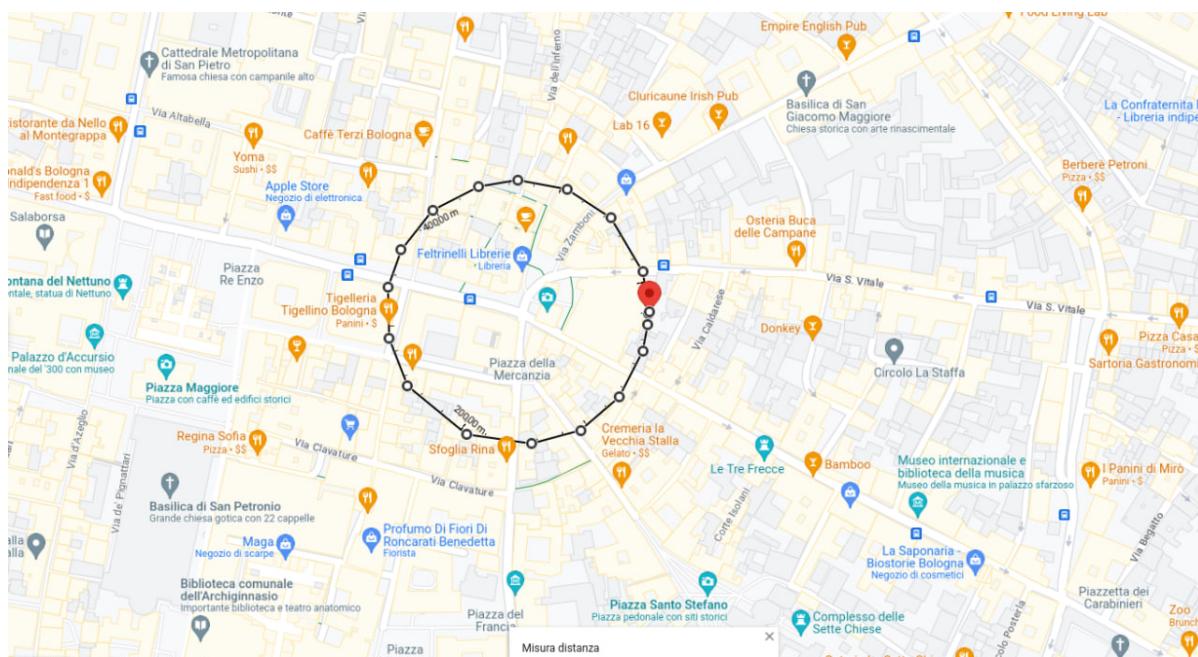


Figura 3.2: Circonferenza ottenuta applicando la proiezioni azimutale equidistante

Come possiamo notare in 3.2 la circonferenza ottenuta applicando la suddetta tecnica risulta ideale.

Questa è stata quindi la soluzione scelta definitivamente nel processo di ottenimento di P.O.I. a partire da una posizione distorta.

### 3.4.2 Il codice per il calcolo della circonferenza

```
1 local_azimuthal_projection = "+proj=aeqd +R=6371000 +units=m +lat_0\n    ={} +lon_0={}".format(x_coordinate, y_coordinate)\n2\n3 wgs84_to_aeqd = Transformer.from_proj('+proj=longlat +datum=WGS84 +\n    no_defs', local_azimuthal_projection)\n4 aeqd_to_wgs84 = Transformer.from_proj(local_azimuthal_projection, '+\n    proj=longlat +datum=WGS84 +no_defs')\n5\n6 # Get polygon with lat lon coordinates\n7 point_transformed = Point(wgs84_to_aeqd.transform(x_coordinate,\n    y_coordinate))\n8\n9 buffer = point_transformed.buffer(radius)\n10 circle = transform(aeqd_to_wgs84.transform, buffer)
```

Listing 3.2: Calcolo di una circonferenza attraverso la proiezione azimutale equidistante

Il codice mostrato in 3.2 genera le coordinate di un cerchio intorno ad un punto geografico esatto definito tramite latitudine e longitudine.

In particolare la prima riga definisce la stringa `local_azimuthal_projection` che rappresenta la proiezione azimutale equidistante locale.

Questa stringa viene creata utilizzando la formattazione di stringhe in Python (`format()`) per sostituire i valori delle coordinate `x_coordinate` e `y_coordinate` nel formato richiesto dalla proiezione.

Le successive due linee creano due oggetti `Transformer` utilizzando la libreria *pyproj*.

L'oggetto `wgs84_to_aeqd` converte le coordinate WGS84 nella proiezione azimutale equidistante locale, mentre l'oggetto `aeqd_to_wgs84` esegue la conversione inversa, ovvero dalla proiezione azimutale equidistante locale alle coordinate WGS84.

La funzione `from_proj()` prende come argomenti le stringhe che rappresentano le proiezioni di partenza e di destinazione.

La linea successiva crea un oggetto `Point` della libreria *shapely* utilizzando le coordinate WGS84 trasformate nella proiezione azimutale equidistante locale. Questo oggetto rappresenta un punto geometrico.



La variabile `buffer` viene inizializzata come un buffer intorno al punto trasformato. Il buffer rappresenta una regione circolare con un raggio specificato dalla variabile `radius`.

Infine, la variabile `circle` viene assegnata alla trasformazione inversa del buffer dalla proiezione azimutale equidistante locale alle coordinate WGS84 utilizzando la funzione `transform()` della libreria `pyproj`.

Il risultato di tutto ciò è un poligono che rappresenta un cerchio avente il centro in coordinate latitudine-longitudine definite.

### 3.4.3 Un esempio di richiesta ed i risultati ottenuti

A scopo illustrativo supponiamo di voler effettuare nuovamente la stessa ricerca di *Point Of Interest* fatta in precedenza.

Questa volta, però, andiamo ad applicare una distorsione alla posizione esatta in modo da poter rimanere più anonimi rispetto alla precedente query.

Definiamo dunque le nostre variabili (le quali saranno per la maggior parte le stesse della ricerca precedente):

- `place_to_search` = "Bar"
- `x_coordinate` = 44.494214008658446 (Longitudine delle Due Torri)
- `y_coordinate` = 11.346444195740402 (Latitudine delle Due Torri)
- `search_range` = 500
- `distortion` = 300 (raggio in metri dal quale vogliamo creare la circonferenza rappresentante la distorsione)
- `random_point` = 10 (numero di punti casuali generati per l'approssimazione della posizione)

Dopo il solito processo di scraping i risultati ottenuti sono i seguenti:

```
1     "name" : "Brek Cocktail Bar",
2     "name" : "Bar Otello",
3     "name" : "Lime Bar",
4     "name" : "Bar Caffè delle Drapperie",
5     "name" : "Bar Vittorio Emanuele",
6     "name" : "NU Lounge Bar",
7     "name" : "Le Mercanzie Lounge Bar",
8     "name" : "Roxy Bar",
9     "name" : "Bar Tabacchi"
```

## 3.5 Metodi di valutazione

Una volta ottenuti, dunque, sia i P.O.I. generati a partire dalla posizione esatta sia quelli generati a partire da una posizione imprecisa (distorta), possiamo finalmente passare all'analisi dei risultati.

Il modo migliore, al meglio della nostra conoscenza, per poter capire quanto la distorsione (decisa arbitrariamente) della posizione iniziale abbia influito nella ricerca dei P.O.I., è analizzare **l'intersezione** tra i risultati ottenuti utilizzando la posizione reale, quindi precisa, e quelli ottenuti utilizzando la posizione distorta, quindi imprecisa. Ci riferiremo successivamente all'intersezione di questi risultati con le parole "*P.O.I. utili*".

In questa maniera possiamo capire quanti dei punti che di norma otterremmo utilizzando una posizione molto precisa riusciamo ad ottenere utilizzando una posizione da noi distorta.

Più è alto il numero di punti ottenuti nell'intersezione, migliore è la qualità dei risultati ottenuti distorcendo la posizione (in termini di numero di P.O.I. trovati nell'intorno della posizione precisa).

Al contrario, invece, più basso è il numero di punti in comune ottenuti minore è la qualità dei risultati.

Ovviamente, più questo numero tende a zero, peggiore diventa (a causa della posizione arbitrariamente distorta) la qualità del servizio che stiamo usando, fino a divenire inuti-

lizzabile.

In linea generale, nella futura analisi dei risultati, ci aspettiamo che all'aumentare della variabile di distorsione della posizione diminuisca il numero di P.O.I. ottenuti nell'intersezione.

### 3.5.1 Esempio di valutazione di un caso d'uso

Per spiegare al meglio la nostra strategia analizziamo ora i file contenenti i risultati ottenuti a partire dai due esempi fatti nei capitoli precedenti.

Quello che facciamo in particolare è verificare i P.O.I. in comune contenuti nei due file di risultati.

Analizziamo il risultato dell'intersezione tra i due file.

```
1      "name" : "Brek Cocktail Bar",
2      "name" : "Bar Otello",
3      "name" : "Lime Bar",
4      "name" : "Bar Caffè delle Drapperie",
5      "name" : "Bar Vittorio Emanuele",
6      "name" : "NU Lounge Bar",
7      "name" : "Le Mercanzie Lounge Bar",
8      "name" : "Roxy Bar",
9      "name" : "Bar Tabacchi"
```

Notiamo immediatamente che di fatto **ogni elemento** ottenuto dalla ricerca di P.O.I. con posizione distorta è presente nei risultati ottenuti a partire dalla posizione esatta.

Cosa possiamo dedurre da quest'analisi, in questo specifico caso?

Deduciamo che una distorsione della nostra posizione reale di **300 metri**, fatta generando esattamente **10 punti** casuali in quel raggio, nella ricerca di P.O.I. **in una grande città**, sembra non aver drasticamente influito sulla qualità del servizio offerto da Google Maps.

Dunque, una maggiore riservatezza della propria posizione (di 300 metri) non influisce

significativamente a peggiorare la qualità del servizio.

Abbiamo perciò provato ad aumentare il fattore riservatezza (aumentando la distorsione), così come il numero di punti casualmente generati e il tipo di luogo in cui svolgere i nostri esperimenti, per poi analizzarne i risultati.

# Capitolo 4

## Risultati

### 4.1 Definizione delle variabili

#### 4.1.1 Le città definite

Per la raccolta di risultati utili, e quindi l'analisi degli stessi, eseguiamo diversi test volti a capire con quali parametri si generano i risultati più attendibili o comunque più facilmente interpretabili.

Per questi scopi eseguiamo più di **700 singole query**: circa 150 per ciascuna città scelta per effettuare i nostri test.

Le città in cui eseguiamo i test sono inizialmente 4.

Il nostro obiettivo è applicare i nostri studi in città con differenti *densità di popolazione* in quanto, logicamente, ci aspettiamo che più alta sia la densità di popolazione in una cittadina più questa sia ricca di P.O.I. che soddisfino le esigenze dei residenti.

Definiamo dunque 4 città in cui i risultati, data la densità prevista dei P.O.I., possano variare fra di loro; questo ci permette, secondo le nostre ipotesi, di studiare quanto la distorsione della posizione possa essere più o meno critica per la qualità del servizio utilizzato anche in base a dove ci troviamo.

I nostri test sono svolti nelle seguenti città:

- **Napoli**, una delle città Italiane con la più alta densità di popolazione: 7 707

abitanti per kilometro quadrato. <sup>1</sup>

- **Bologna**, ulteriore città Italiana discretamente popolosa, se pur meno di Napoli: 2 754 abitanti per kilometro quadrato. <sup>2</sup>
- **Aradeo**, un piccolo paesino di provincia: 1 149 abitanti per kilometro quadrato. <sup>3</sup>
- **Lopagno**, un paesino di montagna con un totale di 693 abitanti: escluso dopo una primissima analisi dei risultati dei test. La quantità di P.O.I. presenti in zona risulta *inaccettabile*, in termini numerici, ai fini della validità dello studio effettuato. <sup>4</sup>

### 4.1.2 I parametri definiti

Una volta definite le città in cui eseguire i nostri test spostiamo l'attenzione su quali parametri utilizzare nei primi studi, ovvero quelli volti a capire con quali parametri sia meglio eseguire le nostre query di ricerca.

Le variabili da definire sono 3:

- **Raggio di ricerca**: il parametro che definisce il raggio, in metri, della circonferenza all'interno della quale l'API Google (*Find Place*) andrà a cercare i P.O.I. in ogni query che facciamo.

Per questo parametro fissiamo 4 valori con cui effettuare gli studi: *100 metri, 200 metri, 500 metri e 1000 metri*.

- **Punti casuali generati**: il parametro che definisce il numero di punti generati casualmente all'interno della circonferenza di distorsione della nostra posizione.

Ricordiamo che a partire dai punti casuali si generano poi i risultati studiati come provenienti da una posizione distorta.

I valori scelti per questa variabile sono 5: *3, 5, 10, 15 e 20*.

---

<sup>1</sup><https://it.wikipedia.org/wiki/Napoli>

<sup>2</sup><https://it.wikipedia.org/wiki/Bologna>

<sup>3</sup><https://it.wikipedia.org/wiki/Aradeo>

<sup>4</sup><https://it.wikipedia.org/wiki/Lopagno>

- **Distorsione:** il parametro che definisce quanto vogliamo, in metri, distorcere la nostra posizione reale prima di effettuare una query.

Essendo questa variabile una delle più rilevanti ai fini del nostro studio i valori scelti per essa sono inizialmente 7: *50 metri, 100 metri, 200 metri, 300 metri, 500 metri, 700 metri, 1000 metri.*

Successivamente, una volta definiti i due parametri di cui sopra, eseguiamo degli studi più approfonditi al variare della distorsione da 0 a 2000 metri con un passo di 50 metri.

Analizziamo ora il comportamento delle query generate al variare di questi parametri. L'idea è quella di definire, alla fine, i parametri più opportuni con cui valutare e quindi trarre delle conclusioni riguardo al Trade-Off che vogliamo studiare.

## 4.2 Analisi dei risultati

### 4.2.1 Il raggio di ricerca

Decidiamo inizialmente di valutare la varianza del **raggio di ricerca**.

Come accennato prima, esso definisce il raggio, in metri, della circonferenza all'interno della quale andiamo a ricercare i P.O.I. in ogni normale query di ricerca.

Dunque, ad esempio, dati dei P.O.I. che si trovano a 300 metri dalle coordinate in cui eseguo la query, essi verranno rilevati, e quindi aggiunti nei risultati, soltanto se il raggio di ricerca impostato sarà maggiore o uguale a 300.

Nel nostro specifico caso, quindi, essi verranno trovati esclusivamente definendo il raggio di ricerca a 500 oppure a 1000 metri. Negli altri casi essi non verranno nemmeno individuati.

Esaminiamo ora dei grafici che ci mostrano il numero di *P.O.I. utili* rilevati al variare della distorsione della posizione in metri, il tutto analizzato per i nostri 4 differenti raggi di ricerca.

Importante ricordare che con **P.O.I. utili** vogliamo definire non semplicemente i risultati ottenuti dalla query distorta; ma, invece, i risultati ottenuti sia dalla query distorta ma

che siano anche inclusi nei risultati ottenuti a partire dalla query eseguita con posizione esatta, prima di distorcere la stessa.

In questa maniera valutiamo esclusivamente quelli che sarebbero i risultati effettivamente *utili* ai fini di una vera ricerca di P.O.I..

Altrettanto rilevante ai fini di questa ricerca è sottolineare come, in ognuno dei casi studiati, la differenza tra i P.O.I. ottenuti in totale nel calcolo con posizione distorta e quelli, tra questi, utili è prossima allo zero e quindi irrilevante.

Tale risultato è sicuramente da attribuire ad un "filtraggio" naturale dei P.O.I. inutili che avviene nella fase del calcolo dell'intersezione dei risultati ottenuti a partire dalle posizioni casuali generate.

Vediamo ora i grafici appena introdotti.



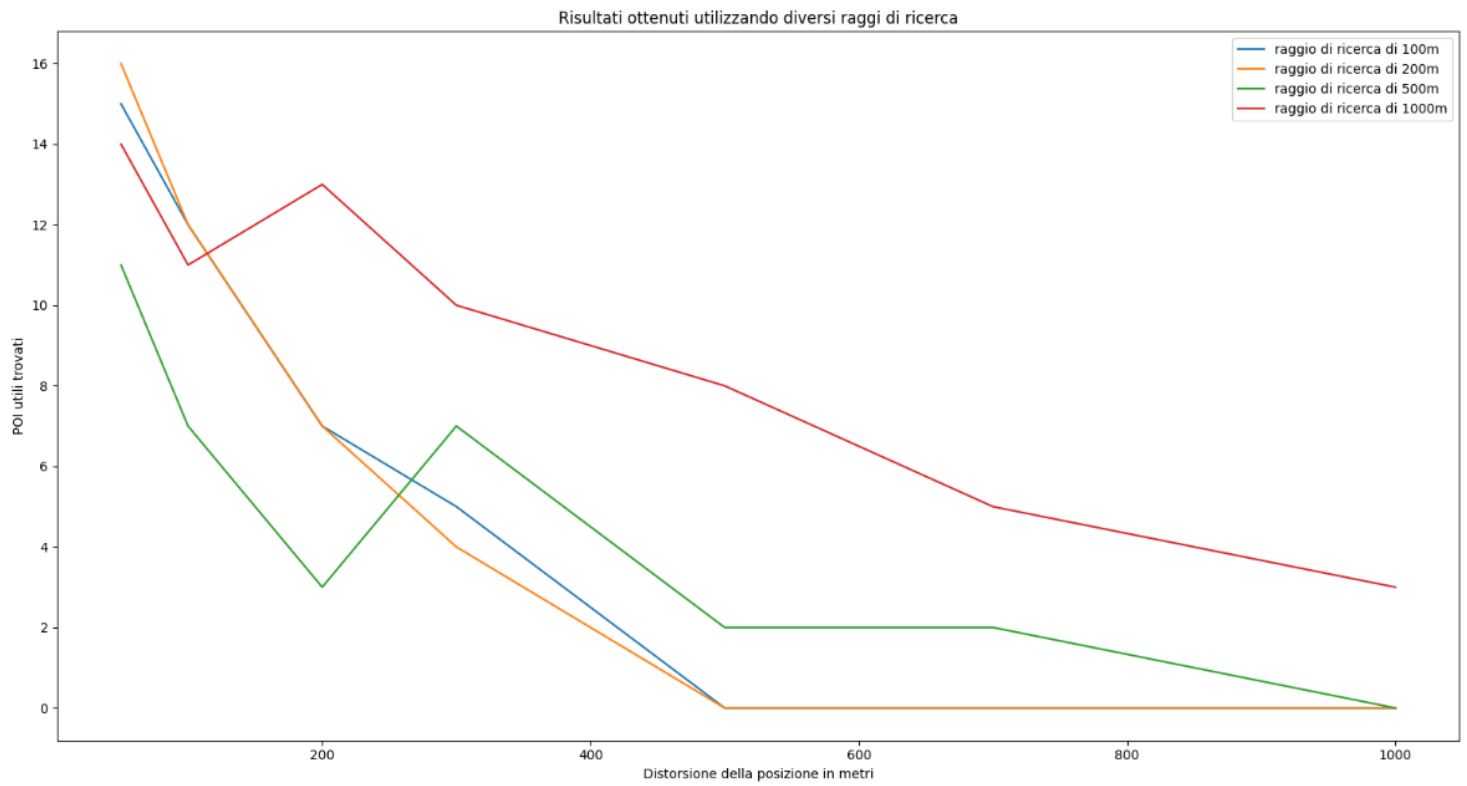


Figura 4.1: P.O.I. ottenuti per differenti raggi di ricerca al variare della distorsione, Napoli.

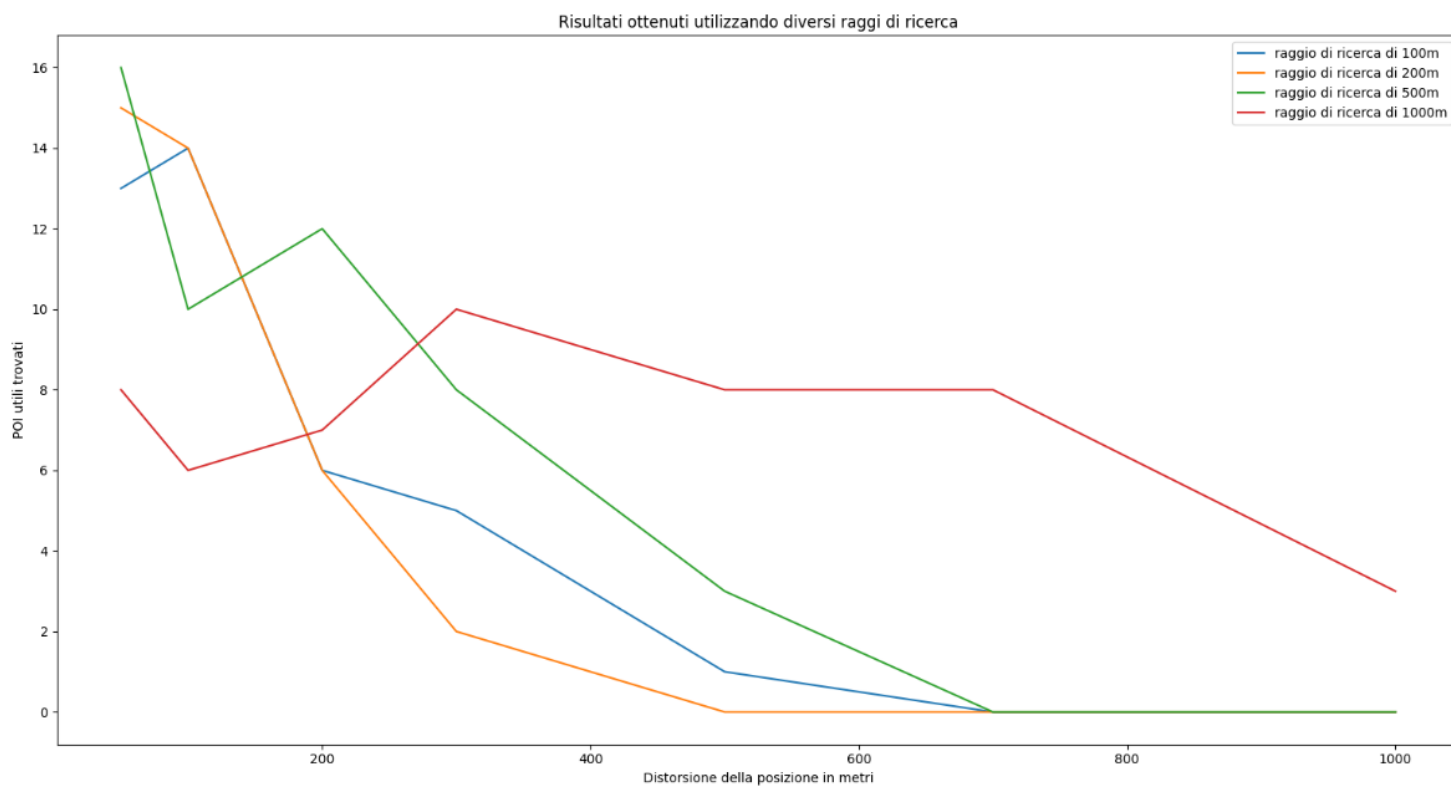


Figura 4.2: P.O.I. ottenuti per differenti raggi di ricerca al variare della distorsione, Bologna.

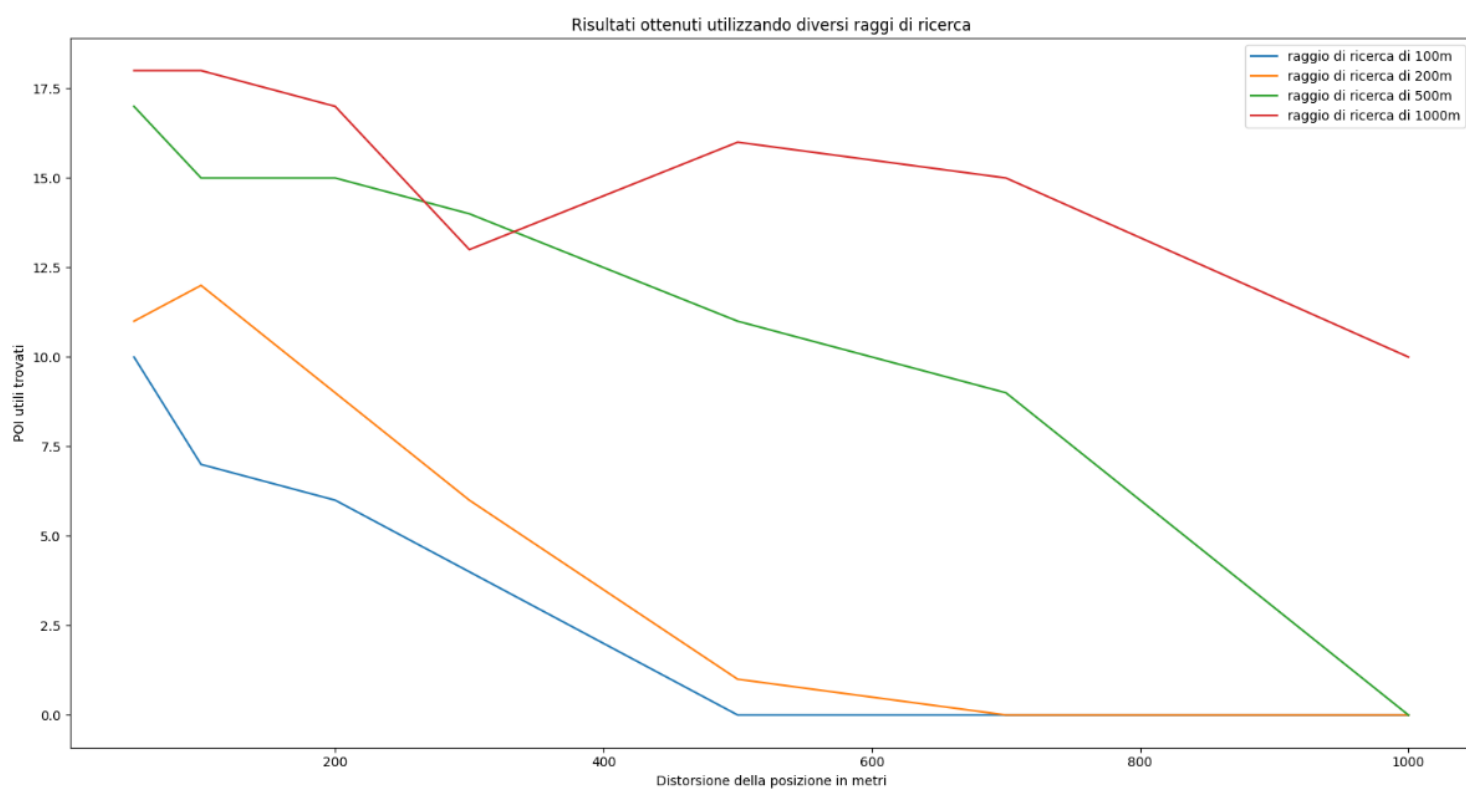


Figura 4.3: P.O.I. ottenuti per differenti raggi di ricerca al variare della distorsione, Aradeo.

La prima cosa che possiamo notare guardando tutti e 3 i grafici è come il numero di P.O.I. utili trovati diminuisca drasticamente con l'aumentare della distorsione della posizione; un qualcosa che ci aspettavamo ma su cui indagheremo nel dettaglio in seguito.

Spostiamo ora l'attenzione su alcune differenze tra i grafici delle diverse città. Notiamo in particolare come, per le città di Napoli 4.1 e Aradeo 4.3 i grafici presentano degli andamenti comunque relativamente costanti e uniformi. E' infatti evidente, in questi due casi, come per ogni raggio di ricerca usato il picco di P.O.I. utili si abbia con i valori più bassi della variabile di distorsione.

Il numero di P.O.I. scende poi gradualmente e con irrilevanti, piccole eccezioni. Risulta inoltre importante notare come i risultati migliori, in termini di numero di P.O.I. utili trovati, siano sempre definiti per il raggio di ricerca più grande in cui eseguiamo i nostri esperimenti: 1000 metri.

Per quanto riguarda, invece, il grafico ottenuto attraverso l'analisi nella città di Bologna 4.2 i risultati sono leggermente differenti, o meglio, presentano delle peculiarità. Possiamo notare come, in questo caso, i risultati ottenuti definendo il raggio di ricerca a 1000 metri siano alquanto bizzarri nella parte iniziale del grafico, ovvero quella con bassi valori di distorsione. Qui infatti, al contrario di quanto visto con le città di Aradeo e Napoli, il maggior numero di P.O.I. utili trovati lo si ha con i raggi di ricerca più bassi, quindi 100, 200 e 500 metri.

All'aumentare della distorsione, tuttavia, il grafico si conforma alle aspettative: comportandosi esattamente come i due grafici analizzati in precedenza e quindi con i migliori risultati ottenuti con raggio di ricerca di 1000 metri.

Non siamo, ad ora, riusciti scientificamente a spiegarci la peculiarità caratterizzante la parte iniziale di questo grafico.

Riteniamo, tuttavia, che si tratti di semplici coincidenze nate a partire dall'altissima concentrazione di P.O.I. nella zona in cui è stata svolta la ricerca e dalla vasta dimensione della zona stessa, corrispondente, nel pratico, all'intero centro di Bologna (più di 4km quadrati).

Un'ulteriore valutazione di questi grafici va fatta ponendo attenzione alle differenze presenti tra i risultati ottenuti nelle grandi città (Bologna, Napoli) e quelli ottenuti nella piccola cittadina studiata (Aradeo).

Possiamo infatti notare come il numero di P.O.I. utili assoluto al variare della distorsione risulti spesso maggiore nel grafico dei risultati di Aradeo. Un risultato apparentemente bizzarro, dal quale potrebbe sorgere spontanea una domanda: il numero di P.O.I. utili trovati non dovrebbe essere sempre minore in una piccola cittadina rispetto che in una grande città?

Inizialmente si potrebbe pensare di sì ma la risposta è invece negativa.

Per giustificarla dobbiamo anzitutto fare un passo indietro all'API utilizzata (Find Place) e ad una delle sue poche limitazioni: il numero massimo di P.O.I. restituiti dall'API ad una query è di soltanto 20 risultati.

Consideriamo, inoltre, come la minore densità di P.O.I. presente in una piccola cittadina rispetto che in una grande città possa essere controintuitivamente d'aiuto nella spiegazione di questi risultati.

Una minore concentrazione di P.O.I. significa, infatti, l'ottenimento di molti meno risultati 'inutili' nell'esecuzione della query distorta. In una grande città invece, nella ricerca con posizione distorta, otterremo molti più P.O.I. che potrebbero interferire tra l'oro (data l'operazione di intersezione eseguita per il calcolo dei risultati finali come spiegato in 3.4.1).

Tale interferenza potrebbe essere maggiormente problematica nel momento in cui le posizioni casuali generate si trovassero particolarmente distanti dalla posizione esatta di partenza e tra di loro.

Date queste due osservazioni riusciamo dunque a comprendere meglio tale risultato inizialmente inatteso.

Concludiamo i risultati di quest'analisi sottolineandone quanto ottenuto in termini di scelta dei parametri: il raggio di ricerca migliore da utilizzare è certamente quello di 1000 metri.

Esso, infatti, ottimizza al meglio la qualità del servizio ottenuto all'aumentare della di-

storsione della posizione di partenza e non inficia, inoltre, nell'efficienza in termini di prestazioni hardware; il lavoro di ricerca intorno all'area è infatti svolto lato server del servizio.

### 4.2.2 Il numero di punti generati casualmente

Secondo parametro molto importante da analizzare è certamente la varianza del numero di punti generati casualmente.

Ricordiamo in breve che i **punti generati casualmente** corrispondono alle nuove posizioni (coppie di coordinate di latitudine e longitudine) generate all'interno del cerchio corrispondente alla distorsione della posizione che stiamo applicando.

Dunque, a partire da quelle coordinate, come già spiegato in precedenza, vengono prima calcolati i P.O.I. per ognuno dei punti e, successivamente, l'intersezione dei risultati che corrisponderà quindi ai P.O.I. finali trovati distorcendo la posizione.

Dati i risultati precedenti, per lo sviluppo dei seguenti grafici il parametro del raggio di ricerca è fissato al valore ottimale stabilito: 1000 metri.

Le seguenti rappresentazioni grafiche sono quindi i risultati ottenuti in termini di P.O.I. rilevanti trovati al variare dei punti generati per il calcolo della distorsione della posizione.

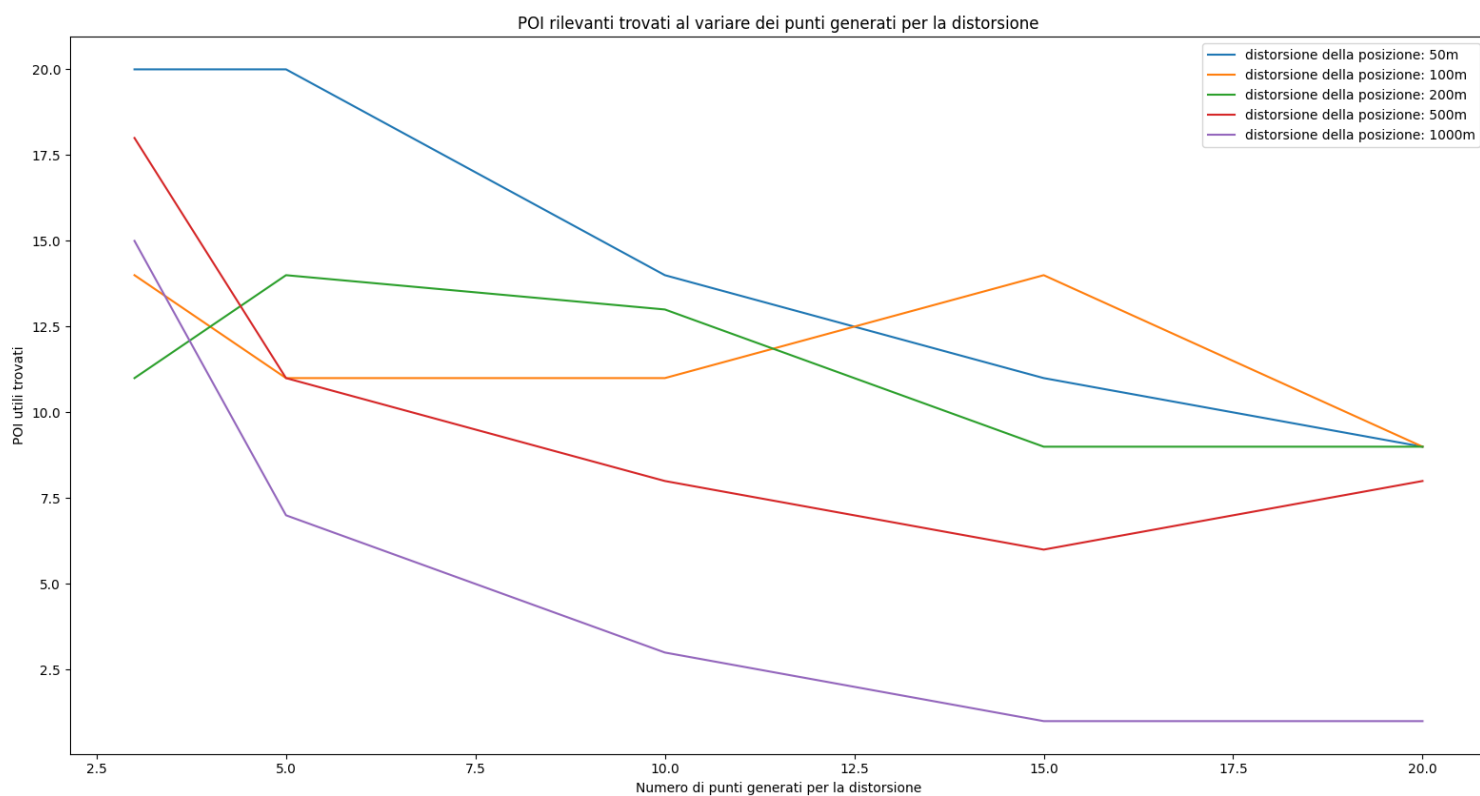


Figura 4.4: P.O.I ottenuti al variare dei punti generati casualmente, Napoli

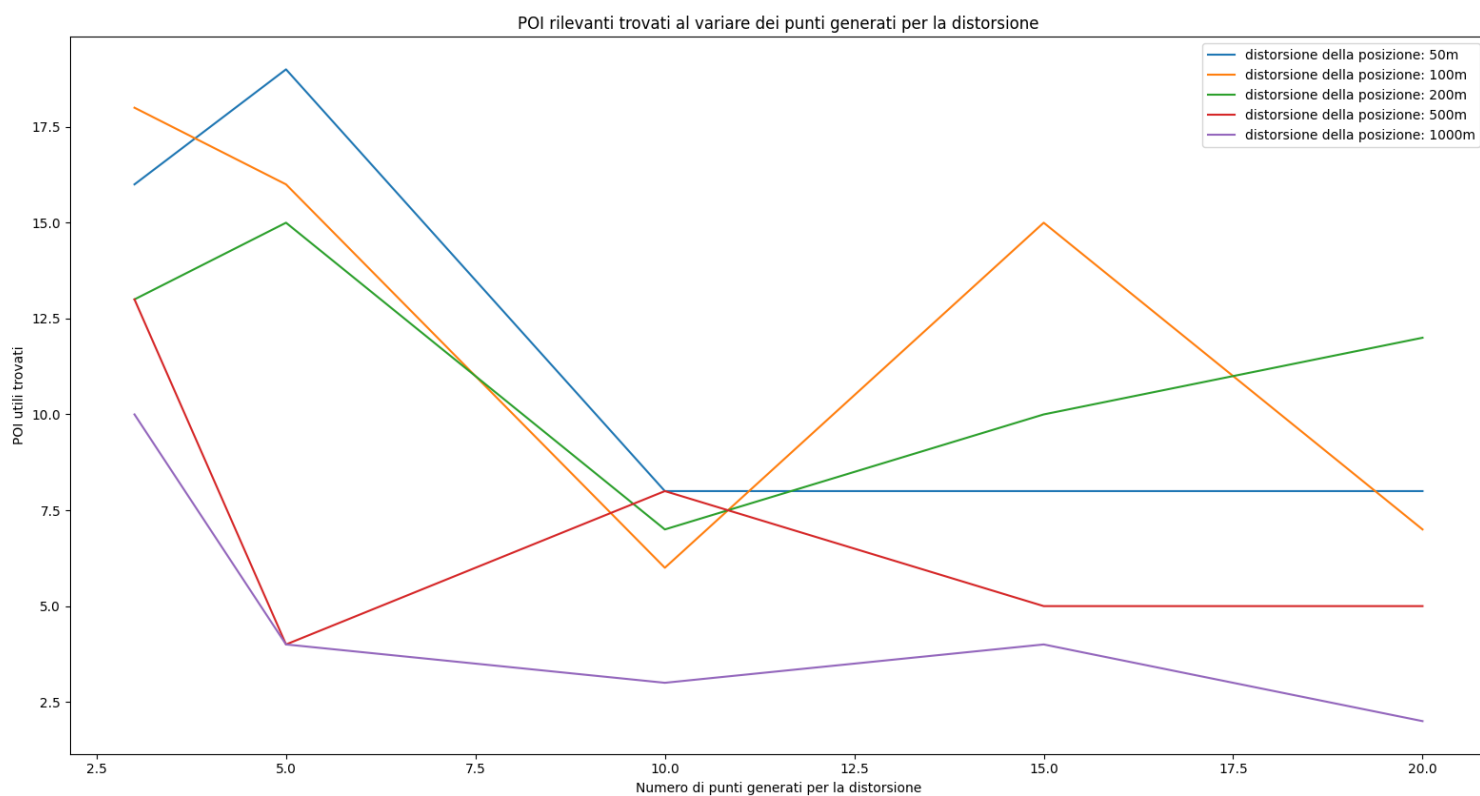


Figura 4.5: P.O.I. ottenuti al variare dei punti generati casualmente, Bologna



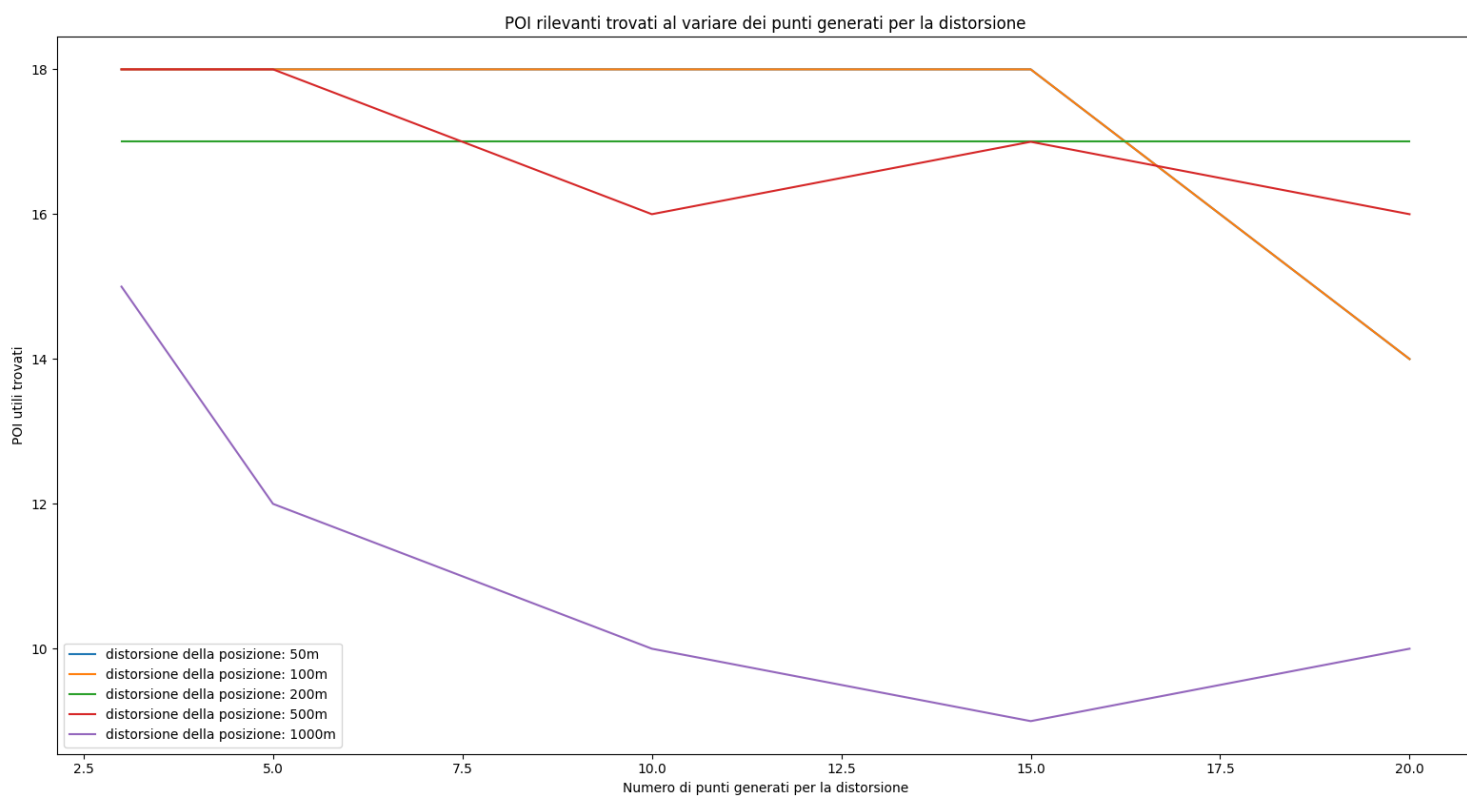


Figura 4.6: P.O.I. ottenuti al variare dei punti generati casualmente, Aradeo

I risultati deducibili da questi grafici, rispetto ai precedenti, presentano delle sorprese inaspettate.

Quello che possiamo immediatamente notare osservandoli tutti e 3 appare a prima vista controintuitivo: generare un numero maggiore di posizioni casuali per il calcolo di P.O.I. con posizione distorta **NON** migliora la qualità dei risultati ottenuti, anzi, la peggiora visibilmente.

Nell'analisi precedente notiamo, inoltre, come sia i risultati ottenuti per la città di Napoli 4.4 che quelli ottenuti per la città di Aradeo 4.6 risultino particolarmente uniformi.

In questi è infatti evidente una costante discesa del numero di P.O.I. utili trovati all'aumentare del numero di punti generato per il calcolo della distorsione. Banale, ma pur sempre importante, osservare come i risultati migliori in termini di P.O.I. utili rilevati si abbiano per distorsioni della posizione più basse.

Per quanto riguarda invece i risultati ottenuti per la città di Bologna 4.5, questi risultano ancora una volta discostarsi leggermente da quelli ricavati dalle altre due illustrazioni grafiche.

Notiamo infatti come, per i valori di distorsione più bassi (quindi 50, 100, e 200 metri), ci sia un crollo drastico nei P.O.I. ottenuti all'aumentare di poche unità il numero di punti generati. Dopo il crollo in due dei tre casi notiamo un evidente miglioramento dei risultati all'ulteriore aumentare del numero di posizioni generate. Nel terzo caso, dopo il crollo, notiamo invece un andamento costante.

Con i parametri di distorsione della posizione più alti (500 e 1000 metri) abbiamo, invece, un comportamento estremamente più uniforme, senza cambi drastici nell'andamento del grafico.

Tali differenze nel comportamento del grafico potrebbero essere un qualcosa di interessante da analizzare in successivi sviluppi futuri di questo studio.

Un'ulteriore osservazione va fatta rapportando i risultati ottenuti nella piccola cittadina rispetto a quelli ottenuti nelle due grandi metropoli.

L'andamento dei P.O.I. utili al variare del numero di punti generati per la distorsione risulta estremamente uniforme, quasi costante in alcuni punti del grafico.

Questo è sicuramente causato dalla minore densità di P.O.I. presente in una cittadina che, come spiegato nell'analisi precedente, causa molte meno problematiche nei calcoli dei risultati ottenuti per la posizione distorta rispetto al caso in cui i P.O.I. sono densamente più presenti sul territorio.

Questo genera quindi un andamento del grafico molto più lineare come quello che possiamo vedere nel nostro caso 4.6.

Da tutti e 3 i grafici possiamo, ad ogni modo, trarre le conclusioni a cui eravamo interessati: il numero migliore di posizioni casuali da generare per il calcolo di P.O.I. distorto è la posizione esatta.

Concludiamo, dunque, che i risultati qualitativamente migliori vengono ottenuti generando circa **5 posizioni casuali**.

Tale scelta risulta infatti essere un buon compromesso tra una copertura abbastanza vasta della circonferenza in cui decidiamo di definire la nostra posizione distorta e la qualità dei risultati ottenuti.

### 4.2.3 I P.O.I. al variare della distorsione applicata

Giungiamo finalmente a poter toccare con mano questo Trade-Off di cui parliamo dall'inizio del nostro documento.

Fissiamo dunque i parametri ottimali ottenuti dagli studi nei due sotto-capitoli precedenti: **raggio di ricerca** di 1000 metri e **punti casuali** generati 5.

Osserviamo i grafici rappresentanti la variazione dei P.O.I. utili trovati al variare della distorsione della posizione iniziale.

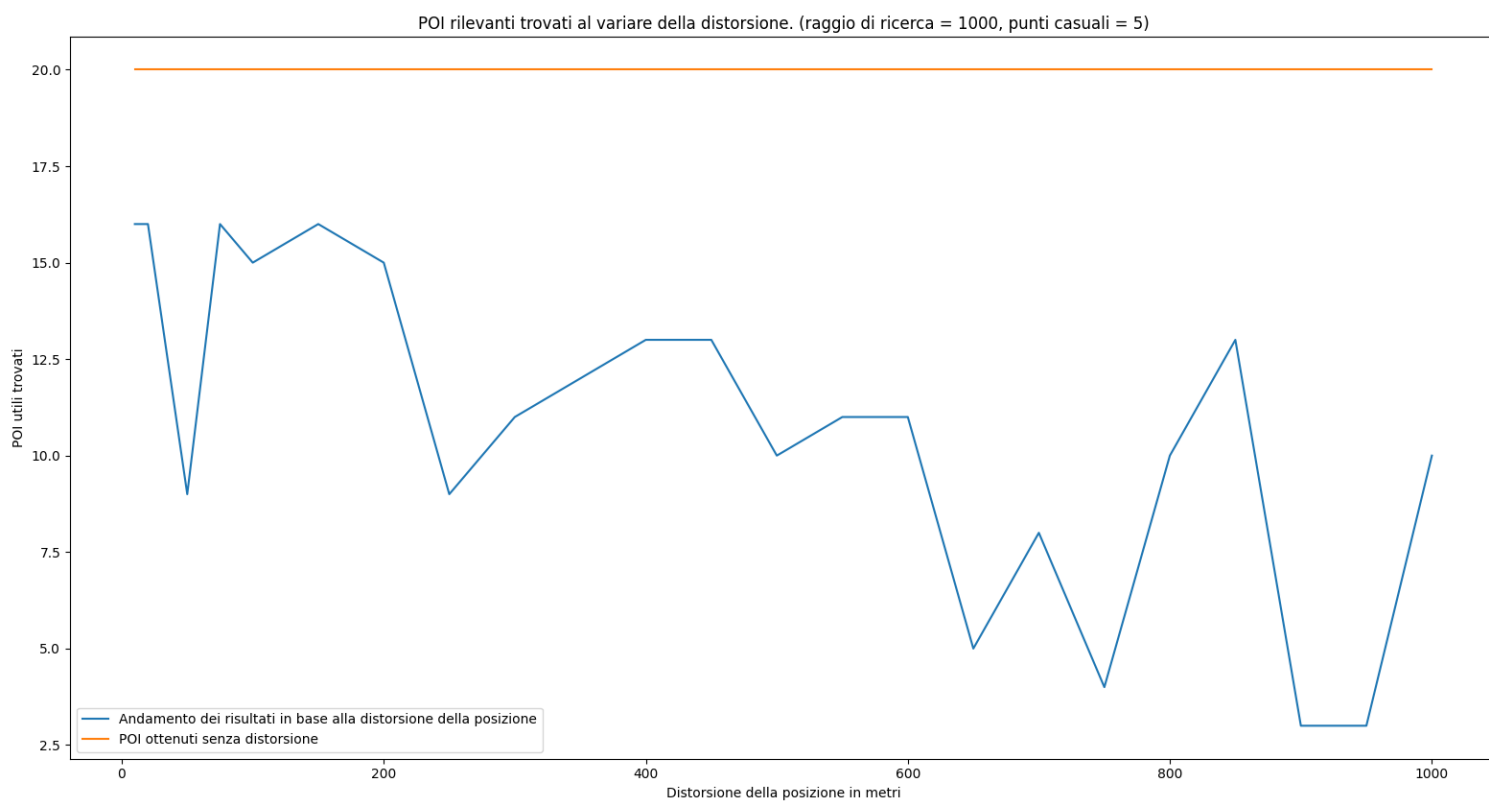


Figura 4.7: P.O.I ottenuti al variare dei punti generati casualmente, Napoli

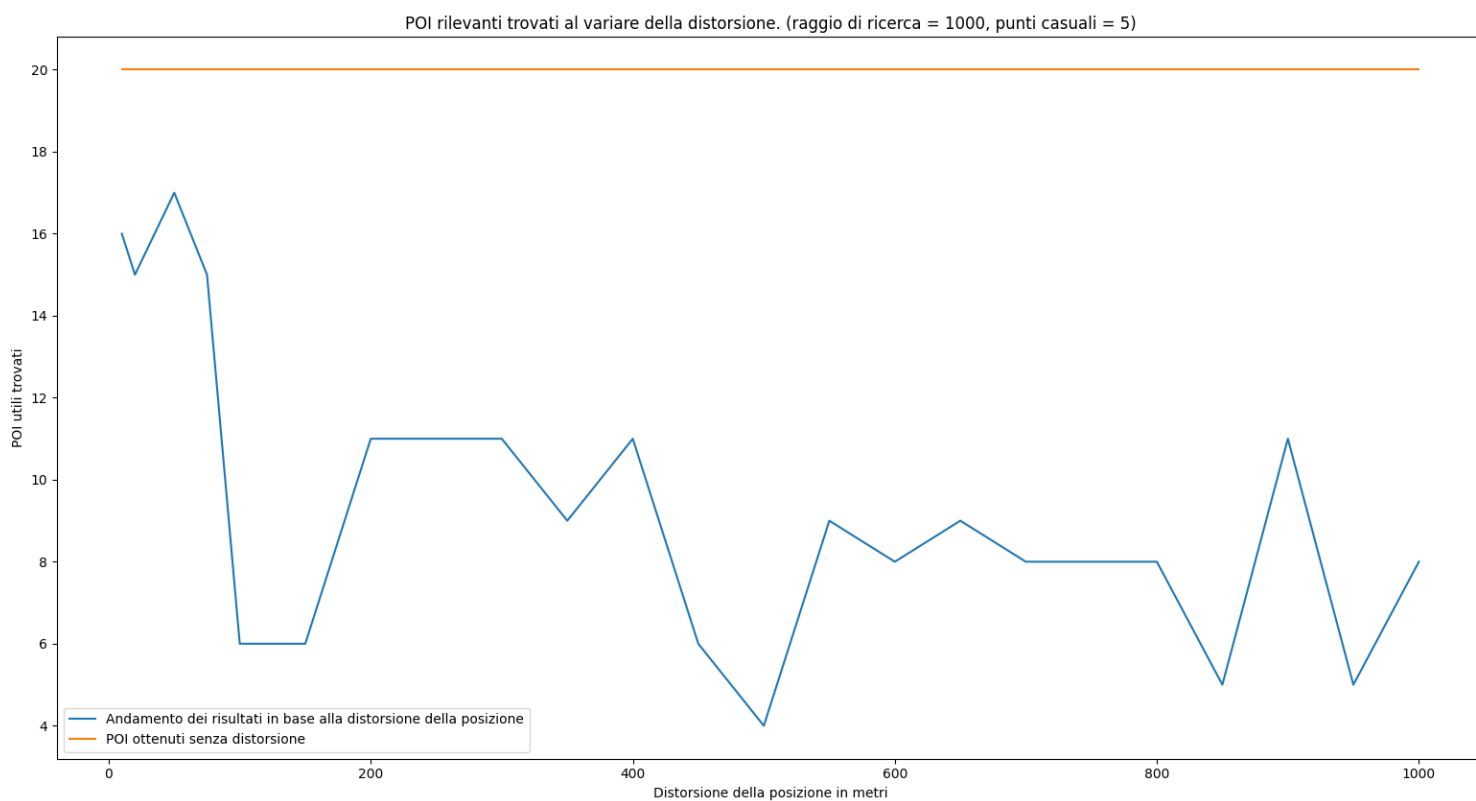


Figura 4.8: P.O.I ottenuti al variare dei punti generati casualmente, Bologna

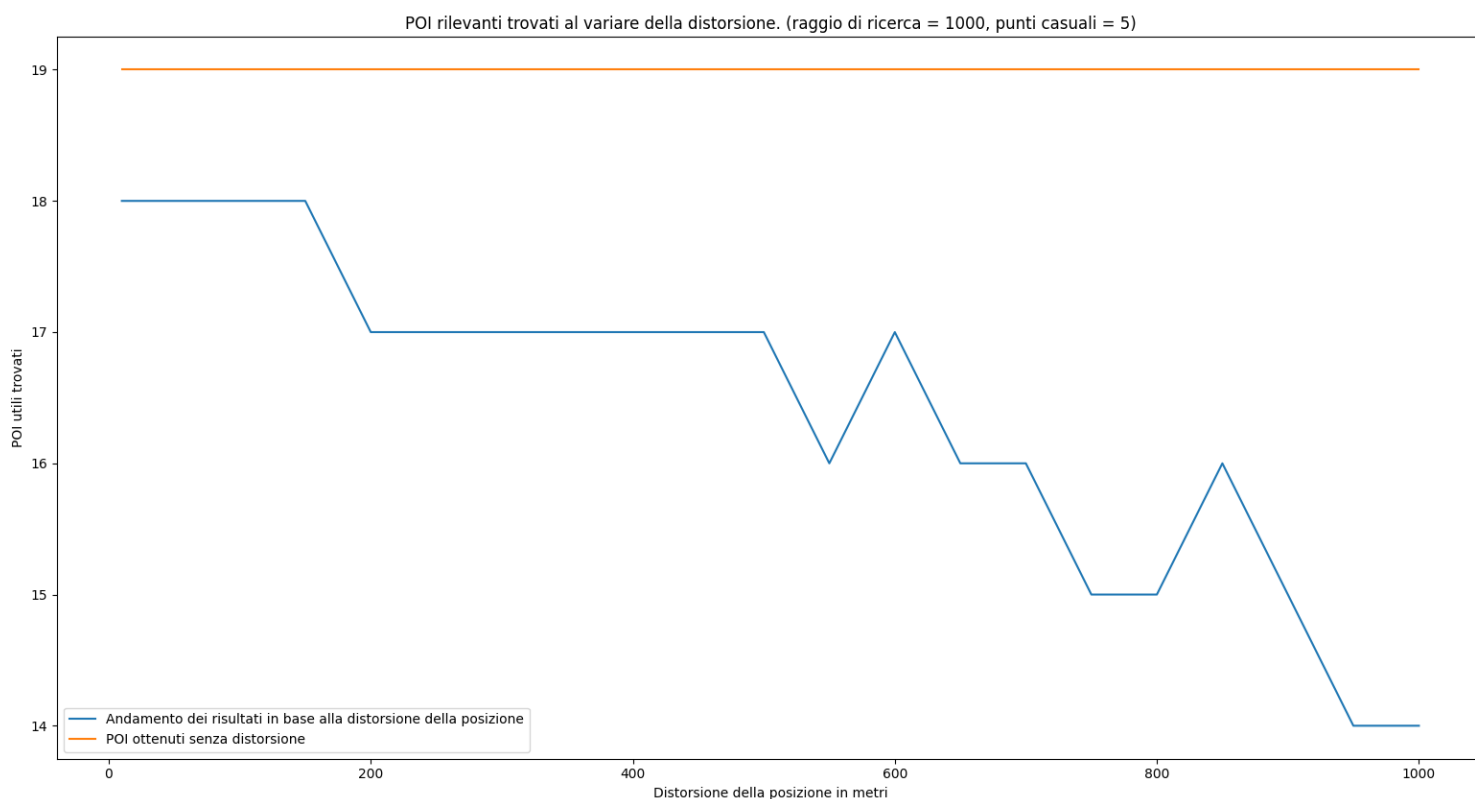


Figura 4.9: P.O.I. ottenuti al variare dei punti generati casualmente, Aradeo

I risultati riscontrati da questi tre grafici rispecchiano esattamente quelli attesi. Possiamo, infatti, in ognuna delle 3 città notare una chiara tendenza nella diminuzione dei P.O.I. trovati all'aumentare della distorsione della posizione. Questa tendenza, se pur ben definita, risulta leggermente ostacolata da qualche piccola curva in risalita nei grafici delle due metropoli (Napoli 4.7 e Bologna 4.8). Tale peculiarità è banalmente riconducibile all'altissima densità di P.O.I. presente in queste due città la quale, come già visto in precedenza, potrebbe generare interferenze (con i migliori risultati attendibili) nel momento in cui, ad esempio, i punti casuali generati si trovano relativamente distanti dalla posizione originale che si sta distorcendo e/o distanti tra di loro. In questi casi, infatti, due punti che siano ad esempio stati generati ai poli opposti, potrebbero ottenere risultati (in termini di P.O.I. ricavati dalla query) particolarmente differenti tra di loro e portare dunque ai saltuari crolli mostrati

nei grafici.

Considerando come possa succedere esattamente l'opposto di quanto appena considerato spieghiamo anche i relativi picchi presenti nei grafici.

Ulteriore riprova della nostra tesi è il fatto che tali picchi siano molto meno presenti nel grafico corrispondente alla città di Aradeo 4.9 e, nel caso in cui sono presenti, risultano marcatamente meno ripidi.

### **Aumentiamo ulteriormente la distorsione**

Dai grafici negli studi precedenti possiamo notare che anche con una distorsione di 1000 metri della nostra posizione il numero di P.O.I. utili ottenuti risulta essere comunque relativamente alto, spesso di molto sopra lo zero, garantendo così una qualità del servizio quantomeno non nulla.

Vogliamo spingerci oltre e studiare quindi come varierebbe tale qualità all'ulteriore aumentare della distorsione, arrivando ad un massimo di 2,5 chilometri.

Con una distorsione di questo tipo la nostra privacy sarebbe evidentemente ben preservata, ma ci chiediamo: sarà un Location Based Service in grado di fornirci una qualità del suo servizio quantomeno accettabile?

Per rispondere a questa domanda analizziamo i seguenti grafici.

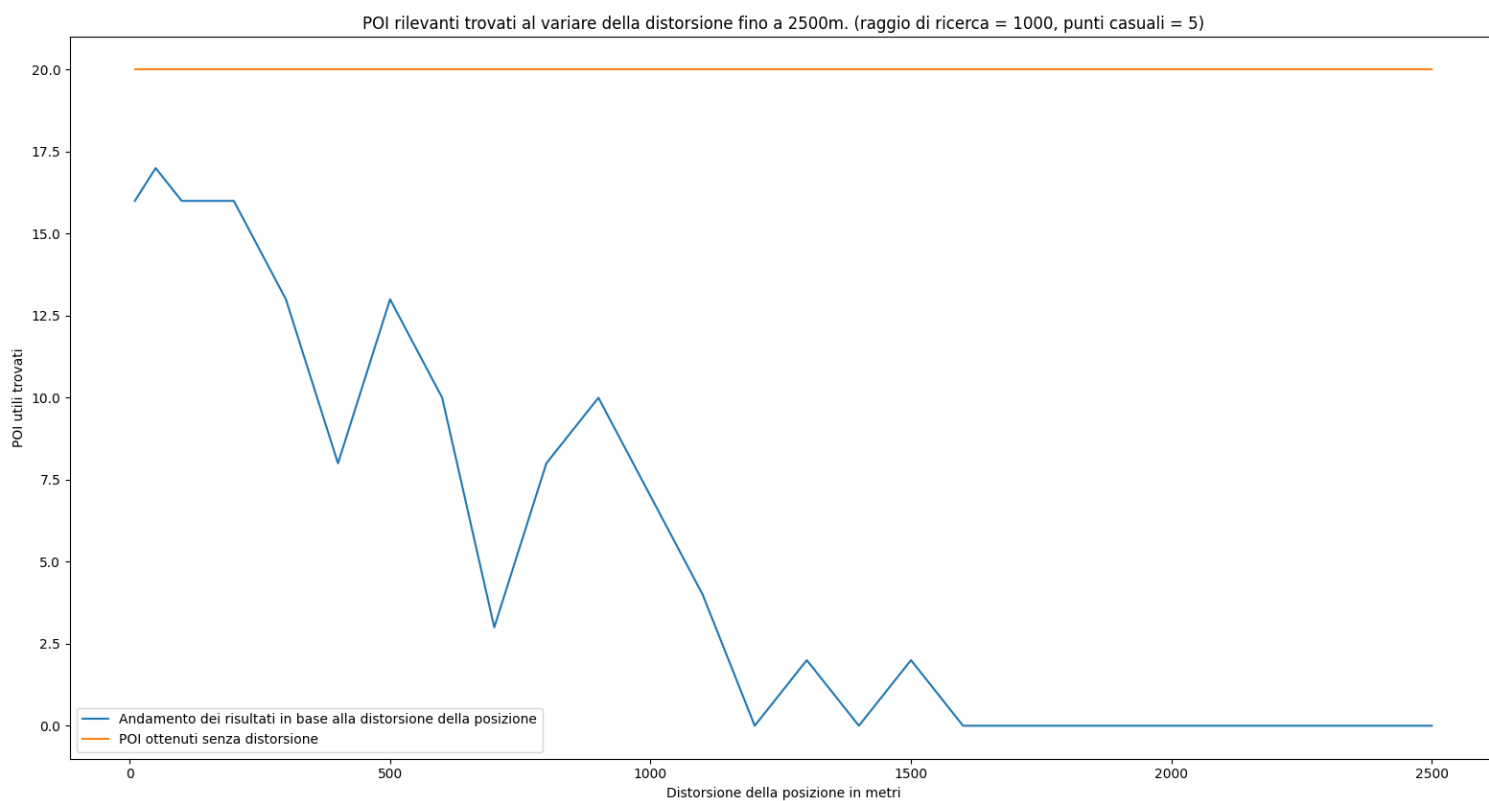


Figura 4.10: P.O.I ottenuti al variare dei punti generati casualmente, Napoli



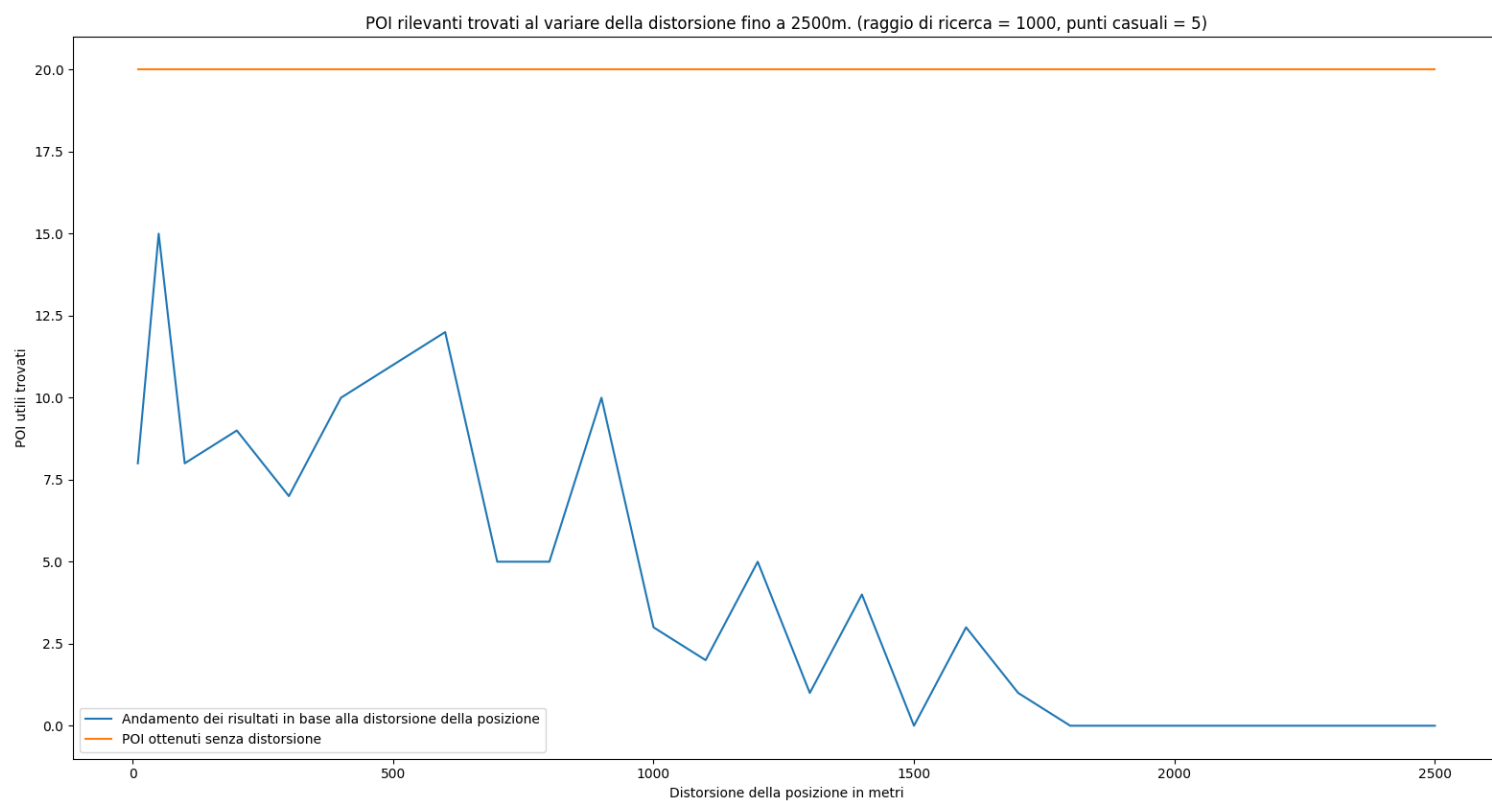


Figura 4.11: P.O.I ottenuti al variare dei punti generati casualmente, Bologna

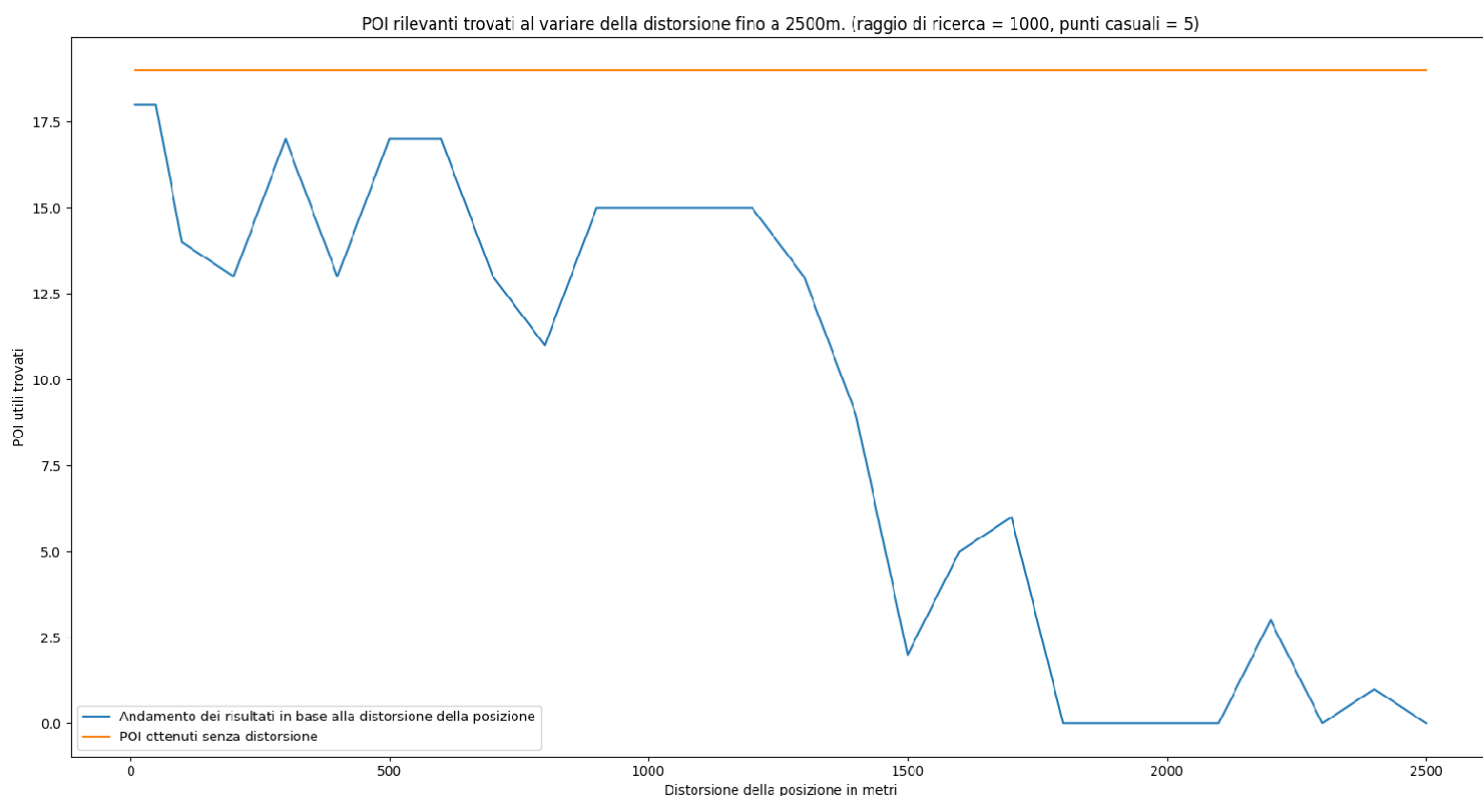


Figura 4.12: P.O.I. ottenuti al variare dei punti generati casualmente, Aradeo

Volendo rispondere alla domanda precedente possiamo dire definitivamente no. Notiamo infatti come intorno ad una distorsione di 1700 metri la qualità del servizio nelle due grandi metropolitane studiate (4.10 4.11) crolla praticamente a zero.

Il servizio risulta praticamente inutilizzabile con un ottenimento addirittura di 0 P.O.I. utili su un massimo di 20 desiderati.

Volendo ricercare invece una quantità di P.O.I. che possa definirsi minimamente costante (quindi un servizio utilizzabile), possiamo definirle un limite di distorsione, sempre parlando esclusivamente di Bologna e Napoli, intorno ai 700 metri.

Situazione differente si ha invece studiando l'andamento dei P.O.I. utili per la piccola cittadina di Aradeo. 4.12

Qui, infatti, la quantità di P.O.I. trovati rimane molto alta fino a circa 1400 metri di

distorsione, dopo i quali crolla arrivando a numeri molto bassi (a volte singole unità), ma comunque maggiori rispetto alla situazione analoga nelle grandi città studiate.

Possiamo dunque intuire come una maggiore distorsione della posizione iniziale sembri essere meno letale in termini di qualità del servizio in paesi piccoli piuttosto che nelle grandi città, una più dettagliata analisi a riguardo è fornita nel capitolo delle conclusioni 5.

## 4.3 L'efficienza in termini di utilizzo di risorse

Appurato che esistono dei parametri più opportuni per il calcolo della distorsione della posizione, ci chiediamo se tale distorsione abbia degli effetti negativi nel dispositivo in cui viene computata.

Ne analizziamo dunque l'impatto dal punti di vista della **banda di rete** utilizzata e dell'**impiego di risorse hardware**.

### 4.3.1 La banda di rete utilizzata

Nel momento in cui parliamo dell'utilizzo di un Location Based Service non possiamo di certo ignorare il fatto che, nella maggior parte dei casi, la connessione a internet utilizzata proverrà da un servizio di rete mobile, con l'ingresso del quale entrerà in gioco anche il concetto di consumo della banda di rete.

E' rilevante, infatti, sottolineare come uno degli aspetti negativi della distorsione di una posizione esatta (perlomeno per come è stata intesa in questo studio) sia certamente quello di dover mandare una serie di query a se stanti al servizio utilizzato.

Attraverso i nostri studi, abbiamo definito come il numero ottimale di posizioni casuali da generare sia 5: per ognuna di queste posizioni invieremo quindi una singola query.

La quantità di banda utilizzata per l'ottenimento di P.O.I. distorcendo la posizione iniziale sarà dunque 5 volte quella necessaria di norma.

### 4.3.2 L'impiego di risorse hardware

Altro aspetto negativo da valutare nella distorsione di una posizione è l'utilizzo di risorse hardware.

Il calcolo delle 5 posizioni casuali attorno a quella reale è un qualcosa che va sempre fatto lato Client (proprio per preservare la nostra privacy) ed è quindi qualcosa che nella maggior parte dei casi sarà interamente gestita dal nostro dispositivo.

Ulteriore richiesta di calcolo computazionale si avrà nella valutazione dei risultati ottenuti da ognuna delle 5 query generate.

La menzione degli impieghi di risorse sia in termini hardware che in termini di banda utilizzata è comunque, secondo noi, un qualcosa di puramente formale.

Infatti, volendo essere cinici, la maggior parte degli Smartphone, o comunque dei dispositivi di calcolo, in utilizzo nel 2023, probabilmente non subirebbe alcun genere di rallentamento, in termini di efficienza, nello svolgimento di questi calcoli.

Trattasi, infatti, di costi computazionali estremamente bassi tanto da essere **irrilevanti**.

Allo stesso modo, l'invio di esattamente 4 query di ricerca in più rispetto alla norma impiegherebbe un utilizzo di risorse di rete altrettanto irrilevante rispetto ai quantitativi di GigaBytes di banda messi a disposizione dagli attuali contratti delle reti mobili.

Per fare un esempio concreto, infatti, un quantitativo anche di 50 query di ricerca in più non sfiorerebbe minimamente l'ammontare di banda necessaria a guardare un video di pochi secondi su un qualunque social network.

# Capitolo 5

## Conclusioni

Analizziamo ora i risultati ottenuti.

Al fine di garantire la massima chiarezza, procederemo a suddividere le conclusioni in sottocapitoli appositamente concepiti per rispondere alle interrogazioni inizialmente sollevate.

### 5.1 Il valore di distorsione ottimale

Rispondiamo dunque alla principale domanda, oggetto dell'intero studio.

Esiste un valore di distorsione in questo Trade-Off che mi permetta di avere maggiore privacy nell'utilizzo di un Location Based Service pur mantenendo una buona qualità del servizio stesso?

Sì, esiste un valore ottimale di distorsione in questo Trade-Off.

Tale valore risulta differente nello studio svolto sulle due grandi città rispetto che a quello svolto nel piccolo paese di provincia.

#### 5.1.1 Il valore ottimale nelle grandi metropoli

Nelle due grandi metropoli studiate, quindi Napoli e Bologna, nonostante la rilevante differenza nella densità di popolazione, concludiamo che il valore di distorsione ottimale sia molto simile, variando tra i 600 e i 700 metri.

**In termini di Privacy**, volendo fare un esempio concreto, tale valore ci permetterebbe di essere identificati al più in un quartiere della città di Bologna; nel caso in cui ci dovessimo trovare di fronte alle *Due Torri*, ad esempio, il Location Based Service a cui inviamo la nostra posizione distorta potrebbe al massimo capire che siamo nel centro della città, ma non potrebbe in alcun modo dedurre informazioni più dettagliate.

**In termini di qualità del servizio**, invece, tale valore ci permetterebbe l'ottenimento di poco più della metà dei P.O.I. utili che otterremmo usando una posizione precisa.

Nonostante sia stato notevolmente messo alla prova, il servizio rimane dunque globalmente accettabile dal punto di vista qualitativo, ma trae notevoli benefici in termini di tutela della privacy.

### 5.1.2 Il valore ottimale nei piccoli paesi

Nel piccolo paesino di provincia studiato, invece, il valore di distorsione ottimale si spinge fino a 1300 metri.

**In termini di Privacy** il servizio effettua dunque dei passi in avanti quasi incredibili; basti pensare, che nel concreto, il Location Based Service a cui inviamo la nostra posizione distorta riuscirebbe al più a dedurre che ci troviamo nella città di Aradeo stessa o in qualche zona rurale limitrofa, ma non avrebbe ulteriori informazioni quali la zona o il quartiere dove ci troviamo.

**In termini di qualità del servizio**, invece, avremmo dei valori eccellenti, addirittura paragonabili a quella che si avrebbero utilizzando una posizione precisa.

In piccoli paesi, dunque, sembra che la distorsione della posizione abbia quasi esclusivamente dei vantaggi senza comportare quindi conseguenze negative rilevanti.

## 5.2 **Sviluppi futuri**

Gli sviluppi futuri di questo lavoro potrebbero essere molteplici e con diversi sbocchi interessanti.

I più immediati sarebbero sicuramente quelli dedicati all'approfondimento tecnico del comportamento di alcuni grafici visti in precedenza, come 4.2 e 4.5.

Nel primo caso, parliamo in particolare dell'approfondimento del comportamento secondo cui il numero di P.O.I. utili risulti, nella parte iniziale del grafico, ovvero per una distorsione poco rilevante, sensibilmente più basso utilizzando un raggio di ricerca più alto.

Tale risultato, a prima vista contro-intuitivo, cela sicuramente le sue ragioni nella gestione di un numero di P.O.I. elevatissimo che si ha nella zona di Bologna in cui sono stati eseguiti i nostri studi; sarebbe tuttavia interessante approfondirne le esatte motivazioni.

Nel secondo scenario, risulterebbe opportuno investigare più approfonditamente la manifestazione della situazione in cui un numero medio di punti casuali generati (per valutare la distorsione) comporta un numero inferiore di P.O.I. rispetto ai casi in cui vengono generati un numero minore o maggiore di punti casuali stessi.

Un ulteriore sviluppo di questa ricerca, andando oltre quanto precedentemente esaminato, potrebbe consistere nella creazione di un'App che consenta di applicare i nostri risultati nella vita quotidiana.

Sarebbe estremamente interessante sviluppare, ad esempio, un'applicazione mobile in grado di distorcere la nostra posizione a piacimento durante l'utilizzo di un servizio basato sulla localizzazione.

Immaginiamo un'applicazione di tale natura che consideri non solo le preferenze dell'utente in termini di privacy, ma anche variabili momentanee come lo stato della batteria, la disponibilità di larghezza di banda e altri parametri simili. In questo modo l'applicazione potrebbe decidere se e in che misura distorcere la posizione esatta in base a tali parametri valutati in tempo reale.

# Bibliografia

- [1] Regulation (eu) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504>. Online; Accessed: July 4, 2023.
- [2] <https://lbedogni.github.io/>.
- [3] Samar Muslah Albladi and George R. S. Weir. User characteristics that influence judgment of social engineering attacks in social networks. *Human-centric Computing and Information Sciences*, 2018.
- [4] S. Chakraborty, K. R. Raghavan, M. P. Johnson, and M. B. Srivastava. A framework for context-aware privacy of sensor data on mobile systems. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, HotMobile '13. Association for Computing Machinery, 2013.
- [5] S. Chitkara, N. Gothoskar, S. Harish, J. I. Hong, and Y. Agarwal. Does this app really need my location? context-aware privacy management for smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3), September 2017.
- [6] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David A. Wagner. Android permissions: user attention, comprehension, and behavior. In *Symposium On Usable Privacy and Security*, 2012.
- [7] Patrick Gage Kelley, Sunny Consolvo, Lorrie Faith Cranor, Jaeyeon Jung, Norman M. Sadeh, and David Wetherall. A conundrum of permissions: Installing



- applications on an android smartphone. In *Financial Cryptography Workshops*, 2012.
- [8] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkatasubramanian. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, 2006.
- [9] Pardis Emami Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujio Bauer, Lorrie Faith Cranor, and Norman Sadeh. Privacy expectations and preferences in an IoT world. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 399–412, Santa Clara, CA, July 2017. USENIX Association.
- [10] G. Simson and L. H. Richter. Usable security: History, themes, and challenges. *Synthesis Lectures on Information Security, Privacy, and Trust*, 5(2):1–124, 2014.
- [11] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, oct 2002.