

Classificazione dei tipi di parcheggio su app mobile

Alessio Luciani

Corso di Laurea in Informatica

A.A. 2019/2020



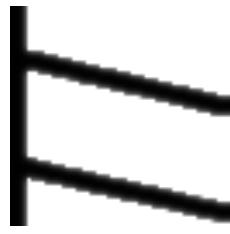
SAPIENZA
UNIVERSITÀ DI ROMA

Tutti i diritti relativi al presente materiale didattico ed al suo contenuto sono riservati a Sapienza e ai suoi autori (o docenti che lo hanno prodotto). È consentito l'uso personale dello stesso da parte dello studente a fini di studio. Ne è vietata nel modo più assoluto la diffusione, duplicazione, cessione, trasmissione, distribuzione a terzi o al pubblico pena le sanzioni applicabili per legge

Obiettivi del lavoro

Sviluppo di una serie di funzionalità integrate in un'applicazione mobile di ambito smart-parking mirate a:

- Creazione di un **modello machine learning** che sia in grado di **classificare** diversi **tipi di parcheggio** (a spina, a pettine, parallelo).
- Sfruttamento dell'informazione ottenuta per diversi scopi di **beneficio per l'utente**.



A spina



A pettine



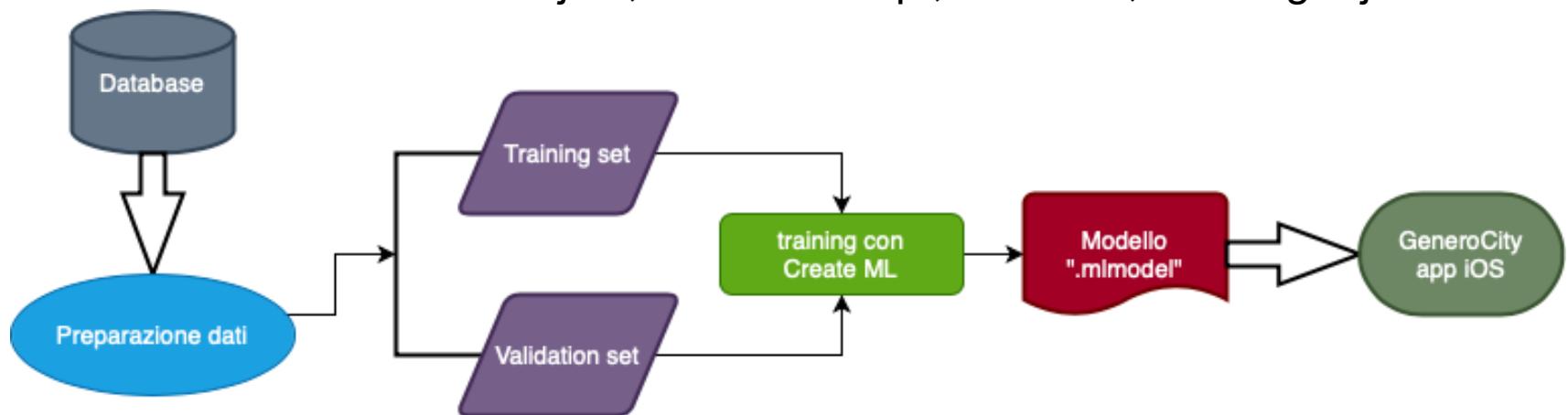
Parallelo

Roadmap

- **Raccolta dati** dai sensori di movimento dello smartphone durante il **parcheggio** del veicolo.
- Utilizzo, **all'interno dell'app**, del **modello ML** ottenuto.
- Rappresentazione dei **parcheggi** sulla **mappa** dell'app.

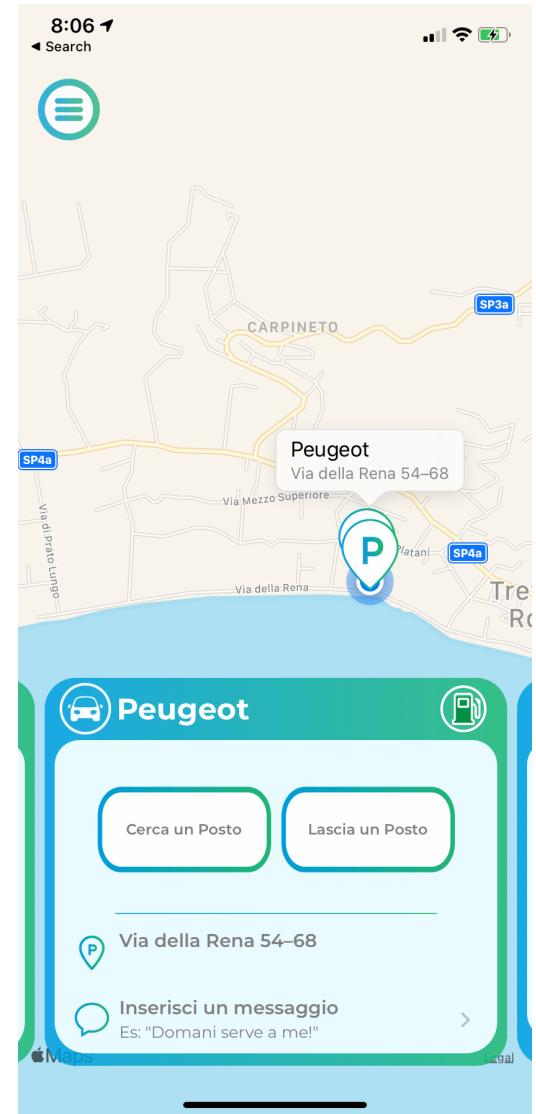
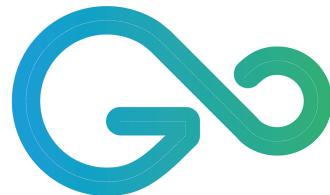
Attività del Percorso di Eccellenza

- Creazione di uno **script** in grado di **processare i dati** raccolti.
- Generazione della struttura di **input** per il **modello ML**.
- **Training e testing** del modello con *Create ML*.
- Sottomissione **paper** con possibilità di rielaborazione e i seguenti riscontri: 1 weak reject, 1 weak accept, 1 neutral, 1 strong reject.



Applicazione GeneroCity (iOS)

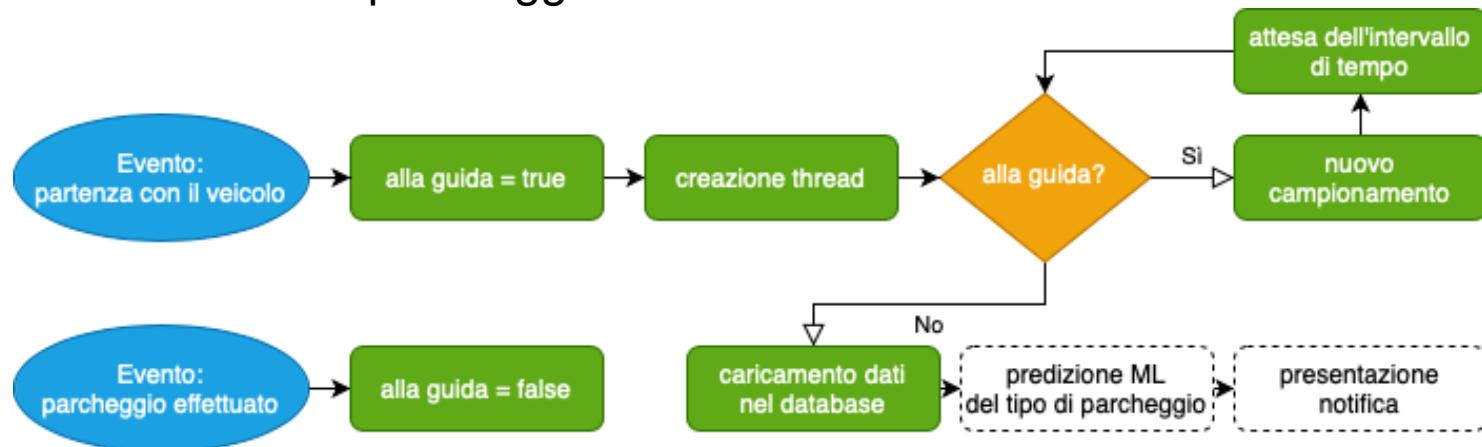
- Mette in **contatto** degli **utenti** che stanno **cercando un parcheggio** con altri che invece ne stanno **lasciando** uno.
- Permette **scambi** "generosi" e reciproci di **posti auto**.



Raccolta dati

Procedura generale

- **Riconoscimento automatico di partenza e parcheggio**, tramite eventi generati da GeneroCity.
- **Campionamento su thread dedicato**, con **cadenza fissa** di un decimo di secondo, durante il **tragitto** del veicolo.
- **Caricamento della registrazione nel database** in seguito all'avvenuto parcheggio.



Raccolta dati

Sensori utilizzati

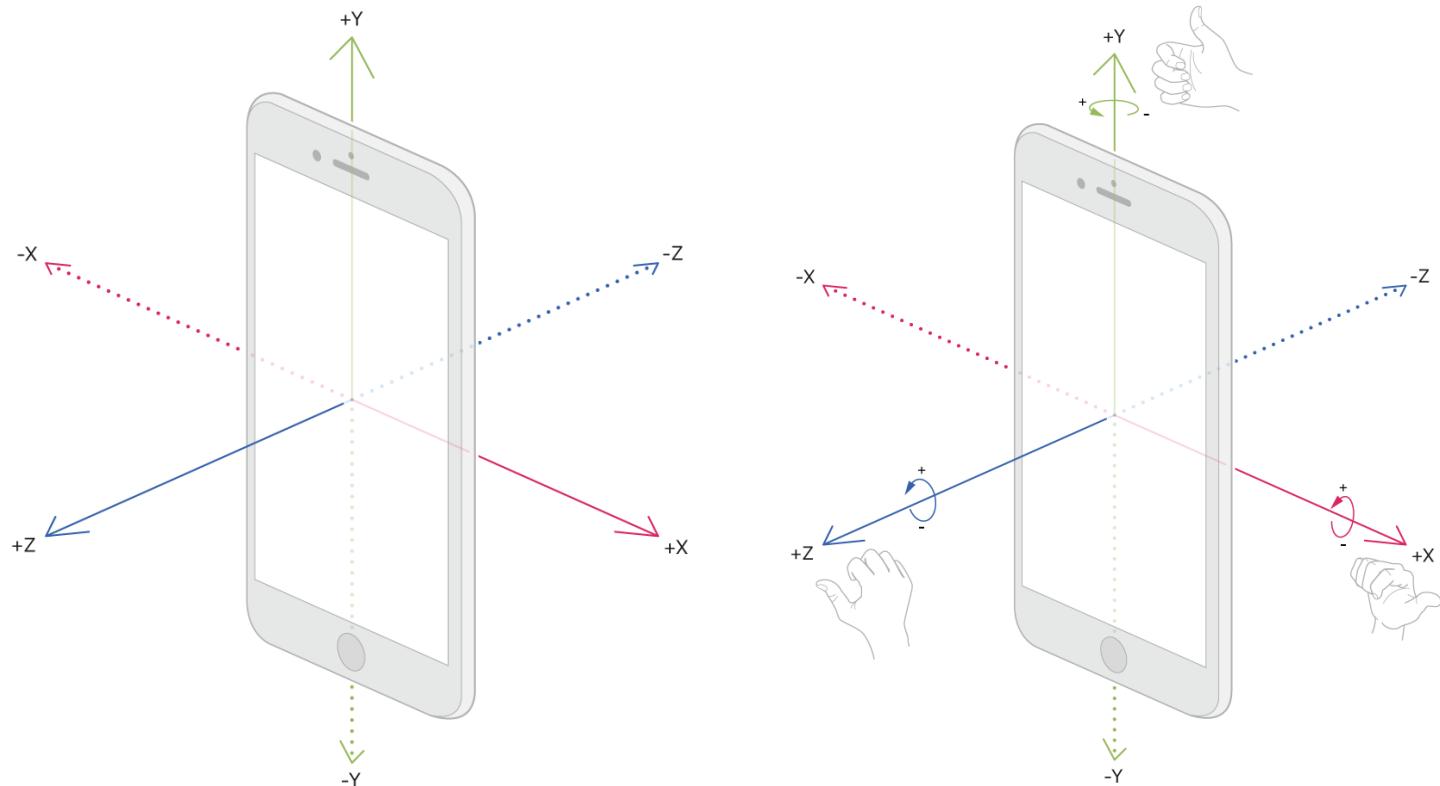
- Dati di tipo **time-series**, indicizzati da **timestamp** relativi all'istante iniziale.
- Ad ogni **timestamp** viene associato un **array** formato da: *heading*, *acceleration*, *rotationRate*, *speed*.
- Formazione di un **oggetto** in formato **JSON**, contenente tutti i campioni della **registrazione**.

```
{  
    "0.0": [264.93505859375, -0.040266513824463, 0.0010430663824081, 0.14339923858643,  
            -0.078436441719532, 0.24959509074688, -0.061101898550987, 12.005425037075],  
    "0.1": [264.44885253906, -0.052589416503906, -0.049757316708565, 0.018725872039795,  
            -0.14559917151928, -0.14420491456985, -0.1076086461544, 12.145743627005],  
    "0.2": [261.88244628906, 0.02913099527359, -0.11715778708458, 0.032332420349121,  
            -0.12509134411812, 0.29885098338127, -0.14259395003319, 12.145743627005],  
    "0.3": [260.48599243164, -0.0021500587463379, 3.640353679657e-5, 0.1856153011322,  
            -0.1112469509244, -0.098763234913349, -0.036537483334541, 12.145743627005],  
    "0.4": [258.86791992188, 0.088488698005676, -0.024656161665916, 0.058977127075195,  
            -0.18753668665886, 0.24890285730362, -0.19531415402889, 12.145743627005],  
    "0.5": [251.50372314453, -0.069232881069183, -0.050441384315491, 0.042986512184143,  
            -0.26215711236, 0.29828360676765, -0.025496302172542, 12.472938705969]  
}
```

Raccolta dati

Sensori utilizzati

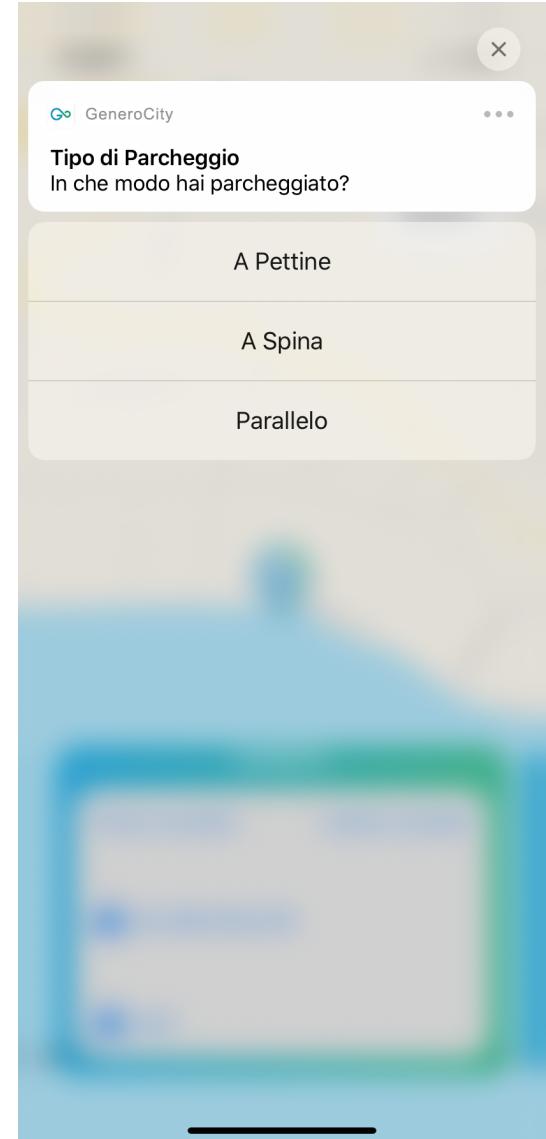
Disposizione degli assi dell'accelerometro e del giroscopio sullo smartphone.



Raccolta dati

Etichetta selezionata dall'utente

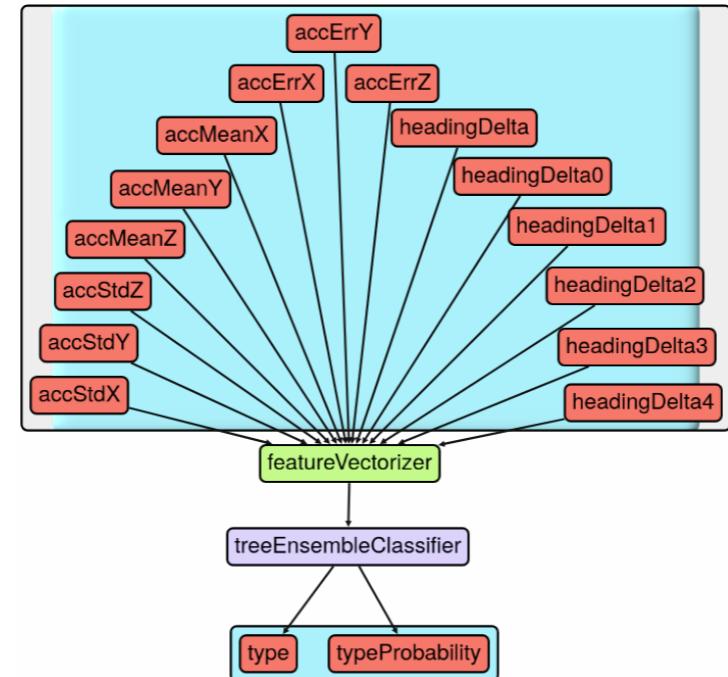
- Dopo un parcheggio, viene mostrata una **notifica** all'utente con le **opzioni del tipo**.
- L'**utente seleziona** il tipo corretto, in quanto **apprendimento supervisionato**.
- La **risposta** viene salvata come **etichetta** della registrazione del parcheggio nel database.



Struttura del modello

Due tipologie di modello classificatore utilizzate:

- Una **rete neurale convoluzionale** (richiede un **grande dataset**), utilizzando il *Motion Activity Classifier*.
- Un **tree ensemble** (richiede **estrazione esplicita di feature**), utilizzando il *Tabular Classifier*.



Classificatore tree ensemble

Utilizzo del modello nell'app

- **Classificazione del tipo di parcheggio in locale nell'app.**
- **Raccolta, preparazione dati e predizione ML in locale.**
- **Caricamento del tipo di parcheggio classificato nel database.**



Rappresentazione dei parcheggi sulla mappa

Idea

- **Mostrare sulla mappa** di GeneroCity i **parcheggi** che sono stati registrati in passato.
- Per ogni parcheggio **rappresentare: posizione, tipo, orientamento**.
- L'**utente** scopre **dove** sono presenti parcheggi e **informazioni utili** riguardanti essi.

Rappresentazione dei parcheggi sulla mappa

Visualizzazione della mappa

L'informazione sull'**orientamento** può essere rappresentata oppure no, dando forma a **due modalità di visualizzazione diverse**.

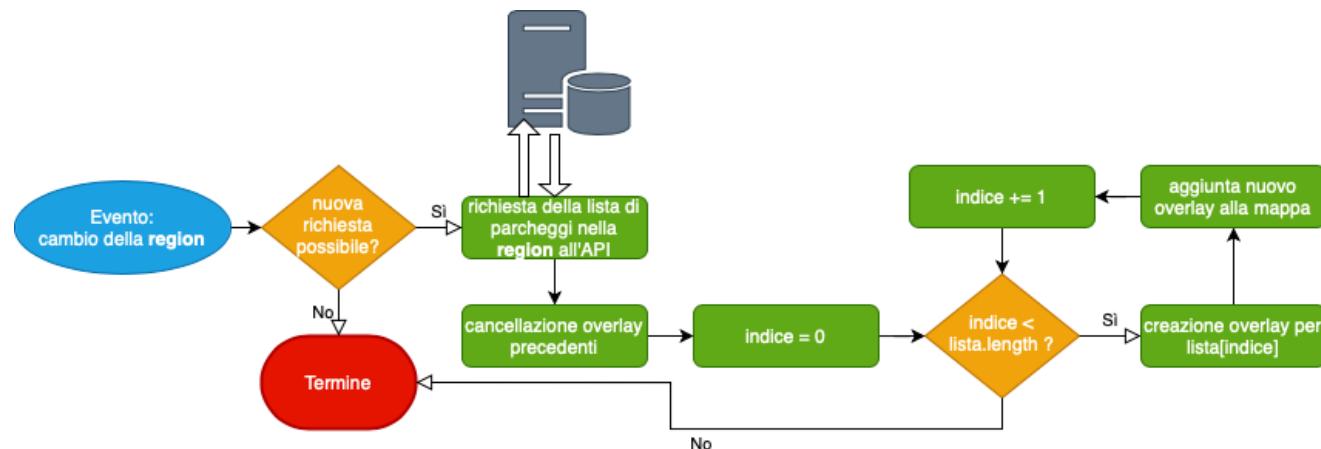


(giallo: parallelo, verde: a pettine, rosso: a spina)

Rappresentazione dei parcheggi sulla mappa

Caricamento degli overlay dei parcheggi

- **Scaricamento dal database** dei dati sui **parcheggi** nella **zona visualizzata**.
- **Ottimizzazione** dello **scheduling** di nuove **richieste** in base al **cambiamento** della **zona** visualizzata.
- **Creazione** di nuovi **overlay** da rappresentare sulla **mappa** utilizzando i dati dei parcheggi ottenuti.



Rappresentazione dei parcheggi sulla mappa

Scheduling delle richieste

- **Nuovo tentativo** ogni volta che la **zona** visualizzata **cambia**.
- **Controllo delle condizioni:**
 - **ultima richiesta effettuata recentemente**;
 - **zona visualizzata simile alla precedente**;
 - **zona visualizzata troppo grande**;
- **Interruzione della richiesta** se almeno una **condizione è soddisfatta**.

```
// Aborting when the parking overlays toggle is off,
// map zoom is not enough, the last query has been
// performed recently or the region has not drastically changed.
if !force &&
  (!env.showingParkingOverlays || mapView.region.span.latitudeDelta > MAX_REGION_DELTA
  || Date().timeIntervalSince1970 - self.lastQueryTimestamp < MIN_TIME_BETWEEN_QUERIES
  || (abs(mapView.region.center.latitude - lastQueryRegion.center.latitude) < lastQueryRegion.span.latitudeDelta/2
    && abs(mapView.region.center.longitude - lastQueryRegion.center.longitude) < lastQueryRegion.span.longitudeDelta/2
    && mapView.region.span.latitudeDelta < lastQueryRegion.span.latitudeDelta * 1.5)) {
  return
}
```

Sviluppi futuri

- **Rielaborazione paper** per nuova sottomissione.
- **Rilascio GeneroCity beta** per **velocizzare** la raccolta dei dati.
- **Stima accurata** e salvataggio dei **parcheggi reali** nel database.
- **Utilizzo** dell'informazione sul **tipo** di parcheggio nell'**algoritmo** di **matching**.

Grazie per l'attenzione!