

WHEELFLOW

Mobile and Social Sensing Systems Project

Benedetta Tessa, Francesca Pezzuti,
Franco Terranova, Pietro Tempesti

WHAT WE ARE GOING TO PRESENT

01

WHAT IS
WHEELFLOW?

02

ARCHITECTURE

03

DATA COLLECTION

04

DATA ANALYSIS

05

ANDROID APPLICATION

06

LANDMARK
SCORES

LET'S START!





01

WHAT IS WHEELFLOW?

THE ISSUE

For wheelchair users, accessibility is a crucial problem

They need to know about inaccessible locations before planning their trips

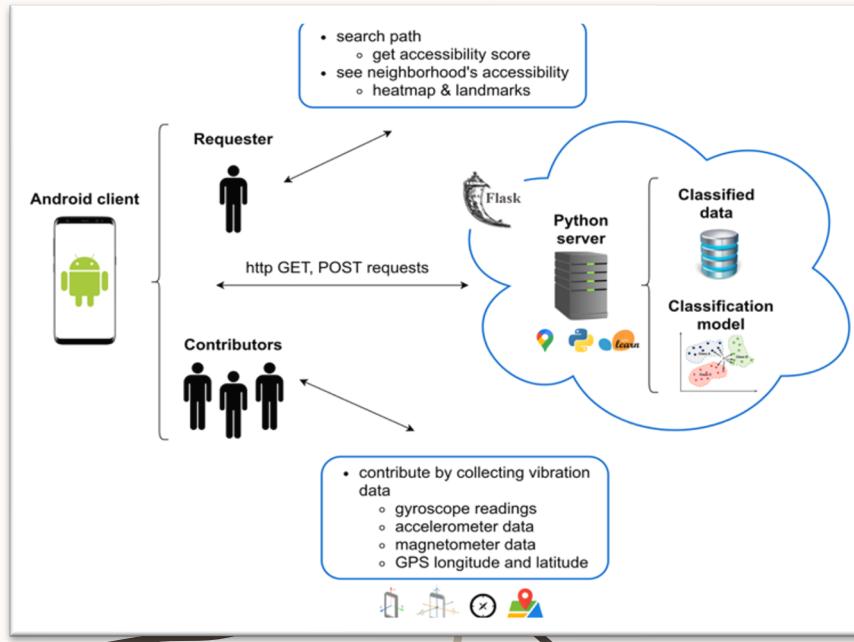
WheelFlow focuses on collecting and sharing geolocalized accessibility information about routes



THE ARCHITECTURE

HOW IT WORKS

WheelFlow is a crowd-sourced Android application where users can search for a path from a source to a destination and receive some accessibility information about the path.



DATABASE STRUCTURE

ID	LATITUDE	LONGITUDE	SCORE	THRESHOLD
1	44.085	10.0377	-4	7
2	43.7222	10.389	5	10
3	43.717	10.397	-3	12

EXAMPLE OF A REQUEST

Post: Send to /locations/update

```
{ "data": [{"timestamp": "2022/05/04-11:36:58.558",  
    "latitude": 43.716141,  
    "longitude": 10.3965,  
    "ACC_X": 1.6759412,  
    "ACC_Y": 1.2737153,  
    "ACC_Z": 7.7667904,  
    "GYR_X": 0.45448375,  
    "GYR_Y": 0.41172317,  
    "GYR_Z": 0.35430184,  
    "MAG_X": 13.6875,  
    "MAG_Y": -0.3125,  
    "MAG_Z": -47.625},  
    ...]}
```

Get: Request to /locations/inaccessible/scores

```
[{"latitude": 44.0495,  
    "longitude": 10.0588,  
    "score": -4,  
    "bound": 7}]
```



03

DATA COLLECTION

RECORD APPLICATION



HOW IT IS PERFORMED



Collect data through a traditional wheelchair
mostly in an assistant – propelled way

- GPS
- Timestamp
- Accelerometer (x,y,z axis)
- Gyroscope (x,y,z axis)
- Magnetometer (x,y,z axis)
- Label

Total of 2 millions records!





04

DATA ANALYSIS

PREPROCESSING PIPELINE

SENSOR DATA TRANSFORMATION

WINDOW CREATION

FEATURE EXTRACTION

FEATURE NORMALIZATION AND
SELECTION

REBALANCING



DATA TRANSFORMATION

RAW DATA FROM SMARTPHONE

3 FEATURES FOR EACH SENSOR

ONE FOR EACH AXIS OF THE SENSORS



SENSOR DATA AGGREGATION

1 FEATURE FOR EACH SENSOR

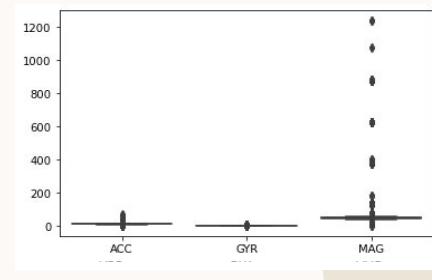
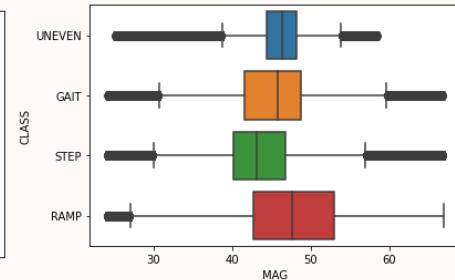
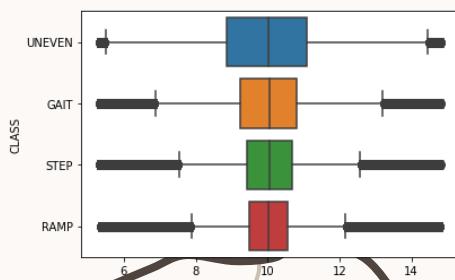
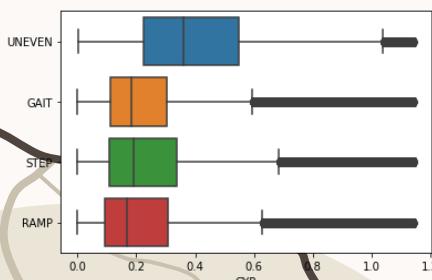
EACH FEATURE IS THE NORM OF A SENSOR'S COORDINATES → OUTLIERS REMOVED HERE



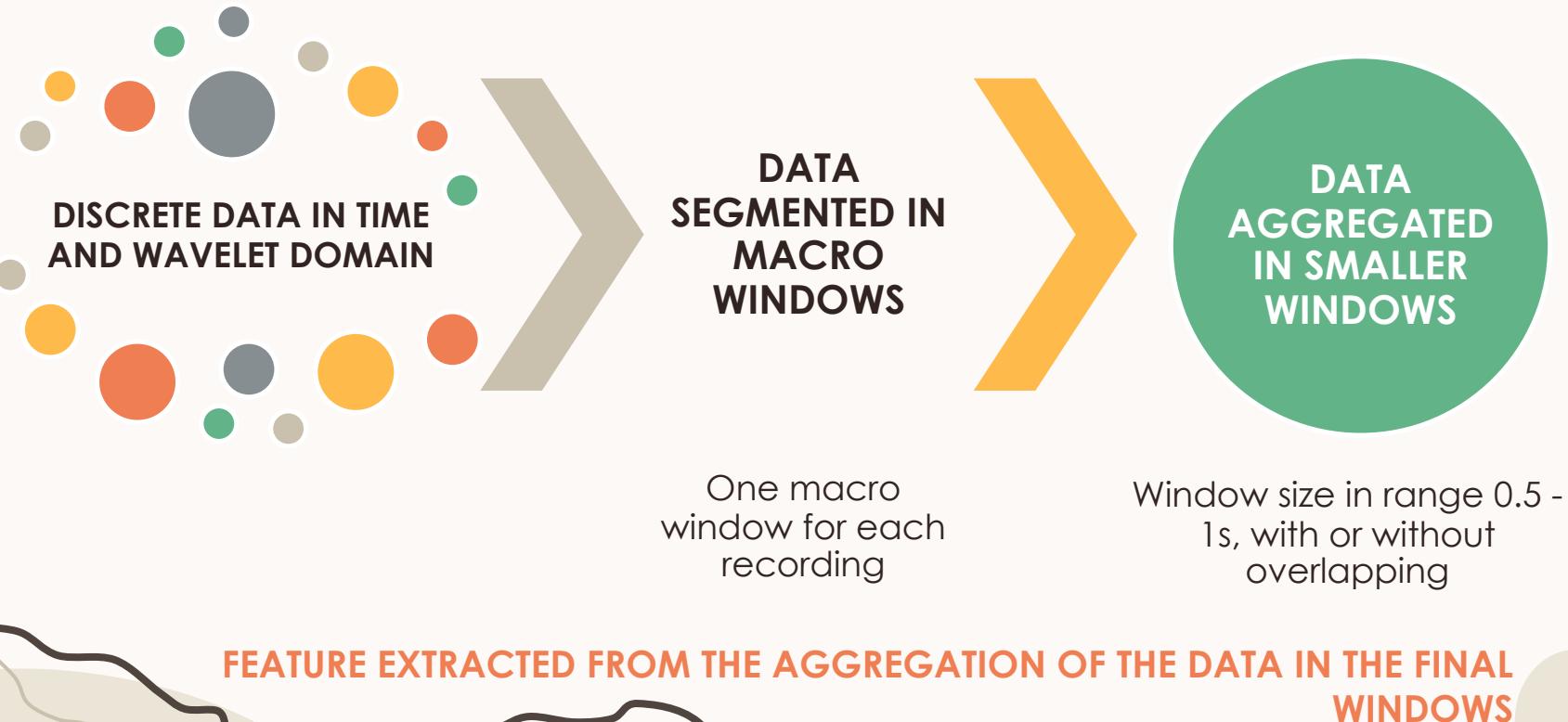
DOMAIN TRANSFORMATION

3 FEATURES FOR EACH SENSOR

2 FEATURES GENERATED BY DISCRETE WAVELET TRANSFORMATION, 1 FOR THE TIME DOMAIN

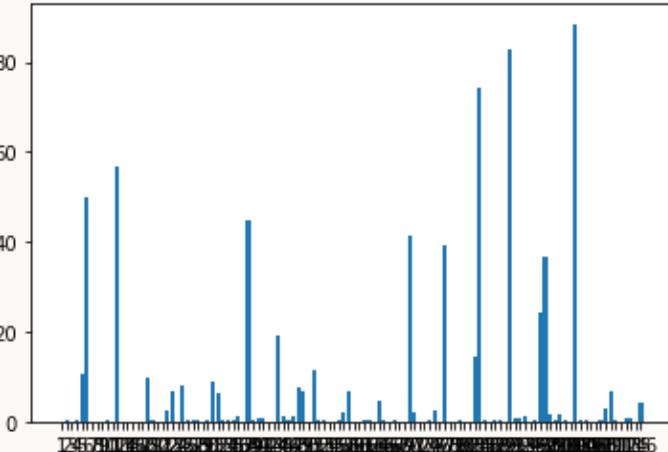


DATA SEGMENTATION



FEATURE SELECTION

Feature importances via coefficients



FEATURE SELECTED
DATA

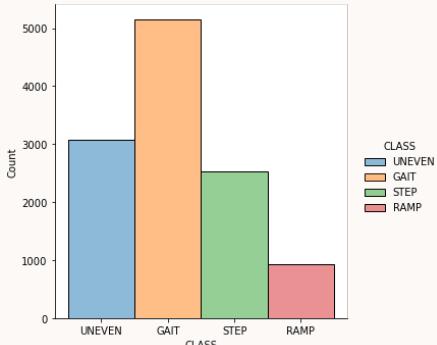
5 – 20
FEATURES



RAW WINDOW
DATA

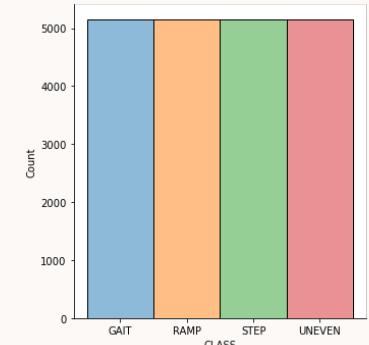
117 FEATURES

REBALANCING



OVERSAMPLING
WITH SMOTE

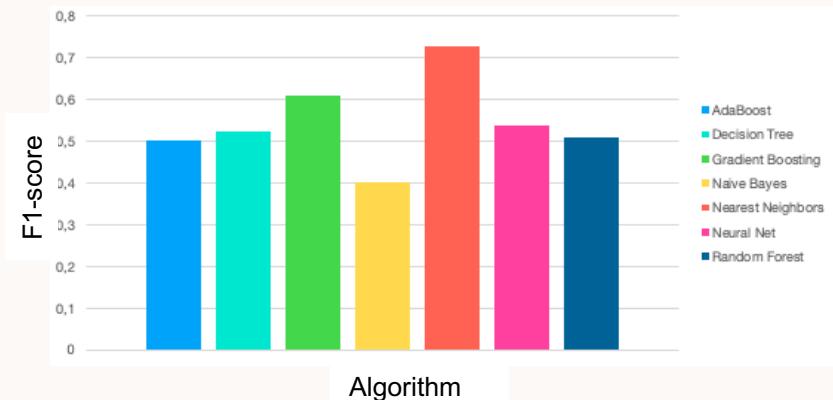
UNDERSAMPLING
WITH ENN



EXPERIMENTAL RESULTS

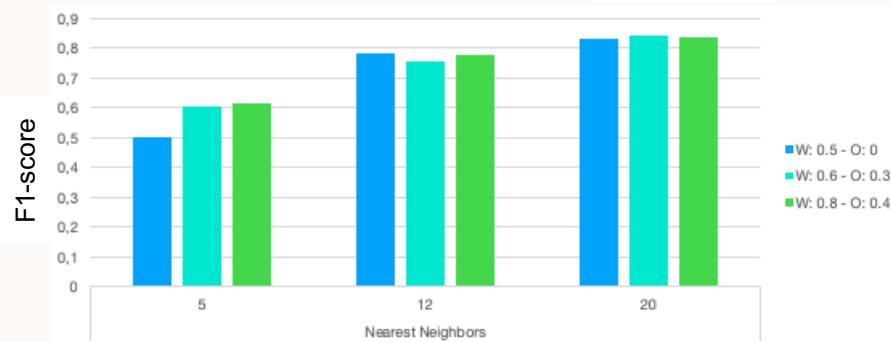
BEST SET-UP

- Window size: 0.6, with Overlap 0.3
- Number of features: 20
- Algorithm: K-Nearest Neighbors



RESULTS

- Accuracy: 0.8511
- F1-score: 0.843





ANDROID APPLICATION

WHAT CAN USERS DO?

EXPLORE

Get an overview of the accessibility barriers of the neighborhood in the form of landmarks in a map



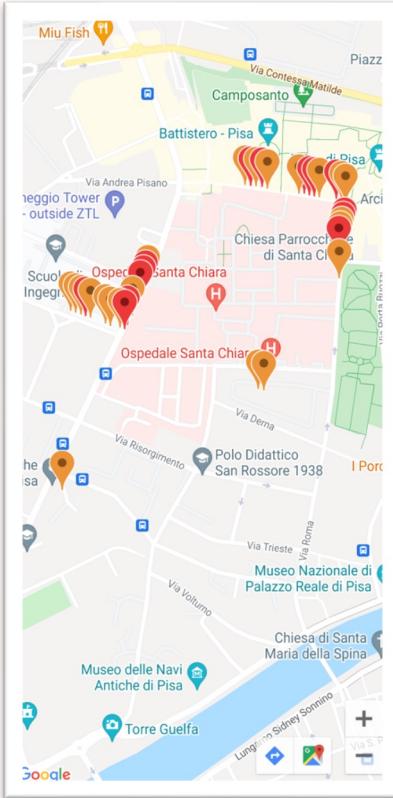
CONTRIBUTE

Contribute to the collection of vibration data while on the move using smartphone's sensors and GPS

SEARCH A ROUTE

Search a path from a source to a destination and discover which is the route accessibility

EXPLORE ACTIVITY



Accessibility barriers of the area are shown using **landmarks** with different colors based on **macro-area inaccessibility scores**:

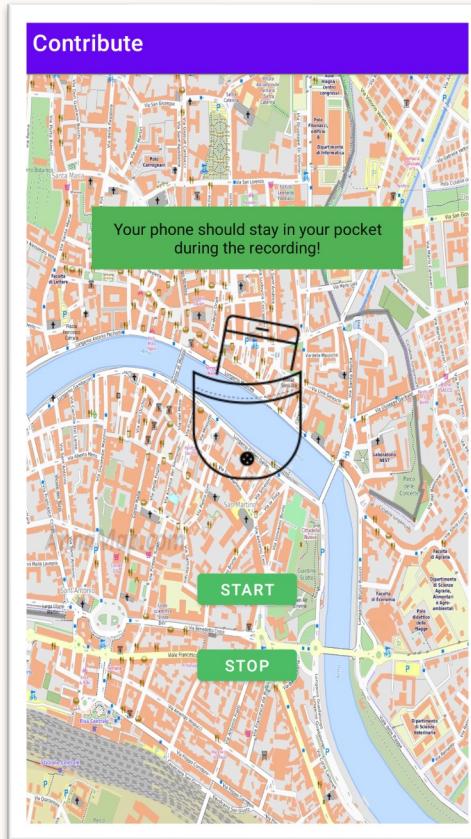


highly inaccessible area: **score ≥ 0.7**



inaccessible area: **score < 0.7**

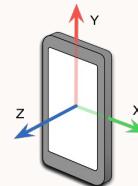
CONTRIBUTION & CROWD-SOURCING



Users can contribute by using *WheelFlow* while they are on the move.

For better results, contributors should put their phone in trousers' pocket during the contribution.

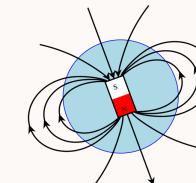
The application exploits:



accelerometer



gyroscope

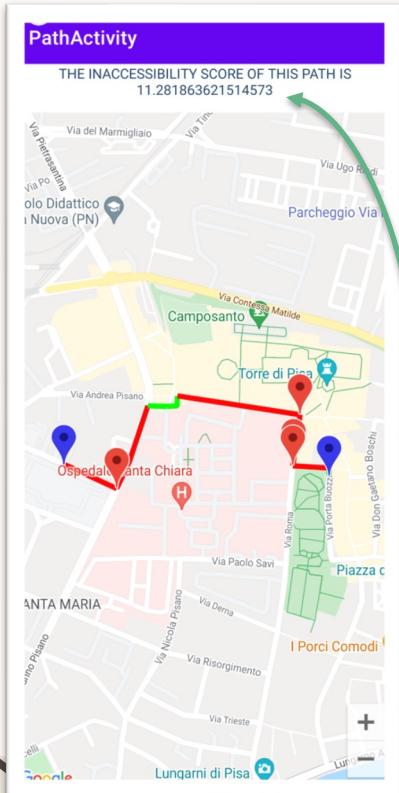


magnetometer



GPS

SEARCH ACTIVITY - ROUTE



Path's accessibility is shown using different colors based on the **sub-route inaccessibility score**:

highly inaccessible sub-route: $\text{score} \geq 3$

inaccessible sub-route: $1 < \text{score} < 3$

accessible sub-route: $\text{score} < 1$

source and destination points

sub-route score = accessibility barriers/km

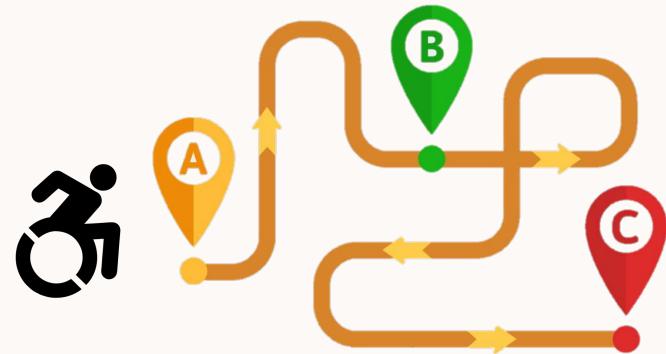
route score = weighted average of sub-route scores



LANDMARK SCORES

ROUTING

How to evaluate a route from a source to a destination for a wheelchair user?

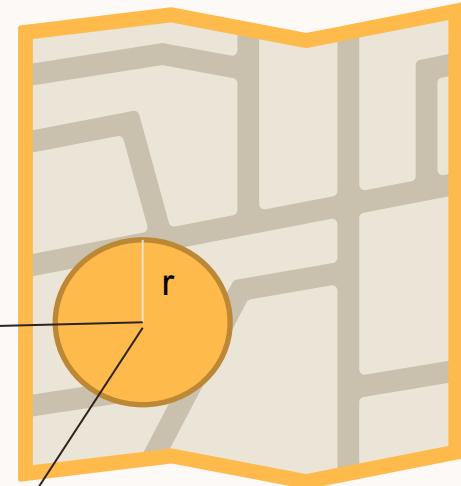
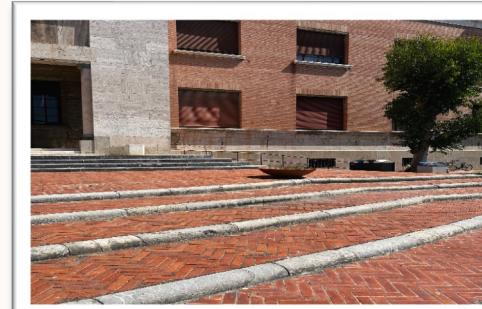


The Google Maps API does not provide information about accessibility for wheelchair users.

INACCESSIBILITY SCORE

Relying on the results of the classifier, we've determined an inaccessibility score for each location.

Close measurements may regard the same architectural barrier. We've decided to compress all accessibility information in a certain radius of coordinates in a **macro-zone**.



CHALLENGES

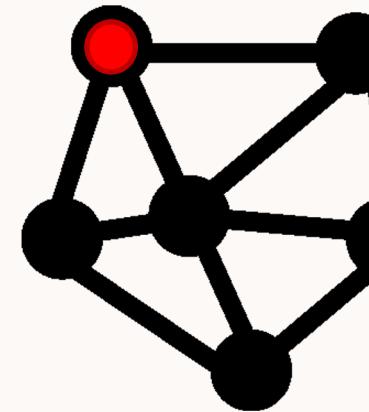
We have multiple potential **unreliable sources of information**.

We need to deal with different factors:

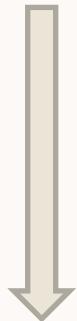
- possible human errors or noises
- unreliability of the classification

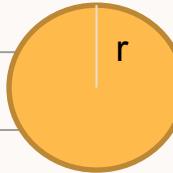
We also have to deal with **outdated** measurements which accounts for temporary or permanent changes.

Solution: not rely on a single classification result but exploit crowdsensing.



MACROZONE'S WINDOW



TIMESTAMP	LOCATION	CLASS
t0			INACCESSIBLE
...			
tN			ACCESSIBLE

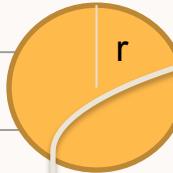


-threshold

0

threshold

MACROZONE'S WINDOW

TIMESTAMP	LOCATION	CLASS
t0			INACCESSIBLE
...			
tN			ACCESSIBLE

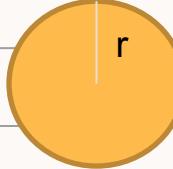


-threshold

-1

threshold

MACROZONE'S WINDOW

TIMESTAMP	LOCATION	CLASS
t0			INACCESSIBLE
...			
tN			ACCESSIBLE



-threshold

+1

threshold

ALGORITHM

Algorithm 1 Accessibility Score

```
1: threshold = function(locationi)
2: score[locationi] = 0
3: for sample close to locationi do
4:   if entry.class == ACCESSIBLE then
5:     score[locationi] = min(score[locationi]+1, threshold)
6:   else
7:     score[locationi] = max(-threshold, score[locationi]-1)
8:   end if
9: end for
```

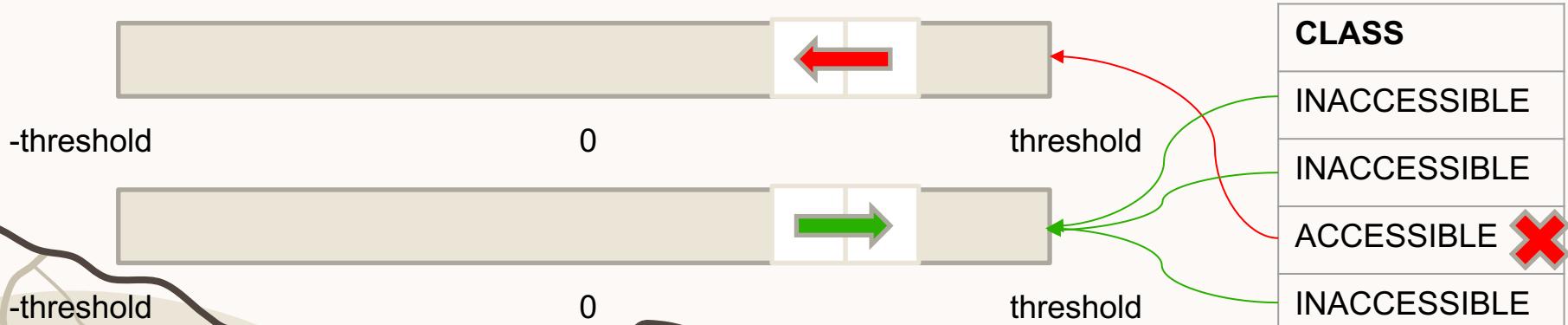
CHALLENGES

This solution allowed us to deal with:

- permanent and temporary changes to the location



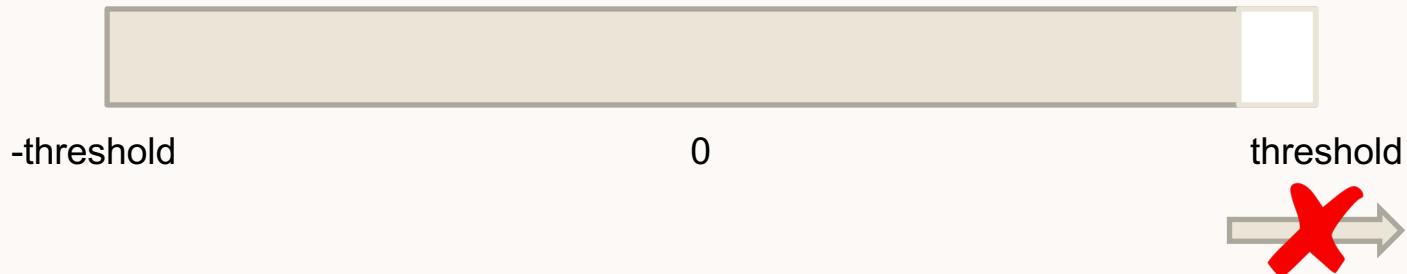
- non reliability of the classification or possible human errors or noise



THRESHOLD

How to choose the threshold?

- very small values -> continuous changes
- very large values -> difficulty in changing the accessibility

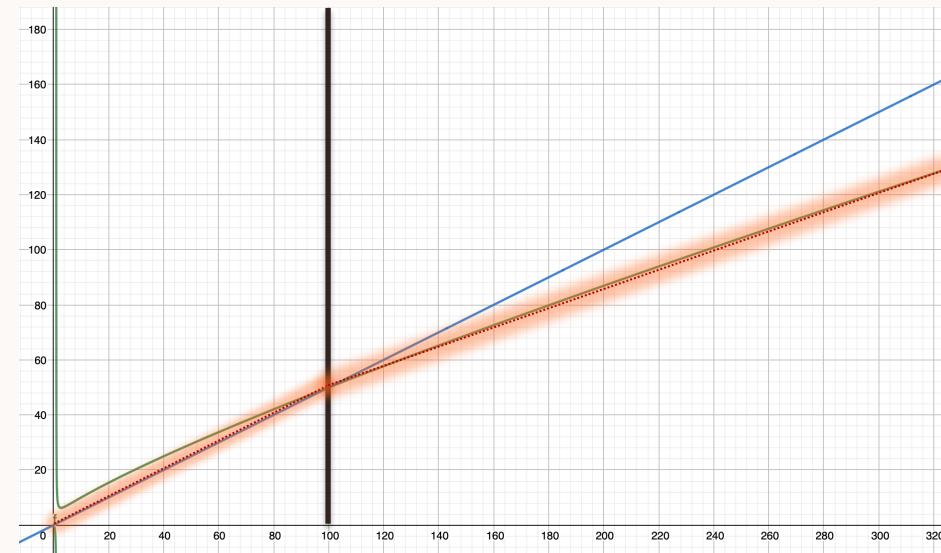


THRESHOLD CALCULATION

Dynamic adaptive threshold proportional to the traffic in the macrozone.

$$\text{threshold} = \text{function}(|\text{samples in the macrozone}|)$$

$$\text{function}(x) = \begin{cases} \frac{x}{2} & x \leq 100 \\ \frac{x}{\log_{10}(x)} & x > 100 \end{cases}$$



CONCLUSIONS

Using our application, local municipalities can discover in real-time about possible local architectural barriers and handling their reparation in a quick manner.

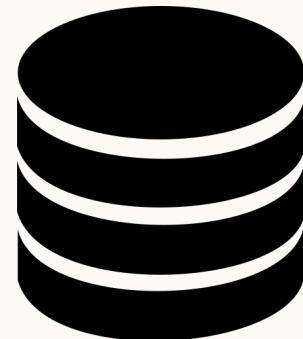


FURTHER IMPROVEMENTS



$|score|$ close to 0

Unreliable
results



More
heterogeneous
dataset



Weight the reliability of
sources of information

THANKS FOR THE ATTENTION!!

