SMARTPHONE APPLICATION TO COLLECT MOBILITY PATTERNS AND GIVE MOBILITY INFORMATION TO USERS

Interdisciplinary project report

SUPERVISED BY:

Prof. Cristina Pronello

Prof. Favio Dovis

DEVELOPED BY:

Zahra Arshadi **S246592**

Elisa Elmazaj **S244180**

Mersedeh Kooshki **S250399**

Berta Mazuecos Velázquez **S263343**

JULY 2019

Table of content

Key words:

Mobility patterns, applications, dynamic zoning, trip purpose.

Table of figures

1. Introduction

Public transport is the transport of passengers operated by public or private companies, typically managed on a schedule, on established routes. The most important transport challenges are often related to urban areas and take place when transport systems, for a variety of reasons, cannot satisfy the numerous requirements of travellers in the urban context. Congestion is one of the most prevalent transport problems in large urban areas. As cities continue to become more dispersed, the mobility needs on longer distance are increasing and the car is the most used mode, increasing the air pollution. Thus, to spur a modal diversion towards public transport, a service more tailored to users' needs is a key condition. To this end, the knowledge of mobility patterns is necessary to better programme the mobility services, making the data collection the main challenge.

The new technologies can help to collect the mobility data; there are a number of smartphone applications that track the users' trips, trying to individuate the transport mode and, in some cases, to induce the purpose of the trip. None of such application performs well due to the difficulty of such task. One of such applications is called "Mobilità DinAMICA" and aims to collect mobility data to give them back to users in form of travel diary to make them aware about their mobility footprint. The interdisciplinary project is focused on adding the information related to the purpose of the trip, to improve the data collection.

The next section will present the literature review; then, the objective of the project and the methodology follow. The results of the work will then be reported and, finally, some conclusive remarks given.

2. State of the art

The first activity in the project was related to the literature review in order to search similar applications. A few apps developed to understand the activity of the users in order to collect their mobility patterns and, hence, find the purpose of the trips are:

- "**Woorti**": the main objective is obtaining a better understanding about how people use and value their travel time, while sharing the results with transport suppliers and public authorities to enable them to offer better experiences for travellers. [4]
- "**E-mission**": is an app that allows you to track your travel modes and your travel carbon footprint and compare them against our other users. [5]
- "**MotionTag**": provides a single data source for all modes of transport across all mobility providers. [6]
- Others app are: "App-ask", "TravelVu", "Sentience".

Less similar applications instead are: "ThetruthSpy", "Spyzie", "MSPY", "Famisafe", etc… Actually, their common point with "Mobilità DinAMICA" is the tracking of the user through GPS, but the objectives are different.

The literature related to the processing part consists of papers on Data Mining and Machine Learning methods, which address the problems of public transport, its features and the key elements in other public transport projects.

3. Objectives and methodology

The main objective of this project is the improvement and extension of the application "Mobilità dinAMICA", with the aim of providing further services to the users as regards the current information; precisely:

- to find the purpose of users' trips;
- to create dynamic maps based on the purpose of the trips. This map is dynamic because changes according to the day of the week and the hour (we consider two slots: morning and night)

To this end, a methodology has been defined articulated in the five following phases as shown in *Fig. 1* and detailed below.

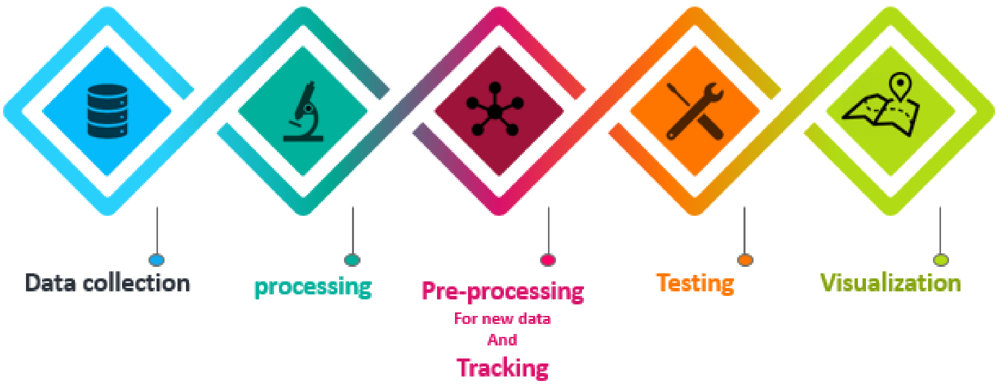

Figure 1. Methodology phases

1. *Data Collection*: collect data from the application "Mobilità Dinamica" and it comes in the same data format: json files, one for each registered trip. For such purposes different collections in MongoDB are created: one containing all old data trips, one that saves the census zones of Turin and one that groups POIs.
2. *Processing:* employ algorithm in Python to predict the purpose of the trips combining spatial (map, POIs, census zones, origin/destination coordinates) and temporal

information (differentiate trips by origin/destination timeslot and day of the week). So, POIs are matched to the zones, the purpose of each trip is induced, and the zones are labelled according to the time of the day.

3. *Pre-processing and Tracking:* the new data collected are filtered in order to get useful trips; then, a map done with Google Maps shows the trips of the user divided in days layers. This aims to generate patterns for predicting the conduct of the users in the future.
4. *Testing*: the model is tested using the new data collected from the application. In this way potential errors can be corrected to ensure a good performance.
5. *Visualization:* the dynamic map showing the new zones of Turin along the day with respective label is finally created. The designed friendly webpage allows direct link to both maps.

In *Fig. 2,* technical steps represent the different parts mentioned above with more details.

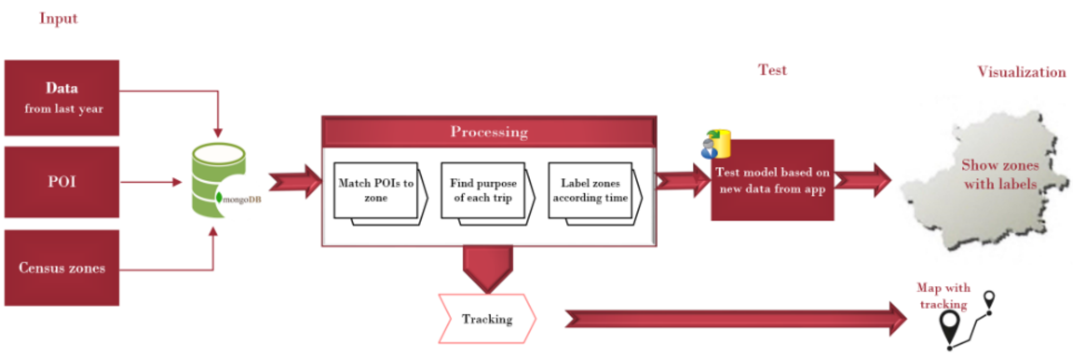

Figure 2. Technical steps

The tools used to deploy the work are the following:

- **MongoDB**: it is the database where reside all data related to the users and trips. It's the most popular database for modern apps and it stores data in flexible, JSON-like documents**.**
- **Python programming language**: it is the language used to develop the program for pre-processing and processing the data.
- **QGIS**: it is exploited with the aim to create, edit, visualise, analyse and publish geospatial information on Windows.
- **Microsoft visual studio**: It is a code editor that can make HTML pages.

Figure 3. Tools

From data collection to the creation of the map with labelled zones, the data follow the steps showed in *Fig. 4*.

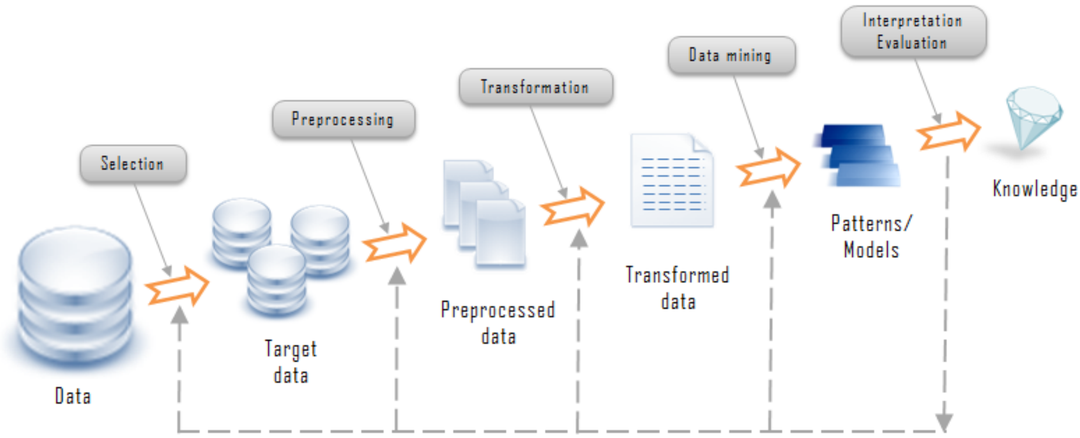

Figure 4. Data steps

- In the "***Selection***" phase the data is selected: a subset of available attributes is considered, discarding useless attributes.
- In the "***Pre-processing***" phase the data is cleaned of possible anomalies and outliers. In our case the anomalies are generated by trips with a short duration, or by erroneous coordinates present in the data. So only the trips with a considerable duration and that are inside the city border are chosen.
- In the "***Transformation***" phase the data is discretized to improve the process analysis. In our case data is collected in MongoDB (JSON file), and then transformed in Geojson for the processing and testing part.
- In the "***Data Mining***" data patterns are created through processing using data collected last year. Then the system is tested with the new data collected by the application.
- In the "***Interpretation***" phase a visual representation of the data allows to infer useful knowledge. In our case, the labelled zones are visualized in a dynamic map. In addition, we made a visualization of tracking users on google map.

3.1 Data collection

Data collection is the systematic approach to gathering and measuring information from a variety of sources to get a complete and accurate picture of an area of interest.

3.1.1 Collect Point of Interest

Point of Interest, or POI, is a specific point location that someone may find useful or interesting. POIs represent our ending points, which mean trips destination points. In this regard the most important POIs in Torino are chosen, related to different activities:

- university;
- hospital;
- leisure (entertainment) places;
- touristic places;
- shopping centres;
- transport nodes, i.e. train station;
- work places;

Some of them are depicted in *Fig. 5, general POIs in Turin*: the yellow points indicate a part of the Turin POIs we collected. They are obtained inserting the coordinates of the POIs (latitude, longitude) in the map.

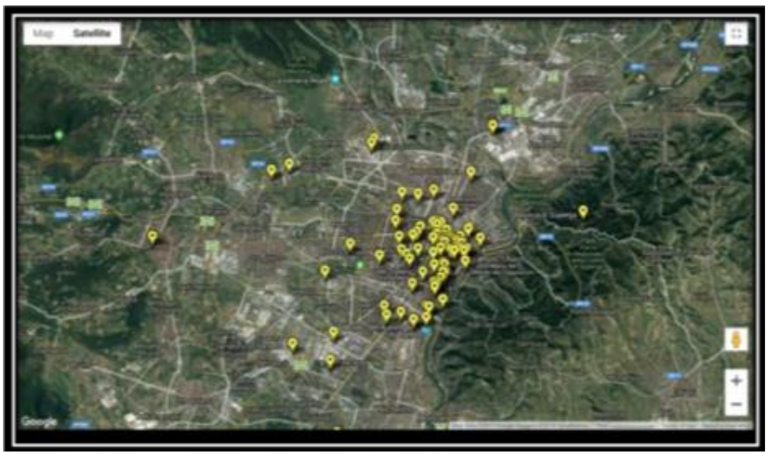

Figure 5. General POIs in Turin

From the project of the previous academic year the file with the Turin census zones (*Fig.6*) was obtained, which are smaller than zones set up by the municipality (*Fig.7)* and usually contained in one or two blocks. In the *Figures* 6 and *7*, it is possible observe how they differ in shape, dimension, quantity and position.

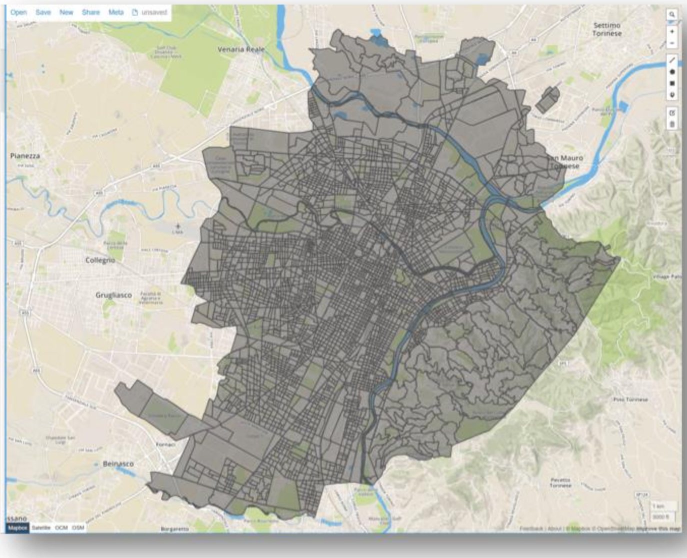

Figure 6. Census zones

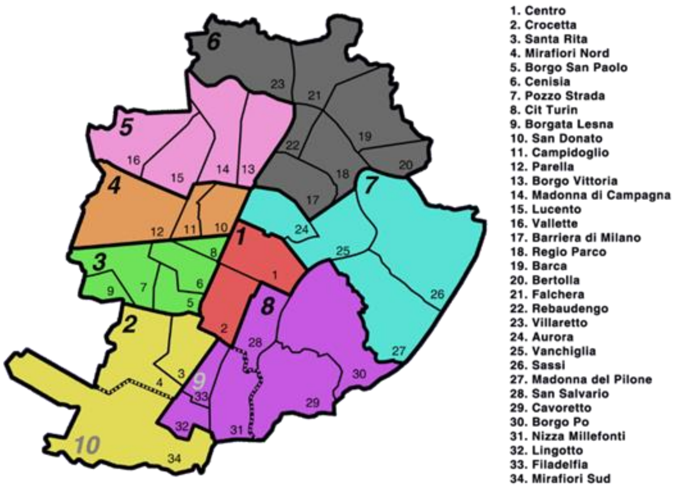

Figure 7. Municipality Turin zones

It is clear that, in *Fig.7,* there are less and bigger zones, while, in *Fig.6* are much more numerous and smaller. Considering census zones, mobility can be studied with more details, this is the reason why they have been used in our project.

For each zone, at least one POI is necessary in order to assign a label to each zone. Actually, only the zones containing destination of users' trips were used in the project. The majority of the users are students, so the most crowded zones observed are related to university.

3.1.2 Get Data directly from the app

Thanks to the availability to download the data form the app through an API, the data were downloaded. In the fields "from=" and "to=", the period of data to observe has been written. Both fields are in Unix millisecond format.

3.1.3 Get Collection Database of last year

As the available data were quite few, the data already collected last year were added, being those more numerous than the current ones. To this end, the API was used, converting, accordingly, the dates in milliseconds.

Data format is a JSON file, which consists of:

- User id;
- Coordinates (latitude, longitude);
- Timestamp;
- Mode of transport.

In *Fig. 8* the collection of the new data in MongoDB with its features is shown. Each field, which represents a trip, has the characteristics listed above.

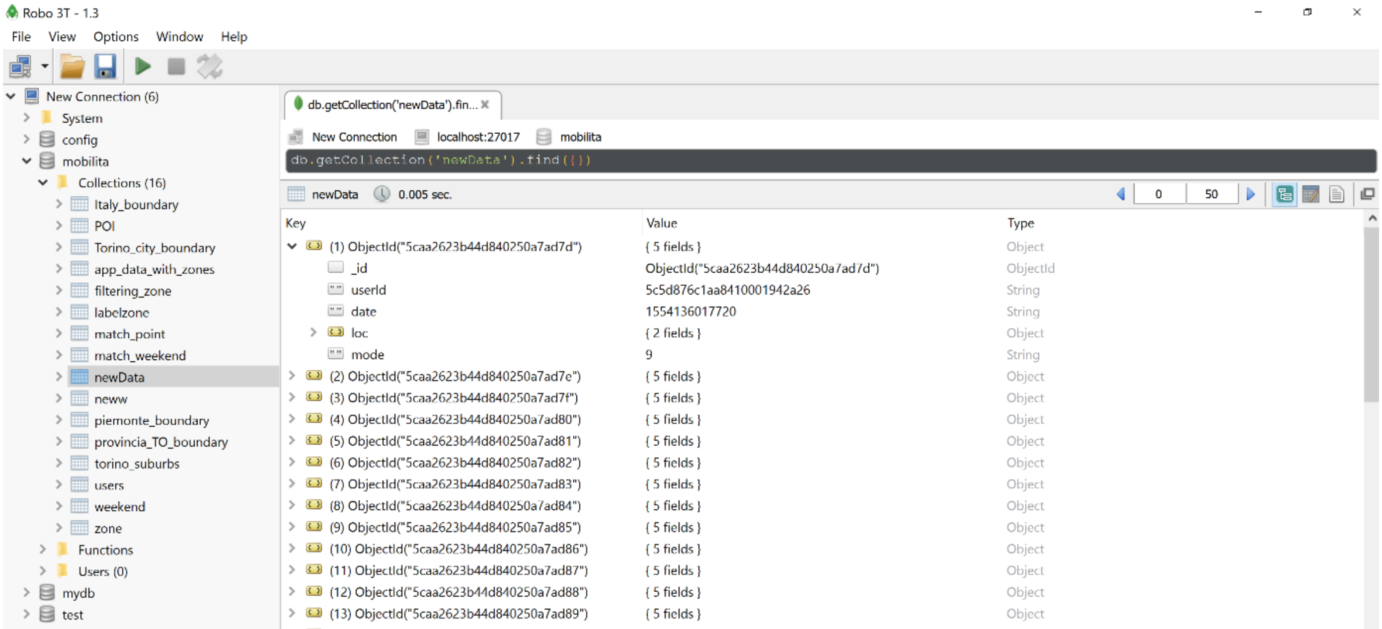

Figure 8. New Data Collection

Then, the pre-processed data is matched with the census zones in MongoDB. In this way for each origin/destination point the relative census zone to which it belongs has been saved; thus, for each trip, we have *origin_zone id* and *destination_zone id,* observable in *Fig. 9, match origin/destination to zone id,* looking at the "properties" field of each trip.

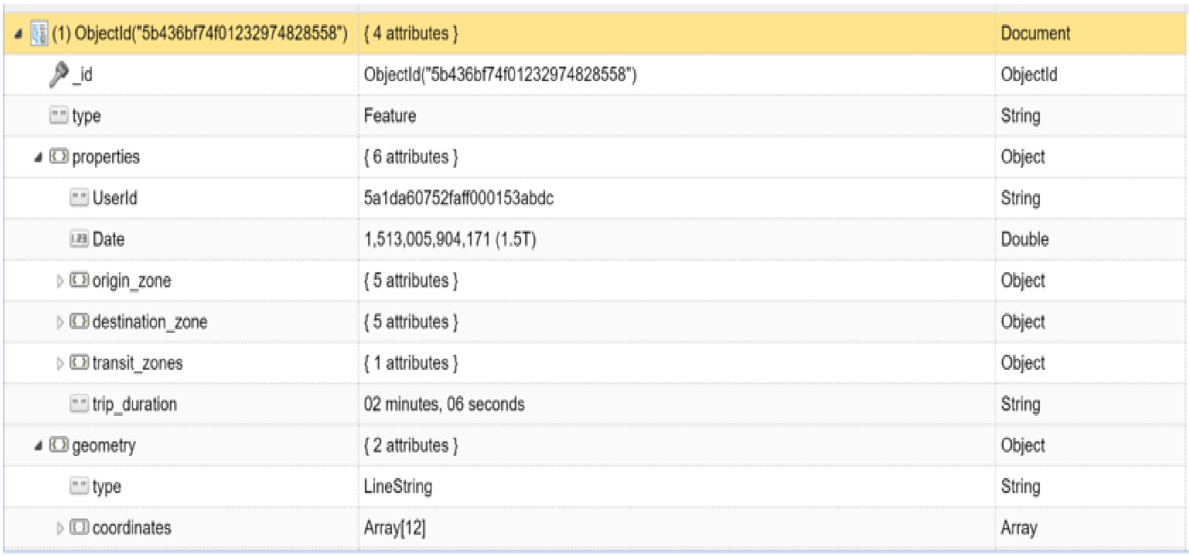

Figure 9. Match origin/destination to zone id

3.2 Data Analysis or Processing

Data analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

3.2.1 Match POI to the zones

The main task is to locate each POI into a specific zone based on its coordinates (Latitude, Longitude). Therefore, each POI is related to a certain zone_Id. It is shown in *Fig. 10, Match POI to zone id.*

In this way each POI will be characterized by three main features:

- Coordinates
- Type
- Zone id

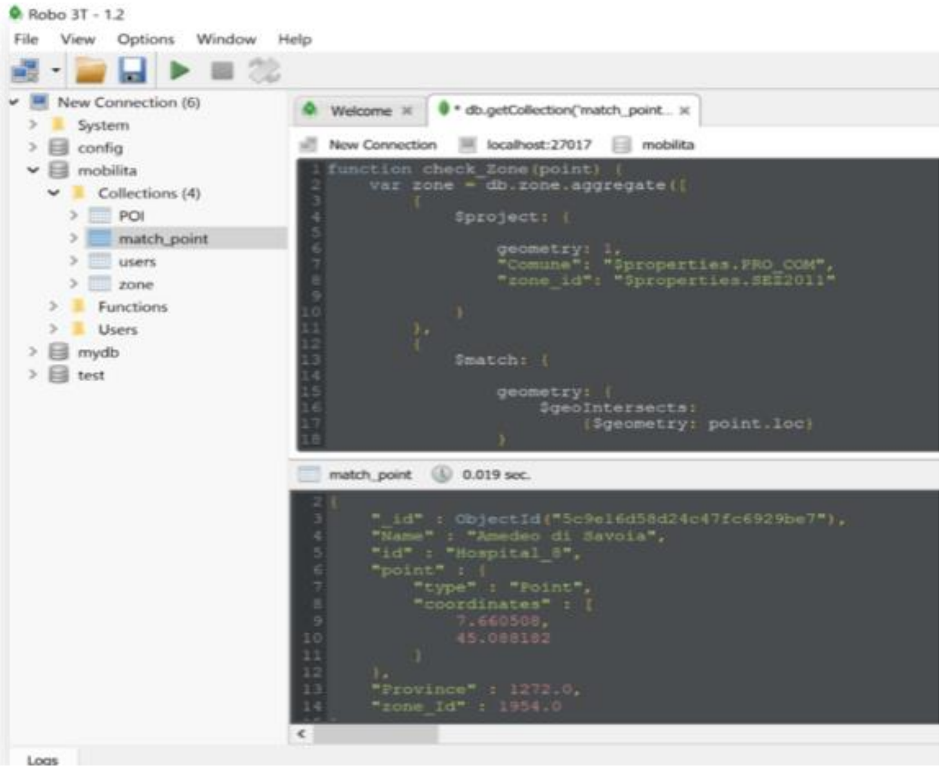

Figure 10. Match POI to zone id

3.2.2 Definition of the algorithm

The steps of the algorithm designed in this project are:

1. apply algorithm to predict purpose of the trips based on the nearest POI, combining spatial and temporal information (label trips);
2. count number of trips with the same purpose in each zone;
3. label each census zone using the information given by CSI Piemonte and the data found in Google Maps;
4. according to the purpose and time, eventually change the label of the zone.

The plot depicted in *Fig.11, labelling steps,* shows how long collected trips last. The duration time in the plot does not refer to the duration time considered in the algorithm:

- duration time in algorithm refers to whole duration time of a trip from starting point to ending point;
- duration time in the plot means the time in which a user stops in a specific location, and if it lasts more than 30 minutes it is considered as a trip with purpose, otherwise (less than 30 minutes) the trip still continues (it is not finished yet) and it is considered as a transit trip.

The block diagram presented in *Fig.11* shows the steps that have been followed to label the trips. First step is the definition of a trip "Transit" or "Trip", if it is "Trip" then the next step will be labelling according to the time of the day.

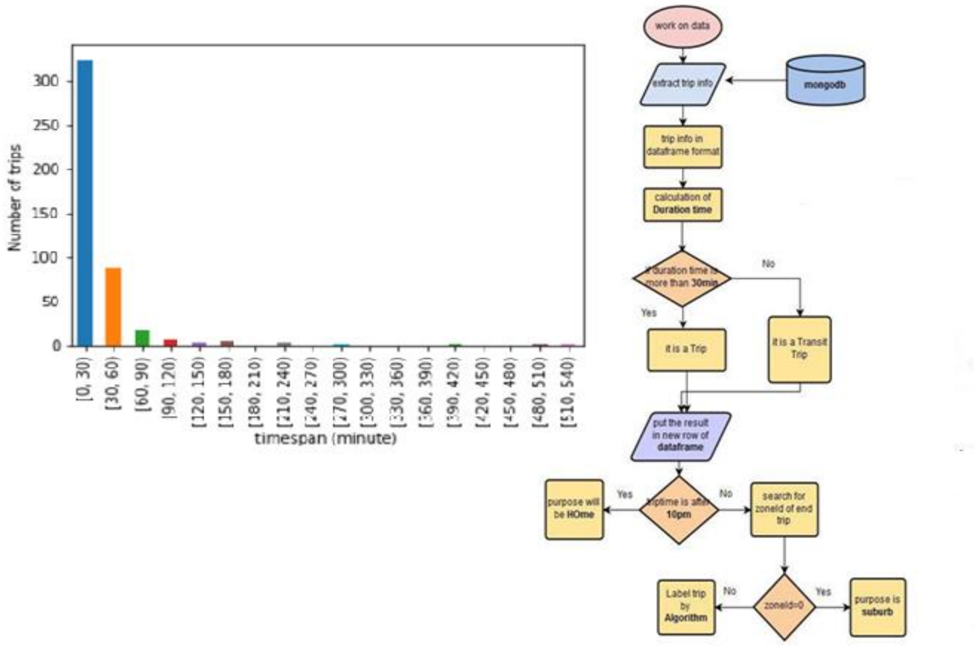

Figure 11.Trips separation

3.2.2.1 Label trips with purpose

One of the main goals to be achieved was finding the purpose of the trips. Firstly, all the trips and all the POIs are stored, each with relative *zone_id*. Then, by using Python programming language, the Euclidean calculation of the distance between the trip destinations coordinate and every POI contained in the destination census zone is calculated: the purpose assigned to the trip is the type of the nearest POI.

To find purpose of each trip, two JSON files are imported in Python:

1. users' data file which consists of *userId*, *destination_zoneId*, *origin_zoneId*, and date in Unix time is converted into human readable format;
2. file related to POIs which consists of *zone_id*, name and coordinates (latitude, longitude).

Then a code in Python is implemented to find the nearest POI for each trip destination (Euclidean distance); in this way, each trip is finally labelled.

Considering the different destinations, people might have during workdays and weekends, trips are divided in two temporal groups: weekday and weekend.

The POIs file is divided into two files, one for weekend and one for weekday because the purpose may change according to the hour of the day and the day of the week. For instance Politecnico is an *Academic* purpose in weekdays (during the day), but, it is an Entertainment purpose (Mixto) in the weekend.

The day is split into two portions of time:

- daytime: from 8am to 7pm;
- night and early morning: from 7pm to 8am.

In *Fig. 12, labelling trips,* is shown the first important result obtained in Python: to each trip a field label is assigned, which represent the "purpose" column.

In [27]: UserData

userId	origin_id	origin_date	longitud_orig	latitude_orig	destination_id	destination_date	longitud_dest	latitud-dest	duration	transit	purpose
994d9c9e77c0001b73c33	231	2017-10-12 17:46:48	7.691520	45.065525	0	2017-10-12 17:47:15	7.695906	45.066072	NaN	trip	stadio_1(Suburb)
994d9c9e77c0001b73c33	237	2017-10-12 20:01:10	7.692031	45.064449	232	2017-10-12 20:04:23	7.692731	45.065297	NaN	trip	touristic_area_restaurant_2
994d9c9e77c0001b73c33	231	2017-10-12 17:44:19	7.691520	45.065525	231	2017-10-12 17:45:23	7.691520	45.065525	NaN	trip	touristic_area_restaurant_2
994d9c9e77c0001b73c33	0	2017-10-12 18:44:14	7.695906	45.066072	227	2017-10-12 18:44:40	7.692144	45.066354	3938.0	trip	touristic_area_restaurant_2
994d9c9e77c0001b73c33	227	2017-10-12 19:48:12	7.692144	45.066354	237	2017-10-12 19:50:18	7.692031	45.064449	NaN	trip	touristic_area_museum_9
l649fcc9e77c0001badd8d	2832	2017-10-12 17:53:22	7.697783	45.097775	1770	2017-10-12 18:12:40	7.696186	45.097878	NaN	trip	Hospital_3
994d9c9e77c0001b73c33	232	2017-10-12	7.692731	45.065297	0	2017-10-12	7.695906	45.066072	621.0	transit	Transit

Figure 12. Labelling trips

3.2.2.2 Count number of trips with same purpose (POIs)

Since the final goal is labelling the zones, the next step is the calculation of how many trips there are for each POI. This is done considering the purpose column depicted in *Fig. 12* and finding out which POI defines the majority of trips purpose, for each specific zone.

3.2.2.3 Label zones and create dynamic zoning

For each zone, the POI with the greatest number of trips is found. The type of this POI is going to give the final label to the relative zone id, and so the *final_purpose*. Labelling is done considering the day of the week:

- weekday (Monday-Friday);
- weekend (Saturday-Sunday);

and two different time slots of the day:

- daytime (8:00-19:00);
- morning and night (19:00-8:00).

In *Fig. 13, labelling zones,* the result obtained with Python is shown: to each *destination_id* (i.e. census zone id of the trip destination point) a *Final_purpose* is assigned. Last column represents the trip counter.

	destination_id	dayOFweek	portion_of_day	Final_purpose	Final_purpose.1
1	0	weekday	daytime	Train	2
2	0	weekday	morn & nigth	Academic	1
4	0	weekend	daytime	Entertainment	1
5	0	weekend	morn & nigth	Entertainment	1
8	28	weekend	morn & nigth	touristic	1
9	44	weekday	daytime	touristic	2
10	67	weekend	daytime	touristic	1
11	80	weekend	daytime	touristic	1
12	97	weekday	morn & nigth	Home	1
13	104	weekday	morn & nigth	Home	1
14	105	weekday	morn & nigth	Home	1
15	117	weekday	daytime	job	1
16	125	weekday	daytime	job	1
17	133	weekend	daytime	touristic	1
18	148	weekday	daytime	Train	1
19	227	weekday	daytime	touristic	2
21	228	weekday	morn & nigth	Home	2
22	228	weekend	morn & nigth	Home	2

Figure 13. Labelling zones

3.3 Data pre-processing

Data pre-processing is a data mining technique that transforms raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends and is likely to contain many errors. It is a proven method of resolving such issues.

3.3.1 Filtering new data

Once finished labelling zones, it is essential to test the model designed before. The data is directly obtained from the dynamic mobility application. Therefore, pre-processing on the new data must be done, in order to discard data which are out of the desired area applying the hierarchical search method, as shown in *Fig.14*.

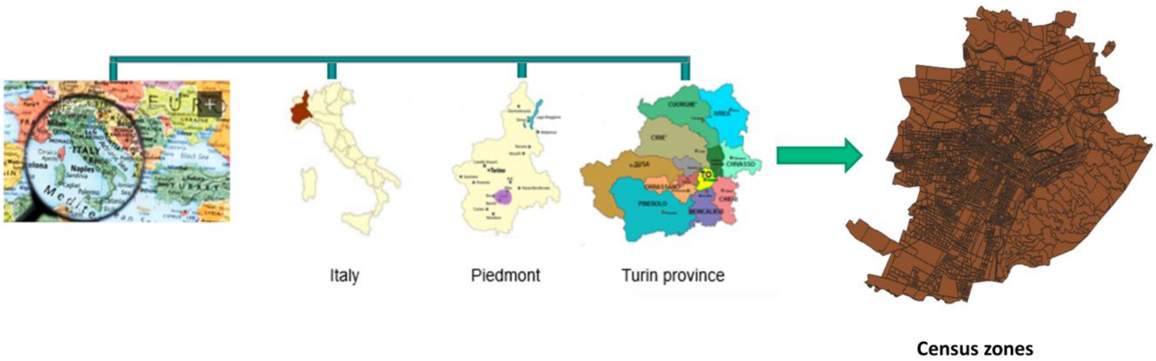

Figure 14. Hierarchical search

3.3.2 Track users' trips

During pre-processing part, filtering on the users' data is performed, till only the fields allowing to track users' trip are extracted, to understand users' behaviour and maybe to do some prediction about users' destination in the future, which can be an alternative to costly surveys. The table in *Fig.15, tracking users,* shows the results obtained thanks to Python.

However, data is very low, so to do the tracking part the best choice was to examine the travels of one user with the largest number of trips.

userId	dest_date	dest_time	dayOFweek	destination_id	Final_purpose	
5900cd88c9e77c0001b57365	2017-10-20	18:47:16	Friday	313	touristic_area_museum_15	1
	2017-10-22	16:37:57	Sunday	778	touristic_area_museum_12	1
	2017-10-23	07:27:04	Monday	775	job_50	1
		12:58:24	Monday	836	university_unito_11	1
		15:30:55	Monday	362	university_politecnico_X2	1
		18:01:52	Monday	313	job_50	1
	2017-10-24	11:30:50	Tuesday	44	touristic_area_restaurant_6	1
		12:20:05	Tuesday	44	touristic_area_restaurant_6	1
		14:02:31	Tuesday	313	job_50	1
	2017-11-24	15:39:13	Friday	797	university_politecnico_4	1
		16:23:39	Friday	797	university_politecnico_4	1
		18:00:16	Friday	797	Entertainment_food	1
		18:05:24	Friday	797	Entertainment_food	1
		18:10:39	Friday	797	university_politecnico_4	1
		18:15:49	Friday	797	Entertainment_food	1
		18:20:59	Friday	797	university_politecnico_4	1
		18:26:16	Friday	797	university_politecnico_4	1
	2017-11-25	00:09:43	Saturday	313	touristic_area_museum_12	1
		09:32:51	Saturday	313	touristic_area_museum_12	1
		10:44:17	Saturday	314	touristic_area_museum_12	1

Figure 15. Tracking users

3.4 Test the Model

Testing is a process of finding out the defects and bugs in a software program. It verifies if all the features that are properly working or not.

3.4.1 Select an appropriate machine-learning algorithm

For this project two machine-learning algorithms were considered, the most suitable to our case:

a. K mean clustering algorithm (un-Supervised algorithm);

b. Random forest algorithm (Supervised algorithm).

At the end, the second one is chosen (*Fig.16, random forest algorithm*) because it is a supervised algorithm and, due to our labelled data, supervised algorithm is needed.

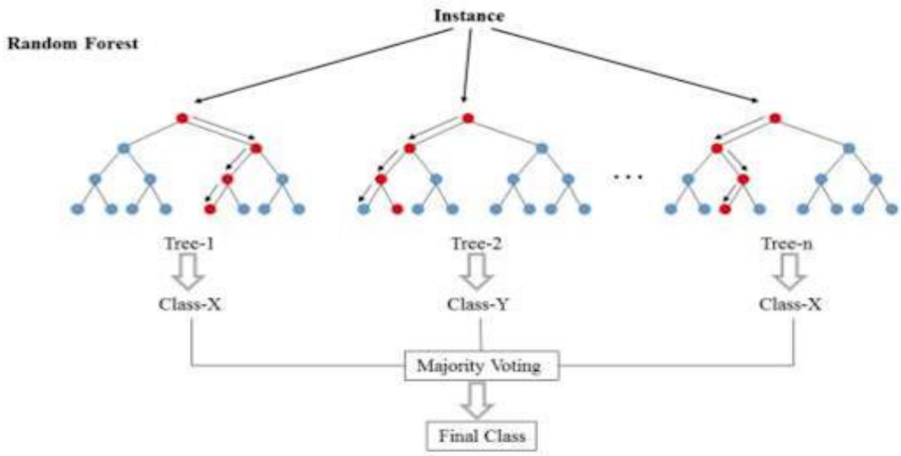

Figure 16. Random forest algorithm

Random forests are made of many decision trees. As it can be observed in *Fig.16*, each decision tree is created by using a subset of the attributes used to classify a given population (they are sub-trees, see above). Those decision trees vote on how to classify a given instance of input data, and the random forest bootstraps those votes to choose the best prediction.

3.4.2 Find errors and solve them

The first time the model was checked, the value of correct prediction was equal to 68%. Since it should be improved. The two specific approaches were used in order to increase prediction value:

1. feature engineering: remove the exact time or day of trips to make it more general;
2. grid search: it is selected appropriate input parameters of machine learning algorithm that can provide better results.

After applying these two methods in our model, the accuracy of prediction increased to 71%. This percentage of correctness seems good enough in our case; indeed, there were not enough data to work with, thus, obviously, with more data the result of prediction could be increased.

3.5 Visualization

QGIS is used in order to project POIs on the census zones, visualise them as well as to export centroid of the zones.

3.5.1 Match POIs to the Torino-zones

The first visualisation of data refers to the matching POIs on the census zones file. POIs are converted into json format, then Torino-zones and POIs files are imported into QGIS software, as shown in *Fig.17*. The dots outside the province of Turin are considered "Suburb" and they are not used in our project.

Figure 17. QGIS

3.5.2 Find centroid of Torino-zones

To obtain the second visualisation, it was necessary to find the centroids of the census zones, as it is shown in *Fig.18*, and exported the centroids in a GeoJSON file. This was done in order to use their coordinates to match with label of zones for final visualisation (zones with their labels).

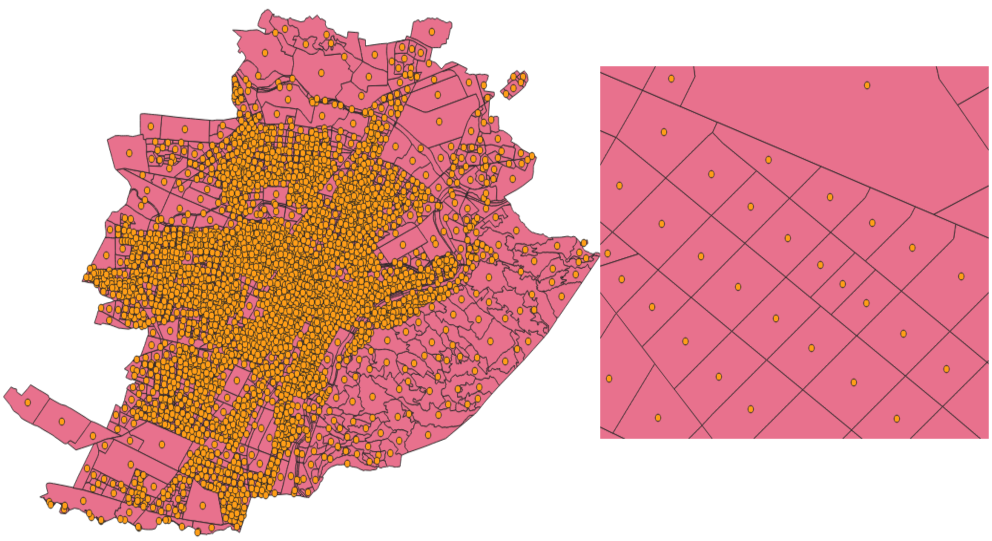

Figure 18. Center of zones

3.5.3 Create map with labelled zones

For the creation of the map, Turin-zones file is imported to Google Maps in different layers and labelling zones files are added. (four files due to two parts of the week and day). The map is accessible from the following site:

https://www.google.com/maps/d/viewer?mid=1XzgDJgLPDUJ9j0189KJULS_oOqW5xn7G&ll=45.07354343825701%2C7.675591499999996&z=11

3.5.4 Create map with a user tracking

Firstly, data is filtered in order to pick those related to the user with more trips.

For the tracking, four csv files are made and imported to Google Maps: they are divided in days and each file contains the user trip related to that day; the fields are:

- origin coordinates (latitude, longitude);
- destination coordinates (latitude, longitude);
- timestamp.

Each layer of the map represents the day of the week in which trips are present, with the relative trips displayed. It is accessible from the following link:

https://drive.google.com/open?id=1b02wt2WNORwp_4HWP6TkDk16B9-Iw6xP&usp=sharing

Both maps will be explained better in the results section.

4. Results

Since our main goal is doing dynamic zoning, the best way to show it is to create a temporary open webpage to visualize the two final maps. It is a friendly-user page, with two pictures of the application "Mobilità DinAMICA", because our data comes from there.

On the top-right corner two important buttons appear:

- Dynamic Map: clicking on it, the user will be re-directed to the map of Turin containing the labelled zones;
- Tracking: clicking on it, the user will be re-directed to page showing the map of tracking.

On the lower left corner there is a blue arrow button: clicking on it a new page opens, where user can download the application from AppStore of Google Store, as shown in Fig.18.

Both maps are created using Google Maps.

Fig. 19 represents the main screen of the website, where the title of our project and the names of the group members are reported.

The URL of it is reported in the section above.

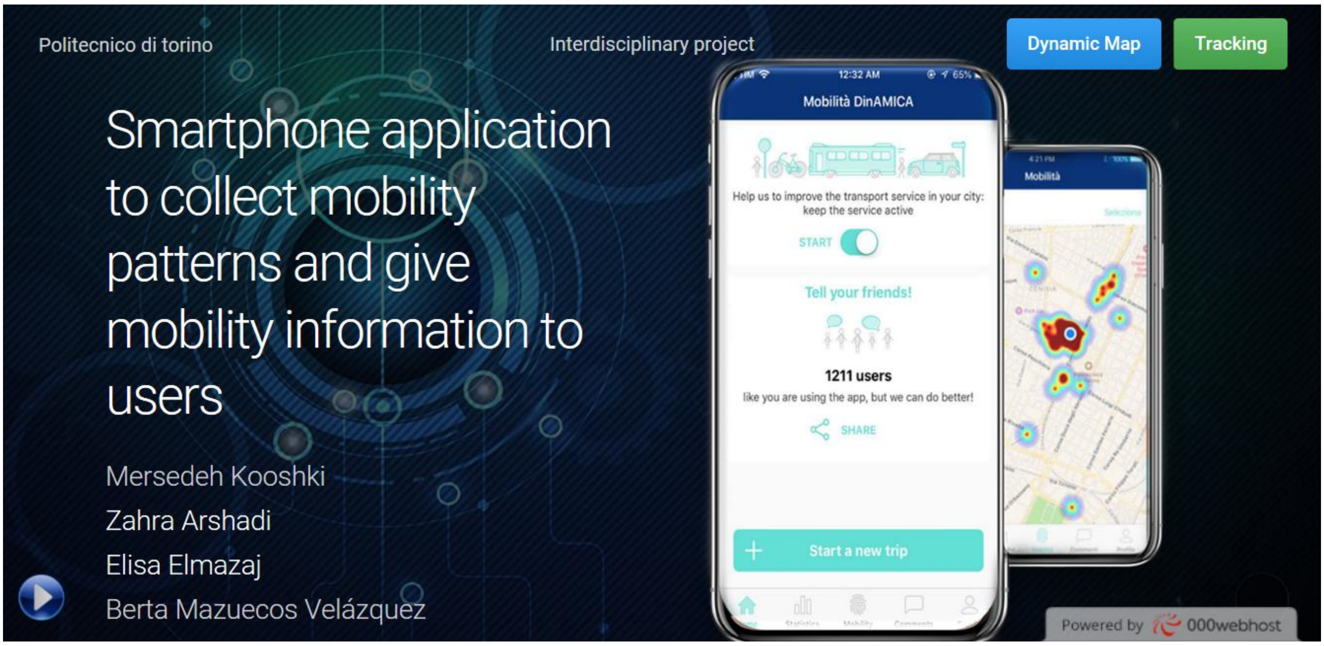

Figure 19. Friendly Website

The second web page is shown in fig.20 and users can download the dynamic mobility application by clicking on the specific buttons.

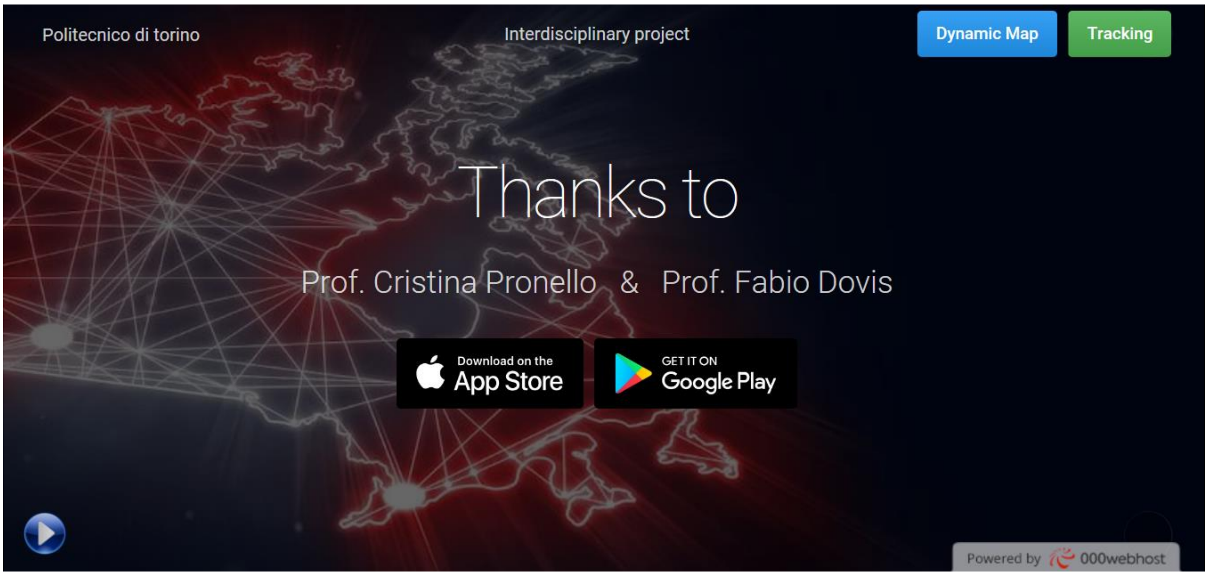

Figure 20 – Second webpage

4.1 Dynamic map

Using Google Maps, we created three layers related to census zone with kml format, then put four layers where each of them covers different days of the week with various time intervals; we need to consider that format of these file is csv. Actually, we create a gradient colour based on different category of labels. These categories are:

- Academic
- Touristic
- Hospital
- Transit
- Entertainment
- Home
- Job
- Train
- shopping

The figures *21, 22, 23, 24* display different options that provide, by clicking each of them, a map with particular labels, specific for each option.

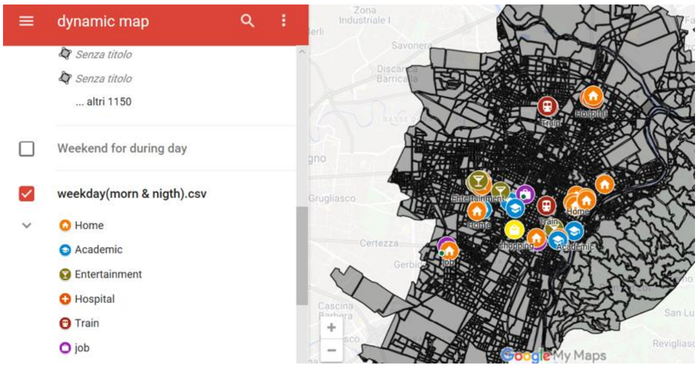

Figure 21. Weekday (Morning & night)

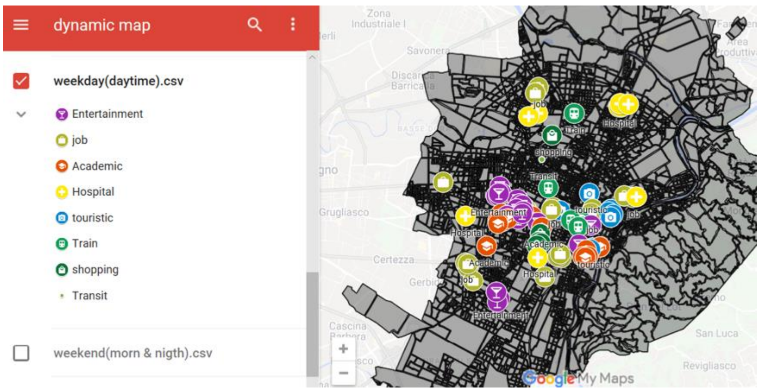

Figure 22. Weekday Daytime

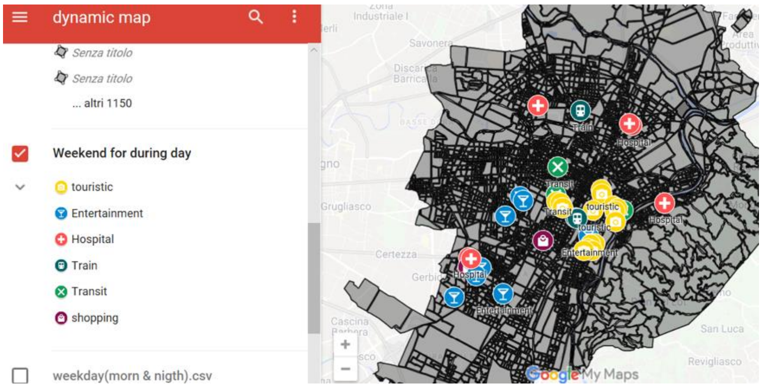

Figure 23. Weekend Daytime

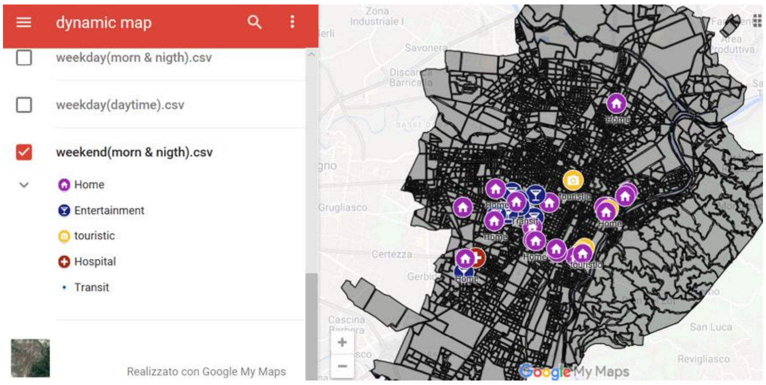

Figure 24. Weekend (Morning&Night)

4.2 Tracking map

The tracking map is created with Google Maps, where four layers are created importing four csv files, one for each day of the week with the related saved trips.

For each layer, trips are designed in this way:

- Yellow point: origin of the trip;
- Purple point: destination of the trip;
- Various symbols showing the possible purpose of the trip;
- Black lines: trip that connects origin and destination.

In *Fig. 25*, trips of Friday are displayed. From these trips we can induce the "home" of the user and the "academic" destination in Politecnico. On Saturday, instead, in *Fig. 26*, there is "home" again and a trip labelled as "entertainment".

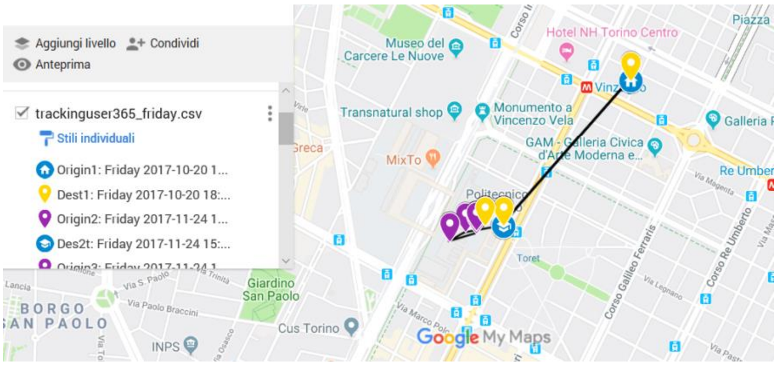

Figure 25. Tracking Friday

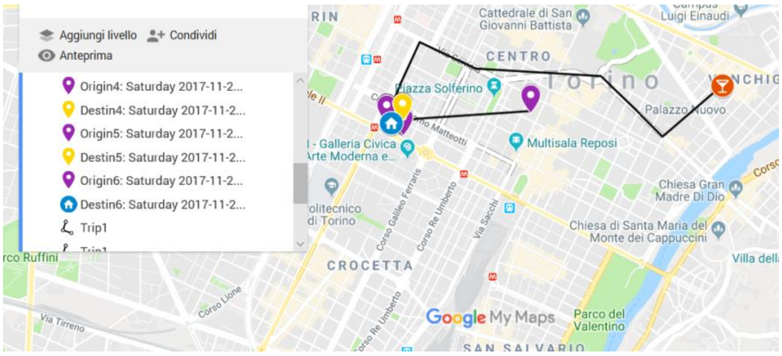

Figure 26. Tracking Saturday

What showed so far highlights how tracking is important to understand the mobility of the user, and ,mostly, to predict his/her travels and habits. Such a knowledge is very useful to improve mobility services and support the planning of future tranport systems.

In Fig.27, the total trips of the selected user in Turin are shown. Unfortunately, as already said, lack of data made difficulty to obtain an accurate prediction or better results.

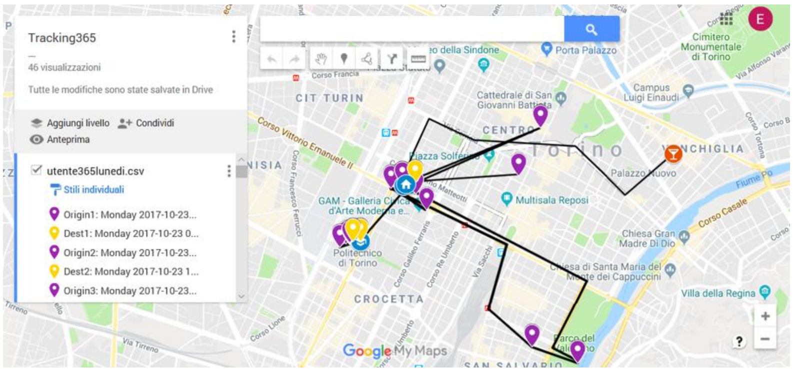

Figure 27. All days tracking

5. Conclusion

Based on the analysis and processing on the few data we had, we were able to label Torino zones and perform dynamic mapping. During the analysis, we faced some challenges:

- although there are many census zones (almost 4000) we only succeed to label 200 zones due to lack of data; indeed, most of the trips were concentrated in the centre of the analysed area so that it was not possible to label a zone without any data;
- having more data for training and testing part of the model, the results would have been more accurate, and the prediction value of machine-learning algorithm would have increased;
- the lack of POIs was another problem; more POIs for each zone would have allowed to obtain a better outcome in defining the purpose of the trips and, consequently, in labeling the zones.

We believe that this project, with more data, could greatly assist transport planning in order to decrease the amount of money spent for costly surveys to figure out users' travel habits in urban areas. However, a basic important challenge is finding a way that encourage people to install the app, enter their information and use it because this is the only solution to get more data and improve this project.

By the way, the photo presented in Fig. 28 is our invitation to all people who may read this report in the future.

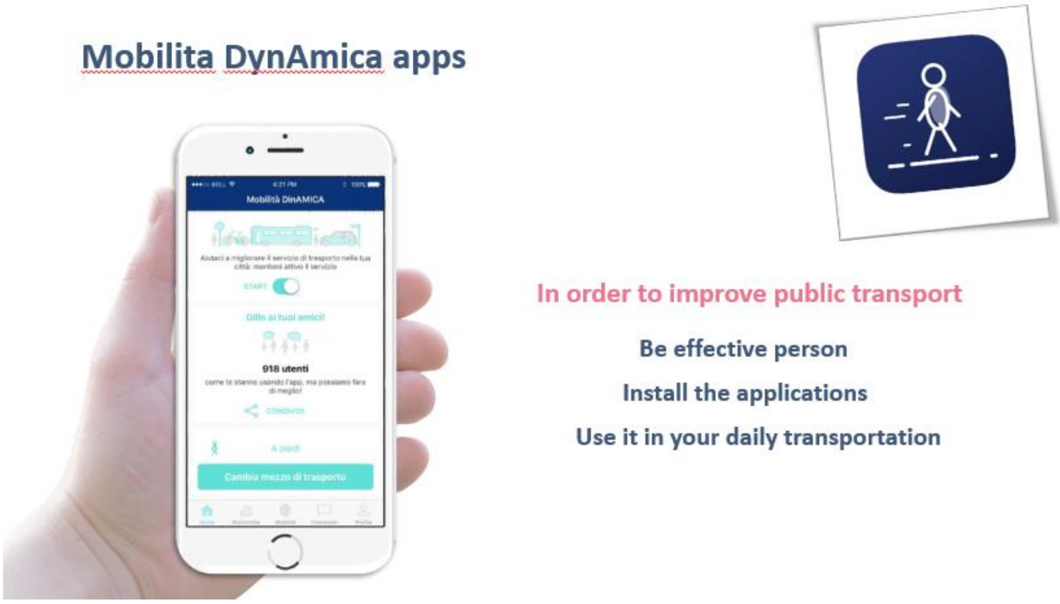

Figure 28. Mobilità dynAmica

6. References

[1] Küpper, A., Bareth, U. and Freese, B. (2011). Geofencing and Background Tracking – The Next Features in LBSs. [online] User.tu-berlin.de. Available at: https://www.user.tu-berlin.de/komm/CD/paper/010221.pdf [Accessed 9 Feb. 2019].

[2] Document has been prepared by the authors in the framework of a project funded by the European Commission, DG Research

[3] References and lectures has been sent by transport group in interdisciplinary project of year 2017-2018

[4] Woorti.com. (n.d.). Woorti - Woorti. [online] Available at: https://www.woorti.com [Accessed 7 Dec. 2018].

[5] Emission.open-to-all.com. (n.d.). [online] Available at: https://emission.open-to-all.com/#/home [Accessed 21 Nov. 2018].

[6] MotionTag. (n.d.). MotionTag. [online] Available at: https://motion-tag.com/en/developers/back-end-api-documentation.html [Accessed 30 Nov. 2018].