EcoTrip: User Transportation Mode inferencing and sustainability evaluation using Machine Learning

Federico Montini, Federica Perrone, Emanuele Tinghi, Veronica Torraca

f.montini1@studenti.unipi.it, f.perrone6@studenti.unipi.it, e.tinghi@studenti.unipi.it, v.torraca@studenti.unipi.it

ABSTRACT

In recent years, climate change issues are becoming increasingly impactful. In this context, much importance is attributed to pollution due to means of transportation.

The application developed collects data from the people's smartphones, then exploited to infer both the type of vehicle used (by means of machine learning algorithms) and the location (by means of the position obtained from the GPS).

The aim is to create an environment in which competition can motivate people to improve their travel habits, by comparing their own travel habits with those of people living in the same city (who should have access to the same modes of travel).

1 Introduction

The goal of this application is to find a way to persuade people to adopt more sustainable choices in their lives. Instead of simply telling people what to do and what to avoid, the application aims to find a more compelling way for people to improve their travels habits, and it proposes to do so presenting it to the user as a challenge against other users.

Specifically, to guarantee fairness, each one is compared with people belonging to the same zone and so to similar habits. In fact, depending on the size of the city in which the user lives, he/she will have different access to different modes of transportation. The geographic information is obtained through GPS data.

The length of the route is also analyzed to see the different transportation modes based on the distance traveled (e.g. if the distance covered is about 1 km, walking might be the best trade-off between time spent traveling and carbon emissions, whereas for longer distances it is unlikely).

To be able to infer the vehicle used, three sensors data are observed: gyroscope, accelerometer and magnetometer. Then those different data are periodically analyzed by a machine learning algorithm and a prediction is given amongst the classes that the classifier is able to recognize.

The user can then see how well she/he has performed on the result page.

Each of the possible transportation modes (bus, train, car, walking) has a different weight, expressing how much it is ecologically sustainable.

For each of the distance ranges (<1 Km, 1-5 Km, 5-10 Km, ≥10 Km), the user's data is analyzed. The weighted average of the trips of the user is calculated, depending on the transportation mode detected, and the result obtained is compared to the corresponding value, for that distance range and geographic zone, of the user belonging to the same city. If it expresses a value ecologically better than the mean of the others' ratings, the user, for that specific distance range, receives a positive feedback. Otherwise, it receives a negative one.

2 Architecture

The project is composed of five main components:

- Activities
- Location Managing
- Sensor Managing
- Classification Managing
- Database

2.1 Activities

In this application there are three activities.

The first one encountered is the *MainActivity*, where the user can access to the application via e-mail and password.

For the user authentication we opted for a solution based on *Firebase Authentication* [1]. In the login phase the user is required to enter its email and password. If already registered, the user can enter the application after email and password check returns a positive result. Otherwise, a new account is generated.

Once login is performed, the user gain access to the *RecordingActivity*, where a new journey to monitor can be started or the results obtained up to now can been shown, divided by the intervals of distance travelled.

If the latter is chosen, the *ResultActivity* is launched. In this activity, it is possible to have feedback about how well the user has performed in each possible range, referring to the last travel completed in that interval.

Instead, if the first option is chosen, the application starts to register travel data. The user is then able to stop the recording pressing the related button.

2.2 GPS Managing

When the user chooses to start a new trip, the system collects the start position of the user (latitude, longitude and city).

Once the start position has been obtained, the system continues to receive GPS update every time the user moves 5 meters. In this way, the system can calculate the distance between each GPS update.

In the end, when the user chooses to stop the trip recording, the system collects the final position and compute the total distance covered.

2.3 Sensors Managing

Each sensor is managed by a separated class, deriving the *SensorEventListener*, which takes care of registering and unregistering the corresponding sensor via a *SensorManager*, when the object is created and when the travel monitoring ends, respectively.

For all three types of sensors chosen, the collected data are in the following form:

- Value in the x-axis
- Value in the y-axis
- Value in the z-axis

Those samples are stored, independently for each sensors, in the *SensorsCollector* class, who is also in charge of extracting the metrics, of data collected every 5 seconds, considered for the classification, such as mean, min, max and standard deviation.

This data will be then used by the classifier to infer the mode of transportation.

2.4 Classification Managing

One of the biggest concerns that we had to deal with was the development of a classifier. After testing various machine learning algorithms and several tools on the dataset at our disposal [2] such as Random Forest, Decision Trees and Neural Networks it has been found that the best choice was J48 with AdaBoost, created with Weka [3] tool, and which returns the results in the table:

	Precision	Recall	F1-Score	ROC
AdaBoost	0.900	0.900	0.900	0.986

For the classification phase, a *ClassificationModule* class was implemented. This class takes care of managing the periodically computation of samples features, performed by the *SensorCollector* class, to be sent in input to the *TransportationModeClassifier* that performs the actual classification.

The result obtained was converted in human-readable format and sent back to the *RecordingActivity*, to be stored in the database to then process the rankings.

2.5 Database

For saving the data we opted for a solution based on *Firebase Firestore* [4].

The system accesses the database firstly in the login phase: if the user is not already registered, a new document in the users' collection is created. In this document, the system will store the results of user last trips.

The database is also accessed when the user stops the recording of the trip, to send to the database the mode of transportation detected and the range of the distance covered in that last trip. These results are stored in an aggregate manner in the users' collection, and also in a cities' collection, where the aggregate results for each single city are maintained.

Another case when the database is accessed is when the user requests to see its own results. A query is issued to retrieve the behavior in the <1 Km range (default setting when entering in this activity). If the user wants to see its rating in the other distance ranges, this can be done by pressing the corresponding section at the bottom of the screen, so the related query is sent to the database to retrieve the data associated to the selected range.

3 Experimental results and problems

The classification of the transportation mode is performed periodically every 5 seconds, but the final result is given by the mode classified the maximum number of times. This has been done to increase the likelihood that the trip is classified correctly.

Moreover, as a default case, also the "still" case has been considered, but it is returned as final result only if the counters of all other labels are zero.

Another problem encountered concerns the GPS. In fact, during tests, it turned out that the updates of the GPS are received only when the screen is active and the app is in foreground. Therefore, if it is running in background and is brought back to the foreground only at the end of the journey, the final distance obtained is approximate, since it is calculated as the crow flies.

3.1 Testing results

In order to leave the end user with a satisfactory user-experience, it is necessary for the application to be sufficiently accurate in making evaluations.

On the basis of the data obtained, we were able to build a classifier capable of classifying trips with a very high degree of accuracy, but as we have no direct control over the mode, variety and amount of data collected, it is necessary to test the application on real devices and contexts. Depending on the means available, various tests were carried out, in which the device was placed in various positions and surfaces in order to understand its effectiveness, at the end of this phase few considerations has been made:

- Accuracy drastically decreases when the user hold the device in his hand especially in car, train and bus
- Accuracy, while walking, do not change depending on where the device is positioned
- Misclassifications occur mostly between bus and machine

These findings are more than reasonable, the car and the bus are indeed the most similar means and, moreover, the dataset contained different amounts of data for each means, in particular the bus had less than half samples respect to the other labels. In order to balance the dataset, SMOTE was used to generate synthetic data through interpolation based on those already present. In the end, keeping the device in the hand or pocket, will significantly decrease the importance of the data recorded from the accelerometer, leading to recurrent misclassification.

3.2 Application Screenshot

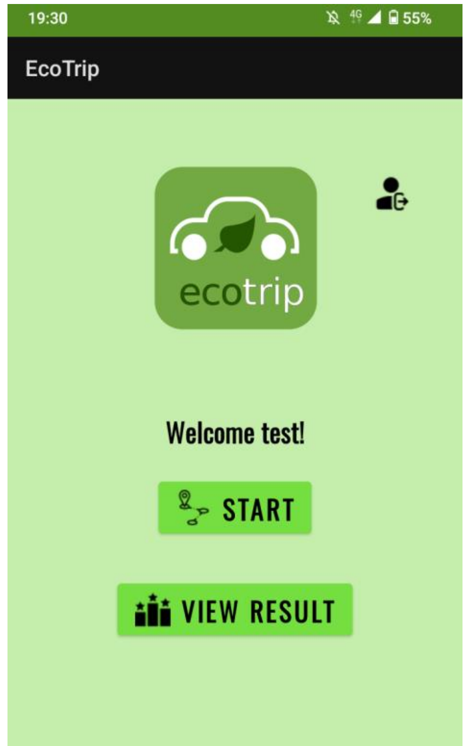

Figure 1: Recording Activity

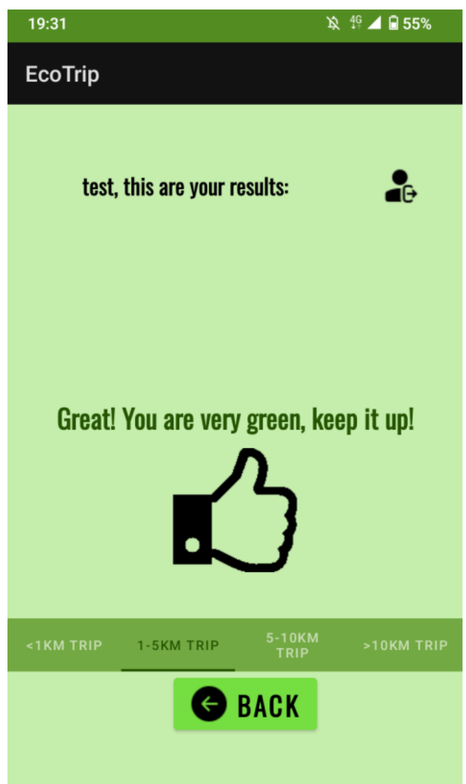

Figure 2: Result Activity

4 Conclusion and future works

To conclude, we can say that the application seems to be a good starting point for promoting sustainable transportation behavior among citizens.

For example, to incentivize ecological and sustainable choices, some other features can be implemented, such as a more detailed and various classification of trips, and a mechanism for earning points for each sustainable travel, always compared to the other users in the same city. To do so a different aggregation mechanism should be implemented, to consider a more general classification that includes all the trips a user made, for example, in the last month, or also considering the different cities in which the user traveled.

REFERENCES

[1] https://firebase.google.com/docs/auth
[2] http://cs.unibo.it/projects/us-tm2017/index.html
[3] https://www.cs.waikato.ac.nz/ml/weka/
[4] https://firebase.google.com/