Course-accompanying project: Development of a personal energy and carbon footprint mobility app

Start:	October 2, 2019
End:	December 3, 2019

Objectives

The aim of this project is to be confronted with a typical situation that requires system design and development skills from the computing area to address a question related to smart energy and sustainability. Concretely, course participants are requested to develop a personal energy and carbon footprint mobility calculator as an Android application.

The application has to continuously run in the background during the day and infer the transportation mode (i.e., on foot, bike, e-bike, motorcycle, car, tram, bus, train) from the user's context. It should subsequently derive the energy used and the carbon emitted that can be attributed to the user for her or his mobility needs on a daily basis, and store and present aggregates.

Deliverables

The project will be conducted in teams of three course participants. Teams start to be formed within the second week of the course. As around 60% of attendees are computer science students, we ask teams to be formed in such way that each one includes at least one computer science student. By Wednesday, October 2, the teams must be formed; one of the team members submits the names of the three members via email to Vlad Coroamă (vcoroama@ethz.ch), cc'ing the others on the team.

The project has three deliverables, required by 11:59 p.m., 3 December 2019:

1. **source code**: The final project must be submitted via email (as an entire Android project, with build files removed, zipped) to Vlad Coroamă (vcoroama@ethz.ch). The code should be well structured and documented.

2. **report**: A short report that summarizes the project should be submitted along with it. The text, together with supporting graphs and screenshots, should highlight important aspects of the app, e.g., its architecture and logic, the model used for transportation mode classification, achieved accuracy, main encountered challenges and deployed solutions. It should also present the main elements of the developed graphical user interface (GUI). The text must be short: 2 pages of text plus an appendix including as many figures as necessary.

3. **presentation**: Two weeks later, during the last lecture on December 18, each team will have 7 minutes to present their app to the entire course. The presentation should focus on similar aspects as the report. Screenshots of the app should also be included. A short emulator-based demo or a video showing the app at work are encouraged, but not required.

Organizational aspects

As this assignment will last for several weeks, we kindly ask course participants to decide early-on whether they want to continue attending the course. Giving up the course in the middle of the project will bring a disadvantage to your team (one we will be able to handle by e.g. reducing the requirements for that team, but this will imply effort on all sides and a less satisfying final result for the rest of your team).

The project is meant to simulate a real-life work assignment, in which your team receives the task of developing such an application for a customer. We thus do not specify all the details; in particular we do not recommend sources of energy and carbon intensities for different transportation modes. Some requirements engineering on top of the general requirements, finding data sources, and the graphical user interface (GUI) are explicit parts of the assignment.

The transportation mode classification can be implemented rule-based or via machine learning. To ease either of these tasks and help students focus on the logic, not on Android programming, we provide a framework that fulfills different functions: i) it handles the Android sensor reading with frequencies that can be redefined, ii) delivers the sensor results in windows of a length that can be determined, iii) presents simple examples of feature extraction from such windows, iv) comprises three simple examples of the Python-based training of machine learning models (XGBoost, MLP, and CNN, respectively), v) shows how to export these models to Android, and vi) how to show their results in a simplistic GUI.

For data analysis and possible model training, we also provide a data collection including 234 *trip legs* (i.e., segments of a trip in which a single transportation mode was used) distributed over all 8 transportation modes and totalling some 36.5 hours. The dataset comprises the user-generated ground truth (i.e., the used transportation mode) together with data recorded from several sensors: GPS, inertial measurement units (IMUs) – i.e., accelerometer, gyroscope, and magnetometer – as well as regular Bluetooth (BT) and WiFi scans. For the privacy of third parties in the environment, the MAC addresses of BT and WiFi devices in the environment have been hashed. As the same MAC address is hashed to the same value for the entire trip, they can still be used for pattern recognition.

While the rule formulation, or the learning, respectively, can take place offline using training data, the final app must run online on the Android smartphone. Google's Activity Recognition API cannot be used as part of the solution, but can be consulted for benchmarking.

For the assignment you can use your own Android smartphone(s), if available; no rooting is required. If needed, we can provide one smartphone per team for the duration of the project.

Grading and minimum requirements

For grading this project, all three deliverables will be taken into account. A general and non-binding orientation on grade requirements for the app itself is listed below:

- **4.0**: a relatively simple (yet not simplistic) automatic transportation mode classification was developed, in which some of the more similar modes were possibly taken together, such as bus and tram or bike and e-bike. The rule definition, feature design or neural networks are of medium complexity, and there is a simple evaluation of the results. The user has to define when a trip starts and ends. A simple GUI presents the user's daily energy and carbon footprints at the end of the day.
- **6.0**: an elaborate classification among all transportation modes was developed after a far-reaching data analysis. The fairly complex rules, feature design or neural networks, respectively, were evaluated in an intricate way, highlighting the strengths and the weaknesses of the developed classification. An appealing GUI presents, besides the daily footprints, also weekly or monthly aggregates.
- **bonus points**: bonus points can improve the grade for this project by 0.25-0.5, if not all requirements for a 6.0 have been reached. For bonus points, the team needs to impress the audience with innovative, out-of-the-box ideas, during the presentation on December 18. As examples for such feature, the user could be congratulated when undertaking a similar ride to one of the previous days but with a less energy- and carbon-intensive means of transport, or the start and end of a trip could be automatically inferred.