

# Status Report #0

Nicolà Lohr

June 22, 2019

## Abstract

Completed so far:

- Basic knowledge of L<sup>A</sup>T<sub>E</sub>X
- This Short Status Report
- Decision on the scope and the time commitment: "KSZ great"
- Book selection: Data Science from Scratch (9781491901427), Data Algorithms (9781491906187)

## 1 Introduction

The first proof of concept indicates that the original aims are achievable: I am able to link every MAC-address to a class or teacher at least. Finding close friends works as well. I also learnt the basics of L<sup>A</sup>T<sub>E</sub>X.

## 2 Details

### 2.1 L<sup>A</sup>T<sub>E</sub>X

I learnt the basic of L<sup>A</sup>T<sub>E</sub>X for the final paper. I'm writing/wrote this report also in L<sup>A</sup>T<sub>E</sub>X to improve my skills.

### 2.2 Time-Commitment

Because of the limited time in the year of 2019 I chose to focus on making a great high school project and paper and not on making an "ETH great" paper.

### 2.3 Books

I selected the two books I am going to read over the summer holidays. I used Safari Books for evaluating as many books as possible to find the best for my needs. I choose Data Science from Scratch and Data Algorithms. Both of these books should give me more inside in useful algorithms for my next steps. The first book should also help to improve my coding skills. Data preprocessing and graphics are covered as well.

### 2.4 Proof of Concept

#### 2.4.1 Basic Tracking

The basic tracking gets the class a user is in only from their MAC address. First I parse the data from the time table into a dictionary so it is easy to get the class from the weekday, room and time. I stored all the log-files from the IT-department in one folder. Next step is to get all this data into a nice and useful format. It is using tuples with the format: (time, user-mac, ap-place). Afterwards, the program filters the data to get rid the part of the data which

have nothing to do with the target. With less data, it's just a comparison between the data and the time-table and tries to find as many matches as possible. The class with the most matches will then be presented as a result.

#### **2.4.2 Reverse Tracking**

Reverse tracking is basically the tracking for all MAC addresses which are found in the log files. That the program doesn't have to run over a thousand times (for every MAC address once) there is a function which takes care of it. Simplified the program doesn't use a filter and goes through all the data. As a result, I get a dictionary with every MAC address and their potential class.

#### **2.4.3 Friends Tracking**

As a side project, I tried to find out who is friends with whom. I looked which people walked together so that their phones switched the access-point at about the same time ( 5 sec). I mostly found friends who are in the same class (walk from one classroom to the next) but also had some success with students in different classes.

#### **2.4.4 Optimisation**

To optimise my program speed I parsed the data as I said above, so I had to look at the files just once. I also used multiprocessing (running part of the program in parallel) to improve speed. Running multiple times the same program in the background came with not much gain of speed and the accuracy dropped a little. For the final version, I most likely will not have the multiprocessing as a standard.

### **2.5 Next Steps**

For the near future, I planned to read the selected books, start writing the paper, especially the results I already got, and research for facts about this tracking which I can use in the paper.

### 3 Brainstorming

Here is the brainstorming from one of the meetings with Wiwi:

