**Description**

For our project, we would like to use the data set '[Crowdedness at the Campus Gym](https://www.kaggle.com/nsrose7224/crowdedness-at-the-campus-gym)' from Kaggle. As the name suggests, the data set includes over 25,000 people counts from a campus gym in the United States. We have chosen this particular data set because on the one hand, it is large enough to perform some interesting measurements, but on the other hand it is not as abstract as some of the other datasets we have worked with. This particular one concerns a situation we are all familiar with: can I use the crowdedness of the campus gym as an excuse not to go?

**Features and data types**

The features in the data set are:

* Date (string)
* Timestamp (integer)
* Day of the week (integer, 0 - 6)
* Weekend? (integer, 0 or 1)
* Holiday? (integer, 0 or 1)
* Temperature (float, degrees Fahrenheit)
* Start of semester? (integer, 0 or 1)
* Month (integer, 1-12)
* Hour (integer, 0-23

**Goal**

Our goal will be to train our classifier in such a way, that we can accurately predict the crowdedness under particular circumstances. We will evaluate what features are relevant predictors, and which ones could maybe be left out. Intuitively, it seems like using a decision tree may be a good algorithm to start with. Moreover, a decision tree has the feature that it can calculate the entropy which will tell us which features are most important to take into consideration when prediction when someone would go to the gym. In addition, we will select some other algorithms, and compare the performance of the data on those. A good metric to compare the data sets will be the f1-score of the different algorithms.

**Type of classification**

We will most likely not use binary classification. Instead, we will assign '1' to the maximum value of people in the gym. Then, for each test example, our algorithm will return a value between 0 and 1, which represents a fraction of the busiest moment. If we were to use a binary classification, we could take some value x, representing a threshold of crowdedness. If predicted crowdedness < x, our algorithm predicts '1' (yes; I will go to the gym). Otherwise, predicted crowdedness > x, and our algorithm will predict '0' (no, it is too busy; I will not go to the gym).