For this assignment, we are given the task of recreating the analysis done by Chowdhury et al. (2017) on the classification of physical activities through ensemble methods. The data in question covers nine individuals who wore special bands (called IMUs) to track their physical data. The goal of the study was to investigate how both sedentary and physical activity are related throughout the day, as well as to investigate whether the use of ensemble learning algorithms improve physical activity recognition accuracy compared to the single classifiers. The ensemble method is carried out in the corresponding Jupyter notebook.

Data on 54 features is collected, with many of these directly related to each other. For example, the positions and temperature of the hand, ankle, and chest were collected throughout the day but each is listed as its own column. There are timestamps, heart rates, activity ID’s and data from the IMUs. These bands contained information on temperature, acceleration, gyroscope and magnetometers. The Activity ID is treated as the classification target in this problem, with the 24 classes being the 24 activities that were prescribed. The nine individuals included one woman and eight men, with average age of about 27 and BMI of 25.

Before we can train our classifiers, we must perform some preprocessing. We start by defining a map for the target, then define some functions to generate comprehensive columns for the IMU data. We then load all 9 subjects in (through a for loop) and combine them into one master set. We first notice that we have many missing values, and we also see that each subject therefore has differing amounts of information (with subject #9 in particular having lost a lot of information). Additionally, each data point is collected sequentially, and thus we will need to treat the data as though it were a time series model.

The paper by Chowdhury et al. addresses the NaN problem through linear interpolation, so we will do the same. We also follow their step of treating the data as time series, and we drop all values with the null (undefined in the map) activity. After conversion, and splitting into X,y dataframes, we are left with an input matrix of size 1,942,872 x 54. Because of the nature of the time series data and the fact that we technically have nine separate data matrices (one for each subject), we will take advantage of this when deciding how to split into training and testing sets. Rather than taking a random sample from the entire master set, we select two arbitrary subjects (which turned out to be #4 and #7) and let them be the testing subset, and everything else is the training set.

Now recall the theoretical definition of the ensemble. In a N-classifier ensemble, let the classifier set and set of classes be defined as  respectively. Each classifier  produces a class label  without any further information. When classifying an object , the N classifier outputs a vector  Then, the decision fusion techniques combine the classifier’s output and provide a single class label. Put more simply, each classifier “votes” to assign a record to a particular class. The final decision can be made several ways, including averaging the results, hard voting (sort of like paper ballots) or

Familiarize yourself with the ensemble package in Python and its use in a Jupyter notebook by utilizing the “Ensemble Methods for Classification of Physical Activities” to complete this assignment.

Note the additional digital resources in Supplemental Digital Content at the bottom of the article. Also, note that the dataset used in the article is available for download from the UCI repository and the direct link is included in the article in the Methods section, Data set 1.

Another useful resource is “A Comprehensive Guide to Ensemble Learning (with Python codes).”

Once you have reviewed the required resources, complete the following:

Follow the steps described in the article for acquiring the data and building a classifier (in Python) that implements an ensemble framework.

It is expected that you will encounter obstacles along the way and not every step mentioned in the article will be straightforward to implement.

Ideally, you will be able to reproduce the project in its entirety.

Less than ideal, but still very useful, would be to attempt most steps, adapt some, maybe eliminate one or two classification methods from the ensemble, but still produce a working classifier.

Given the breadth and depth of the projects you worked on in this course and given the detailed resources provided that cover both theory and implementation, you are expected to successfully complete this project.

References:

Chowdhury, A. K., Tjondronegroro, D., Chandran, V., & Trost, S. G. (2017). Ensemble Methods for Classification of Physical Activities from Wrist Accelerometry. *Medicine & Science in Sports & Exercise*, *49*(9), 1965–1973. https://doi.org/10.1249/mss.0000000000001291

<https://towardsdatascience.com/ensemble-learning-using-scikit-learn-85c4531ff86a>

https://machinelearningmastery.com/random-forest-ensemble-in-python/