Kristianstad University
www.hkr.se

Fredrik Frisk
Computer Science Department

Content and purpose of this lab

During the lab we will start acquiring some data for late machine learning experiments. some machine learning experiments using your own recorded data. We will work with our own recorded data. We will start with a smaller recording to just get used to the needed software.

We will use pandas in this lab, mostly due to the plotting functions and the handling of csv-files.

Part one will focus on record data from sensors, import the data as pandas objects, actually dataframe. We will also do some mathematical operations on the objects as well as plot them.

Save the code you are writing in this lab for future use. To pass the lab you need to solve/program the different bullet points and be able to explain your results. If you are not finished with the all the bullet points the remaining ones are a part of the required preparations for part 2 of the labs. The lab report in the end is a an individual report, but you are allowed to work two and two with one exception all of you have to record your own sensor data.

Preparations

Work through the tools_pandas.ipynb notebook. You need to have an understanding of the following items before the lab:

- Panda Series
- Panda Dataframe
- .loc
- .iloc

That means that you have to know what these items are and how to use them (in some respects). This notebook also refers to dictionaries. You need to know this python object as well, some information you can find here:

https://www.w3schools.com/python/python_dictionaries.asp

Acquire some data

We will start in an easy pace. Record accelerometer data and magnetometer data, all three directions, for two different positions,

- standing and
- lying down.

Put the phone in your pocket while you do the recording. Transfer the data so it is accessible in your python environment. Each recording should be at least 5 s.

Needed imports:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

The dataset is imported using the panda method shown below:

```
dataset_acc = pd.read_csv(filename, usecols=[1,2,3],
names=['ax', 'ay', 'az'], header=None)
```

- Plot the accelerometer values using the the plot functions accessible for panda dataframes. Plot all accelerometer values, i.e. all samples for all components of the accelerometer vector.
- What are we measuring with the accelerometer?

As you may have noticed the scale on the x-axis is the index of the rows in the dataframe.

- There is time information both in the csv-file and in the setup in the app. What is the sample time of the accelerometer data? Sample time is the timdifference between two consecutive samples? What is the sampe frequency? The sample frequency is the inverse of the sample time.
- By looking at the plots can you see the difference between the two classes? Explain, and take some notes for future work.

- Work through the above four bullet points for the magnetometer values as well.

You can easily plot a dataframe by using the method .plot()

Some Vector algebra

Both the accelerometer and the magnetometer sensors are acquiring sensor data in three directions x, y, z.

- Calculate the angle between the acceleration values and the z-direction for all samples, store it in a dataframe and plot it. Do this for both classes.
- Calculate the angle between the magnetometer values and the y-direction for all samples, store it in a dataframe and plot it. Do this for both classes.
- Calculate the magnitude (length,norm) for the accelerometer data for all samples, store and plot.
- Calculate the magnitude (length,norm) for the magnetometer data for all samples, store and plot.

Python

We need more data! Add a new class, sitting. Record at least three recordings for each class. You should have at least one for lying down and standing from the beginning of the lab. Each recording should be at least 5 s. Now you should have:

- Stand : 3 files with accelerometer data and 3 files with magnetometer data.
- Sit down: 3 + 3 as well
- Ly down: 3 + 3 here also

Divide the data into two sets, training set and test set. Store 2 of 3 files in the training folder and 1 of 3 files in the test folder. Name the files so you can identify which class and which sensor data it contains.

- Create a python function that can read all files from a class for both sensors. The function should return a dataframe *x* with all the data and another dataframe *y* with information of which class the data belongs to.

Here you need both `.concat` and `.append` dataframe methods. Also be careful to take into account the sampling frequencies. Are they the same for accelerometer and magnetometer data?

- Create a function that returns all the training data and the respective classes for the whole set of training data.

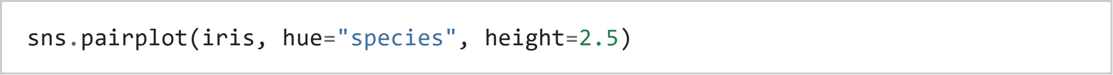
- Create a function that returns all the test data and the respective classes

Plots

One useful plot is the scatterplot. We will be using the seaborn package to make scatterplots.

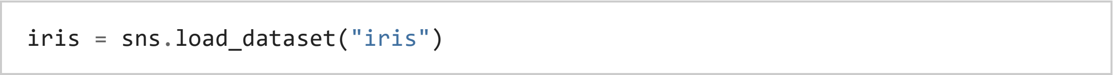
https://seaborn.pydata.org/tutorial/axis_grids.html#plotting-pairwise-data-relationships

At the end of this page you will find how pairplot is used:



```
sns.pairplot(iris, hue="species", height=2.5)
```

Note you need to load the iris dataset first:



```
iris = sns.load_dataset("iris")
```

You also need as stated previously import the seaborn package.

- Plot a `pairplot` for all features, 3 accelerometer values, and 3 magnetometer values. Explain what you see.

- Plot a `pairplot` of the features magnitude of the accelerometer and magnetometer data as well as the two angles defined previously. Explain what you see.