Chapter 2
Basics of Sensory Data

Before we discuss some of the details of the various machine learning approaches, we will focus on the topic of sensory data itself. Since rapid technical advances are being made in this area, we will refrain from explaining the workings of each potentially useful sensor out there. Rather, we will dive into a representative dataset used throughout the book. The dataset originates from *crowdsignals*[1] which has generously been made available for experimentation for us as authors and for you as reader of our book. The dataset has been collected using an application that gathers data from both a smartphone and a smart watch. In addition, users were asked to label the activities they were conducting (e.g. "I am currently running"). We will first describe the measurements included in the dataset. We will then show how to move from the raw data we collect to a dataset usable in machine learning tasks. This process is described in the context of the crowdsignals dataset but is representative for most of the sensory datasets we have worked with. Finally, we explore the resulting dataset and identify suitable machine learning tasks.

2.1 Crowdsignals Dataset

An overview of the sensory data in the crowdsignals dataset is shown in Table 2.1. In the table, we focus on the *sensors* and *user labels* categories, for the others, please explore the full crowdsignals dataset description, which is available via the aforementioned website. We were not able to include all sensor and user label measurements in the experiments we present in this book. Those that have been included are marked with a "yes" in the last column.

[1] http://www.crowdsignals.io.

M. Hoogendoorn and B. Funk, *Machine Learning for the Quantified Self*,
Cognitive Systems Monographs 35, https://doi.org/10.1007/978-3-319-66308-1_2

Table 2.1 Sensors and labels in crowdsignals dataset

Sensor	Purpose	Device(s)	Values	Time point / Interval	Used
Sensors					
Accelerometer	The acceleration of the device	phone/ watch	x, y, and z acceleration	time point	yes
Gyroscope	The angular speed of the device	phone/ watch	x, y, and z angular speed	time point	yes
Magnetometer	The magnetometer value of the device	phone/ watch	x, y, and z magnetometer value	time point	yes
Heart rate	The heart rate of the user	watch	heart rate (beats per minute)	time point	yes
Temperature	Ambient temperature	phone/ watch	temperature (in ^{o}C)	time point	no
Light	The light intensity	phone/ watch	light intensity (in lux)	time point	yes
Pressure	The current pressure	phone/ watch	pressure (in mercury millibars)	time point	yes
Humidity	The current humidity	phone/ watch	relative humidity (%)	time point	no
Proximity	Distance of user from phone	phone	distance (meters)	time point	no
Audio record	Record of audio obtained via the microphone	phone	audio recording	time point	no
User labels					
Activity label	Record of the activity a user is conducting	phone	label (walking, running, ...)	interval	yes

A huge variety of sensors exist. Three popular sensors do dominate the landscape of smartphone sensors and are also included in our dataset: the accelerometer, magnetometer, and gyroscope. The *accelerometer* measures the changes in forces upon the phone on the x, y, z-plane. The orientation of the phone compared to the "down" direction (the earth's surface) and the angular velocity are measured by means of the *gyroscope* (measured on the same three axes as the accelerometer does). Finally, the *magnetometer* measures the x-, y-, and z-orientation relative to the earth's magnetic field. Micro-electromechanical systems (MEMS) form the technical basis of these sensors. MEMS employ the effect that the resistance of semiconductors is stress-sensitive, or put in other words, changes when mechanical forces are applied—this phenomenon discovered in the 1950s is called piezoresistance and the basis of a large industry today [22].

Table 2.2 Snapshot heart rate data

Sensor_type	Device_type	Timestamps	Rate
heartrate	smartwatch	1454956086325639687	175
heartrate	smartwatch	1454956086684549167	176
heartrate	smartwatch	1454956087523516770	175

Table 2.3 Snapshot label data

Sensor_type	Device_type	Label	Label_start	Label_end
interval_label	smartphone	On Table	1454956132985999872	1454956366574000128
interval_label	smartphone	On Table	1454956393088000000	1454956578385999872
interval_label	smartphone	On Table	1454956608515000064	1454956813323000064
interval_label	smartphone	Sitting	1454956894057999872	1454957092968000000

Of course, there are many more sensors used in today's smartphones that you are familiar with. Just think of a GPS signal that measures your position by means of your distance to a number of satellites of which the position is known. For a full overview of sensors, we refer the reader to books dedicated to modern sensors, for example [48].

Let us have a look at how the data has been recorded. All data is stored with a reference to when the data was measured. Some recordings cover measurements for a certain period or interval while others are only valid for a specific point in time. For example, the heart rate is measured for a specific time point while the label provided by the user is specified for an interval (I was walking between time point t and time point t'). In Table 2.2, we can see a snapshot of the heart rate data, whereas an example for the label data is shown in Table 2.3. Time points are expressed in nanoseconds since the start of time (which is January 1st 1970 following the UNIX convention).

We are still far away from the specification of a dataset we have seen in Chap. 1, where $\mathbf{X}^{\mathcal{T}}$ denotes a matrix with rows representing the measurements of an individual time point (if the dataset has a temporal nature, which we clearly have here). Next, we will show how we move from our current dataset to the desired matrix format.

2.2 Converting the Raw Data to an Aggregated Data Format

In order to convert the temporal data, we first need to determine the time step size we are going to use in our dataset. This is also referred to as the level of granularity (selecting a Δt). We could say that we want to have instances covering a second of data for example, or even a minute. The selection of the step size depends on a

variety of factors, including the task, the noise level, the available memory and cost of storage, the available computational resources for the machine learning process, etcetera. Once we have selected this step size we can create an empty dataset.

We start with the earliest time point observed in our crowdsignals measurements and generate a first row $x_{t_{start}}$. Iteratively, we create additional rows for the following time steps by taking the previous time step and adding our step size, e.g. $x_{t_{start}+\Delta t}$. Each row x_t represents a summary of the values encountered in the interval defined by the time step it was created for until the next time step, i.e. $[t, t+\Delta t)$. We continue until we have reached the last time step in our dataset. Next, we should identify the columns in our dataset (our attributes) that we want to aggregate. As we have seen, we can distinguish between numerical values (e.g. the heart rate) and categorical values (e.g. the labels) and need different approaches for both. For the former, we create a single column for each variable we measure while for the categorical values we create a separate column for each possible value. Of course, for the categorical attributes we could also include a single column where each row would contain a single value for that measurement. However, since we are discretising time steps it is very likely that we will encounter multiple values for our categorical measurement per time step (e.g. the user performing the activity driving and walking within the same time step). We cannot accommodate for this if we can just insert a single value: which one should we select?

Once we have defined the entire empty dataset, we are ready to derive the values for each attribute at each discrete time step (i.e. each row). We select the measurements in our crowdsignals data that belong to the specific discrete time step (when either the associated time stamp falls in the window, or the interval expressed falls (partly) within it) and aggregate the relevant values. We can aggregate numerical values by *averaging* the relevant measurements (e.g. for heart rate) or we can *sum* them up (e.g. when the measurements concern a quantity) or use other descriptive metrics from statistics such as median or variance. Since often it is not clear a priori which type of aggregation to choose, you could also use different measures and later let machine learning techniques select relevant features. For categorical values we can count whether at least one measurements of that value has been found in the interval (*binary*) or we can count the number of measurements that have been found for the value (*sum*).

In our case we have selected the averaging method for numerical values and the binary method for categorical attributes. When taking a Δt of 1 day and aggregating the data we have seen in Tables 2.2 and 2.3 we would end up with the table shown in Table 2.4. As mentioned before, all these approaches have been implemented and are available on the website accompanying the book, including the code used to process the crowdsignals dataset.

Table 2.4 Example resulting dataset

Time	Heart_rate	Label On Table	Label Sitting
2016-02-08 19:28:06	175.333	1	1
2016-02-09 19:28:06	-	0	0

2.3 Exploring the Dataset

Let us consider the entire dataset with the sensors we have marked as "yes" in Table 2.1. We have a set of measurements that covers approximately two hours of labeled data of a participant. If we take a granularity of 1 minute, we obtain a dataset that is shown in Fig. 2.1. The dataset contains 133 instances (i.e. 133 minutes). We can see that we have quite a nice dataset, although the data does seem a bit too smooth, especially regarding the accelerometer, gyroscope, and the magnetometer

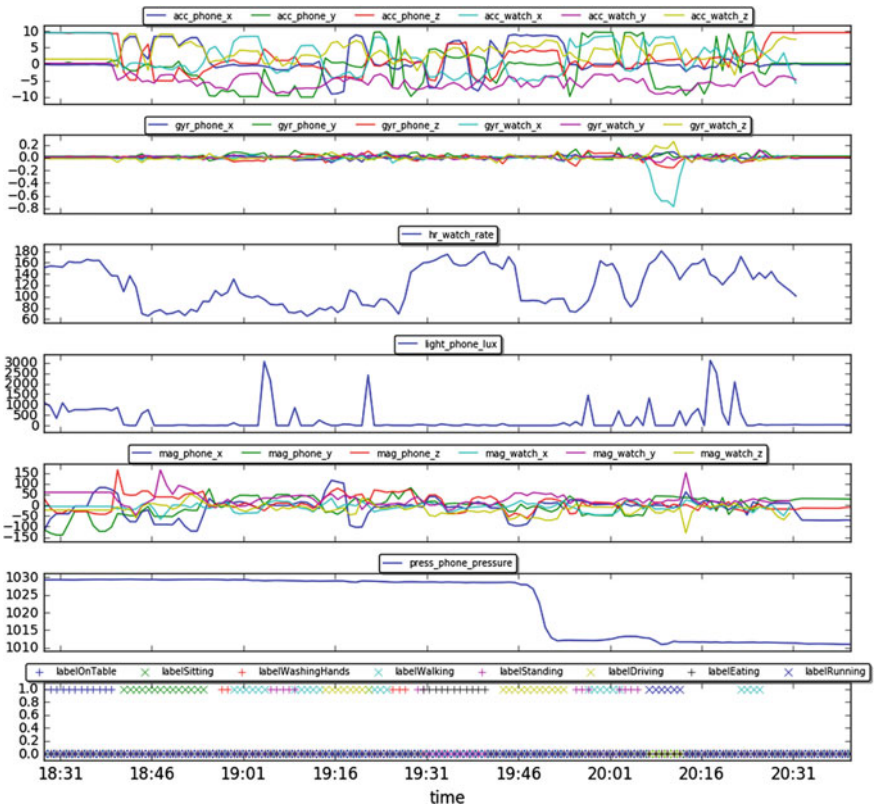


Fig. 2.1 Processed CrowdSignals data ($\Delta t = 60\,\text{s}$)

data. To be more specific, we know that walking should provide us with some periodic changes in the accelerometer data (usually with a frequency in the order of 1Hz) but this information is lost as a result of the aggregation. If we consider a more fine grained dataset with $\Delta t = 0.25\,\text{s}$, i.e. four instances per second, we are likely to capture the stepping motion. The result is shown in Fig. 2.2 and contains a total of 31838 data points. Indeed we see a lot more variance in this data. Previously, we had just aggregated too much and lost the fine details in our dataset that might be of great value. The choice of Δt highly depends on the task. For example, if you want to determine the step frequency of a person, your Δt should be significantly smaller than the corresponding step period. On the other, if you want to learn about the motion state of a person, e.g. walking or sitting, $\Delta t = 1$ minute might not only be sufficient but also optimal with respect to the predictive capabilities of a model based on the aggregated data.

We have created some summary statistics of the two datasets with different Δt in Table 2.5 to signify the differences. In addition, Fig. 2.3 shows the differences of the accelerometer data in a boxplot. We see that the extreme values and standard deviation show substantial differences. We observe higher standard deviation and

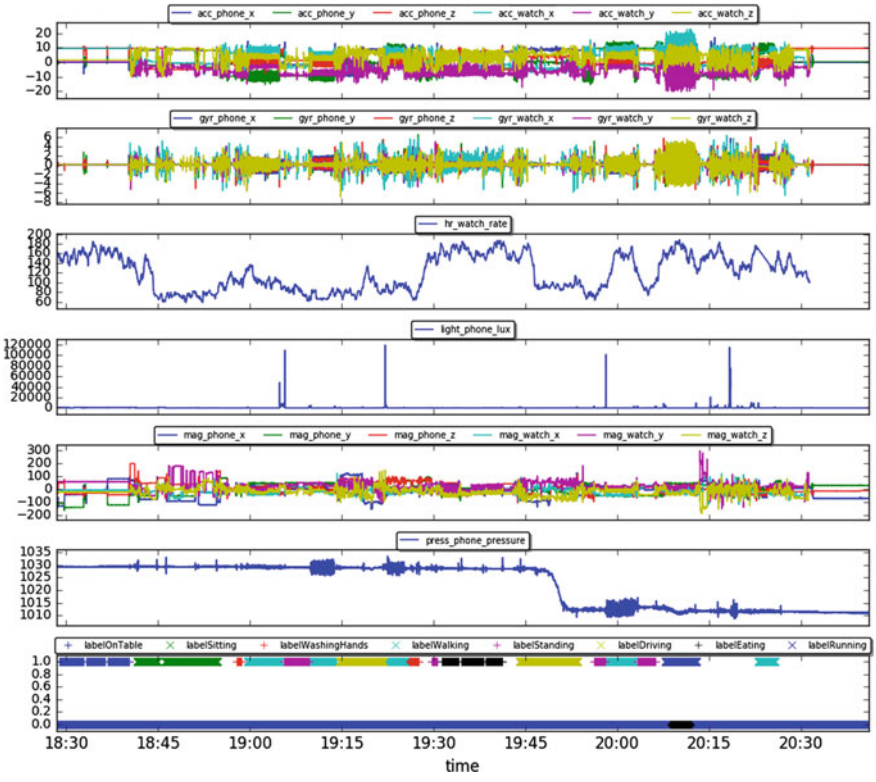


Fig. 2.2 Processed CrowdSignals data ($\Delta t = 0.25\,\text{s}$)

Table 2.5 Statistics of processed dataset (first number listed is for $\Delta t = 60\,\text{s}$, second value for $\Delta t = 0.25\,\text{s}$)

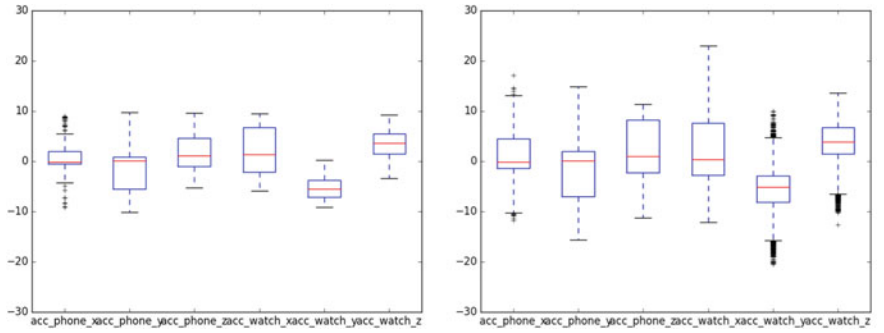
Numerical										
Attribute	Missing (%)		Mean		Standard deviation		Minimum		Maximum	
acc_phone_x	0.0	0.0	1.1	1.1	4.2	4.7	−9.1	−11.8	9.0	17.1
acc_phone_y	0.0	0.0	−0.9	−0.9	5.6	6.4	−10.1	−15.6	9.8	14.9
acc_phone_z	0.0	0.0	2.0	2.0	4.7	5.4	−5.3	−11.3	9.6	11.4
acc_watch_x	7.5	8.8	2.0	2.1	4.9	5.8	−5.8	−12.2	9.6	22.9
acc_watch_y	7.5	8.8	−5.2	−5.2	2.4	3.5	−9.1	−20.6	0.2	10.0
acc_watch_z	7.5	8.8	3.6	3.6	2.7	4.0	−3.4	−12.6	9.2	13.7
gyr_phone_x	0.0	0.0	0.0	0.0	0.0	0.6	−0.1	−4.0	0.1	5.7
gyr_phone_y	0.0	0.0	0.0	0.0	0.0	0.4	−0.1	−5.0	0.2	6.5
gyr_phone_z	0.0	0.0	0.0	0.0	0.0	0.5	−0.2	−5.4	0.1	5.9
gyr_watch_x	8.3	8.9	0.0	0.0	0.1	0.7	−0.8	−6.7	0.1	6.3
gyr_watch_y	8.3	8.9	0.0	0.0	0.0	0.6	−0.1	−5.5	0.1	5.0
gyr_watch_z	8.3	8.9	0.0	0.0	0.0	0.8	−0.1	−7.0	0.3	5.5
hr_watch_rate	7.5	76.4	119.2	121.0	35.5	35.2	65.4	58.0	180.7	188.0
light_phone _lux	0.0	10.4	278.34	281.5	596.3	2220.9	0.0	0.0	3109.3	118985.0
mag_phone_x	0.0	0.0	−13.7	−13.5	46.9	50.6	−121.8	−156.4	115.5	126.6
mag_phone_y	0.0	0.0	−3.7	−3.8	44.9	47.9	−139.7	−165.4	80.7	96.8

(continued)

Table 2.5 (continued)

Numerical										
Attribute	Missing (%)		Mean		Standard deviation		Minimum		Maximum	
mag_phone_z	0.0	0.0	7.5	7.6	35.2	40.0	−61.2	−106.4	164.1	198.0
mag_watch_x	8.3	8.9	−9.2	−9.1	17.7	26.1	−66.0	−138.0	31.7	122.8
mag_watch_y	8.3	8.9	27.2	27.3	29.7	39.6	−47.6	−151.3	163.6	297.4
mag_watch_z	8.3	8.9	−20.0	−20.0	24.2	31.6	−130.3	−186.7	51.4	149.7
press_phone_pressure	0.0	10.3	1022.3	1022.3	8.3	8.3	1011.0	1008.6	1029.4	1033.5

Categorical			
Attribute	Value	Percentage of cases (%)	
label	OnTable	9.0	7.8
label	Sitting	10.5	8.6
label	WashingHands	3.8	2.0
label	Walking	18.8	14.7
label	Standing	10.5	7.3
label	Driving	14.3	12.4
label	Eating	8.3	6.8
label	Running	4.5	3.8



(a) $\Delta t = 60$ seconds (b) $\Delta t = 0.25$ seconds

Fig. 2.3 Boxplots of all accelerometer data

more extreme values for the more fine grained dataset, which is to be expected given our averaging approach to compute the values for a specific discrete time step. This is also reflected in the percentage of data points associated with each of the labels. In terms of missing values we do not see many differences for the numerical values, except for the heart rate. It seems that the sampling rate of the heart rate values is lower than the level of granularity. We will see in the next chapter how we can handle these missing values. Based on the insights we have just gained, we select the most fine grained dataset for the remainder of this book as we feel that we would lose too much information and also valuable training data if we were to use the coarse-grained variant.

2.4 Machine Learning Tasks

Given that we have defined and created our dataset now, we should also define some goals we want to achieve with the application of machine learning techniques to the above dataset. In general, we can set goals in sync with the different learning approaches we have briefly discussed in Sect. 1.3.2. Focusing on supervised learning we define two tasks: (1) a classification problem, namely predicting the label (i.e. activity) based on the sensors, and (2) a regression problem, namely predicting the heart rate based on the other sensory values and the activity. In the rest of the book we will see how accurate we can perform these two tasks with our dataset.

2.5 Exercises

2.5.1 Pen and Paper

1. When we measure data using sensory devices across multiple users we often see substantial differences between the sensory values we obtain. Identify at least three potential causes for these differences.
2. We have seen that we can make trade-offs in terms of the granularity at which we consider the measurements in our basic dataset. We have shown the difference between a granularity of $\Delta t = 0.25\,\text{s}$ and $\Delta t = 60\,\text{s}$. We arrived at a choice for $\Delta t = 0.25\,\text{s}$ for our case, but let us think a bit more general: think of four criteria that play a role in deciding on the granularity for the measurements of a dataset.
3. We have identified two tasks we are going to tackle for the crowdsignals data. Think of at least two other machine learning tasks that could be performed on the crowdsignals dataset and argue why they could be relevant to support a user (when doing so, keep in mind the different learning approaches discussed in Sect. 1.3.2).

2.5.2 Coding

1. Create your own dataset for the quantified self by using your smartphone. You can create the dataset using measurement apps on your smartphone (e.g. at the time of writing Funf, SensorLog, phybox, or SensorKinetics) or other devices. Include repeated periods with different activities (please incorporate some we have seen in the crowdsignal data and some that are different) and study the variation you see in the sensory values. Be sure to include periods without any specific activities to study the background noise of the sensors. Log the intervals at which you performed the different activities.
    a. Plot and describe the data you obtain using the libraries provided with the book.
    b. Try different values for Δt and describe the differences you see.
2. Compare the sensory values you have obtained with your measurements to those in the crowdsignals dataset over comparable activities. What would be the best way to compare the values given that the values might result from different sensors with different scales? And how different are the two datasets?
3. Find a dataset on the web that covers data from multiple users (for a list of data sources check the book's website). Note, that there are quite a few datasets that come along with an accompanying scientific article, see for example Anguita et al. [7], Banos et al. [10], or Zhang et al. [131]. Study and describe the variation you see in terms of sensory values over different users. Plot some differences that stand out and identify potential causes for these differences (e.g. by considering the ones you listed under the *pen and paper* exercises).

http://www.springer.com/978-3-319-66307-4

Machine Learning for the Quantified Self
On the Art of Learning from Sensory Data
Hoogendoorn, M.; Funk, B.
2018, XV, 231 p. 89 illus., 72 illus. in color., Hardcover
ISBN: 978-3-319-66307-4