Data preparation

1. Data preparation

00:00 - 00:06

Great job so far! I'm Hadrien, and I will now tell you about data preparation.

2. Data workflow

00:06 - 00:11

Data preparation happens after collecting and storing the data.

3. Why prepare data?

00:11 - 00:41

Data rarely comes in ready for analysis. Real-life data is messy and dirty. It needs to be cleaned. Skipping this step may lead to errors down the way, incorrect results, or throw off your algorithms. You would not use vegetables without cleaning, peeling and dicing them, as your soup would taste weird and no one would eat it. Well, if you don't clean, peel and dice your data, your results will look weird, and no one will use them!

4. Let's start cleaning

00:41 - 00:48

Let's take a simple, but dirty, dataset, and clean it together. Maybe you can already notice a few things.

5. Tidy data

00:48 - 01:21

One fundamental aspect of cleaning data is "tidiness". Tidy data is a way of presenting a matrix of data, with observations on rows and variables as columns. This is not the case here. Our observations (people) are in columns, and their features are on rows. Let's take care of that. It's easy to do that programmatically with Python or R. They also help with the other cases you're about to see.

6. Tidy data output

01:21 - 01:23

The data looks much clearer this way.

7. Remove duplicates

01:23 - 01:33

In general, you want to remove duplicates. Python and R make them easy to identify. Here we can see that Lis appears twice.

8. Remove duplicates | output

01:33 - 01:36

Let's remove the duplicate.

9. Unique ID

01:36 - 01:48

What if there's another person called Lis? Then, you want a way to uniquely identify each observation. It can be a combination of features (name plus last name plus year of birth, for example),

10. Unique ID | output

01:48 - 01:56

but the safest way is to assign a unique ID. Sara's ID is now 0, Lis' 1 and Hadrien's 2.

11. Homogeneity

01:56 - 02:16

Something fishy is going on in the size column. Lis simply can't be that tall (or Hadrien and Sara that small, depending where you're from). Lis is in the US, she inputted her size in feet. Sara and Hadrien are based in Europe, they use the metric system. All variables should use the same standard.

12. Homogeneity | output

02:16 - 02:26

Programmatically, you can filter values above 2.5 meters, and apply a division by 3.281 to get the metric value. Here we go.

13. Homogeneity, again

02:26 - 02:36

Similarly, countries should follow the same format. The United States and France are abbreviated, but Belgium is written in full. Let's fix that.

14. Homogeneity, again | output

02:36 - 02:38

Looking better already!

15. Data types

02:38 - 03:03

Another common issue relates to data types. The tools you use might be able to infer data types for each column, but you'd better make sure they are correct. Here, the Age column is encoded as text. If you try to get the mean, you'll get an error, because the average of two words doesn't make sense. You should change the type of this feature to numbers.

16. Data types | output

03:03 - 03:08

Ages are now numbers; you can see the quotes have disappeared.

17. Missing values

03:08 - 03:53

Last but not least, missing values. They are common and occur for various reasons: the agent doing the entry was distracted, the person surveyed did not understand the question, or it's on purpose, for example an event that has not happened yet. There are several ways to deal with missing values. You can substitute the exact value if you have access to the source. For example, you can take an aggregate value, like the mean, median or max depending on the situation. You can drop the observation altogether, but each observation you remove means less training data for your model. Or, you can keep it as is and ignore it, if your algorithm allows it.

18. Missing values | output

03:53 - 04:00

Here, we take the mean, 27.5, and round it up to get 28, which happens to be the correct value.

19. Let's practice!

04:00 - 04:03

Let's check your understanding!