Other approaches to data transformation

As you know, data comes to us in many different forms. For categorical or qualitative data types, data professionals often need to transform (or encode) this type of data into numeric digits to complete their analysis, design their data visualization, or build their machine learning algorithm. In this reading, you will learn about the two main types of categorical data encoding and when to use each type.

Label encoding

You've already learned about **label encoding,** which is one type of data transformation technique where each data value is assigned a distinct number instead of a qualitative value.

If you remember from the video, the example provided was label encoding mushroom types:

Mushroom Type	Code
Black truffle	0
Button	1
Cremini	2
Hedgehog	3
King Trumpet	4
Morel	5
Portobello	6
Shiitake	7
Toadstool	8

As you can tell, for this hypothetical data set about mushrooms, each mushroom type was assigned its own number, starting with zero.

Some potential problems with label encoding

Imagine you're analyzing a dataset with categories of music genres. You label encode "Blues," "Electronic Dance Music (EDM)," "Hip Hop," "Jazz," "K-Pop," "Metal," " and "Rock," with the following numeric values, "1, 2, 3, 4, 5, 6, and 7."

With this label encoding, the resulting machine learning model **could** derive not only a ranking, but also a closer connection between Blues (1) and EDM (2) because of how close they are numerically than, say, Blues(1) and Jazz(4). In addition to these presumed relationships (which you may or may not want in your analysis) you should also notice that each code is equidistant from the other in the numeric sequence, as in 1 to 2 is the same distance as 5 to 6, etc. The question is, does that equidistant relationship accurately represent the relationships between the music genres in your dataset? To ask another question, after encoding, will the visualization or model you build treat the encoded labels as a ranking?

The same could be said for the mushroom example above. After label encoding mushroom types, are you satisfied with the fact that the mushrooms are now in a presumed ranked order with button mushrooms ranked first and toadstool ranked eighth?

In summary, label encoding **may** introduce unintended relationships between the categorical data in your dataset. When you are making decisions about label encoding, consider the algorithm you'll apply to the data and how it may or may not impact label encoded categorical data.

Fortunately, there is another method for categorical encoding that may help with these potential problems.

One-hot encoding

As you learned in a previous video, you can create dummy variables in Python. If you remember, a dummy variable is a variable with values of 0 or 1, which indicate the presence or absence of something. The idea is to create a new column for each category type, then for each value indicate a 0 or a 1 — 0 meaning, no, and 1 meaning, yes.

This creation of dummies is called **one-hot encoding.** As a reminder, a table with one-hot encoding ends up like this:

N/A	Mild	Scattered	Heavy	Severe
0	1	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	1
6	0	0	0	1
7	0	0	0	1
8	0	0	1	0
9	0	1	0	0
10	1	0	0	0
11	1	0	0	0
12	0	1	0	0

You'll find the values from the lightning strike dataset covered in the video are labeled as "mild" and have a "1." "Mild" refers to the lowest quartile of lightning counts in the dataset. For any other value in the "mild" column that is NOT mild, there is a zero in that cell. With this method, we solve the problem of the unintended and problematic relationships that label encoding presented.

But one-hot encoding does present its own set of problems, particularly when it comes to logistic and linear regression. You will learn more about that in a future course.

Label encoding or one-hot encoding: How to decide?

There is no simple answer to whether you should use label encoding or one-hot encoding. The decision needs to be made on a case-by-case, or dataset-by-dataset basis. But there are some guidelines to help you.

Use label encoding when:

- There are a large number of different categorical variables — because label encoding uses far less data than one-hot encoding
- The categorical values have a particular order to them (for example, age groups can be grouped as youngest to oldest or oldest to youngest)
- You plan to use a decision tree or random forest machine learning model

Use one-hot encoding when:

- There is a relatively small amount of categorical variables — because one-hot encoding uses much more data than label encoding.
- The categorical variables have no particular order
- You use a machine learning model in combination with dimensionality reduction (like Principal Component Analysis (PCA))

Key takeaways

Label encoding and one-hot encoding are techniques for transforming categorical data to numeric data. Label encoding is best for large numbers of different categorical variables and for categories that have an inherent order to them. One-hot encoding is best for smaller amounts of categorical variables and for categories that have no order.

Mark as completed