Why One-Hot Encode Data in Machine Learning?

by **Jason Brownlee** on July 28, 2017 in **Data Preparation**

Tweet**Share**

Last Updated on April 27, 2020

Getting started in applied machine learning can be difficult, especially when working with real-world data.

Often, machine learning tutorials will recommend or require that you prepare your data in specific ways before fitting a machine learning model.

One good example is to use a one-hot encoding on categorical data.

- Why is a one-hot encoding required?
- Why can’t you fit a model on your data directly?

In this post, you will discover the answer to these important questions and better understand data preparation in general in applied machine learning.

Let’s get started.

Why One-Hot Encode Data in Machine Learning?

Photo by Karan Jain, some rights reserved.

What is Categorical Data?

Categorical data are variables that contain label values rather than numeric values.

The number of possible values is often limited to a fixed set.

Categorical variables are often called nominal.
Some examples include:

- A “*pet*” variable with the values: “*dog*” and “*cat*“.
- A “*color*” variable with the values: “*red*“, “*green*” and “*blue*“.
- A “*place*” variable with the values: “first”, “*second*” *and* “*third*“.

Each value represents a different category.

Some categories may have a natural relationship to each other, such as a natural ordering.

The “*place*” variable above does have a natural ordering of values. This type of categorical variable is called an ordinal variable.

What is the Problem with Categorical Data?

Some algorithms can work with categorical data directly.

For example, a decision tree can be learned directly from categorical data with no data transform required (this depends on the specific implementation).

Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.

In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.

This means that categorical data must be converted to a numerical form. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form in order to present them or use them in some application.

How to Convert Categorical Data to Numerical Data?

This involves two steps:

1. Integer Encoding
2. One-Hot Encoding

1. Integer Encoding

As a first step, each unique category value is assigned an integer value.

For example, "*red*" is 1, "*green*" is 2, and "*blue*" is 3.
This is called a label encoding or an integer encoding and is easily reversible.

For some variables, this may be enough.

The integer values have a natural ordered relationship between each other and machine learning algorithms may be able to understand and harness this relationship.

For example, ordinal variables like the "place" example above would be a good example where a label encoding would be sufficient.

2. One-Hot Encoding

For categorical variables where no such ordinal relationship exists, the integer encoding is not enough.

In fact, using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results (predictions halfway between categories).

In this case, a one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

In the “*color*” variable example, there are 3 categories and therefore 3 binary variables are needed. A “1” value is placed in the binary variable for the color and “0” values for the other colors.
For example:

```
red,      green,    blue
1,                  0,                  0
0,                  1,                  0
0,                  0,                  1
```

The binary variables are often called “dummy variables” in other fields, such as statistics.

One Hot Encoding Tutorials

Looking for some tutorials on how to one hot encode your data in Python, see:

- Data Preparation for Gradient Boosting with XGBoost in Python
- How to One Hot Encode Sequence Data in Python

Further Reading

- Categorical variable on Wikipedia
- Nominal category on Wikipedia
- Dummy variable on Wikipedia

Summary

In this post, you discovered why categorical data often must be encoded when working with machine learning algorithms.

Specifically:

- That categorical data is defined as variables with a finite set of label values.
- That most machine learning algorithms require numerical input and output variables.
- That an integer and one hot encoding is used to convert categorical data to integer data.

Do you have any questions?

Post your questions to comments below and I will do my best to answer.