# One Hot Encoder

## What is one hot encoding?

Categorical data refers to variables that are made up of label values, for example, a “color” variable could have the values “red,” “blue,” and “green.” Think of values like different categories that sometimes have a natural ordering to them.

Some machine learning algorithms can work directly with categorical data depending on implementation, such as a decision tree, but most require any inputs or outputs variables to be a number, or numeric in value. This means that any categorical data must be mapped to integers.

One hot encoding is one method of converting data to prepare it for an algorithm and get a better prediction. With one-hot, we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector. All the values are zero, and the index is marked with a 1.

Take a look at this chart for a better understanding:

| Type |
|---|
| AA |
| AB |
| CD |
| AA |

**Onehot encoding** →

| Type | AA_Onehot | AB_Onehot | CD_Onehot |
|---|---|---|---|
| AA | 1 | 0 | 0 |
| AB | 0 | 1 | 0 |
| CD | 0 | 0 | 1 |
| AA | 0 | 0 | 0 |

## Why use one hot encoding?

One hot encoding is useful for data that has no relationship to each other. Machine learning algorithms treat the order of numbers as an attribute of significance. In other words, they will read a higher number as better or more important than a lower number.

While this is helpful for some ordinal situations, some input data does not have any ranking for category values, and this can lead to issues with predictions and poor performance. That’s when one hot encoding saves the day.

One hot encoding makes our training data more useful and expressive, and it can be rescaled easily. By using numeric values, we more easily determine a probability for our values.

Inparticular, one hot encoding is used for our output values, since it provides more nuanced predictions than single labels.

## One hot encoding with Pandas:

We don’t have to one hot encode manually. Many data science tools offer easy ways to encode your data. The Python library Pandas provides a function called get_dummies to enable one-hot encoding.

```python
import pandas as pd

df = pd.DataFrame({"col1": ["Sun", "Sun", "Moon", "Earth", "Moon", "Venus"]})
print("The original data")
print(df)
print("*" * 30)
df_new = pd.get_dummies(df, columns=["col1"], prefix="Planet")
print("The transform data using get_dummies")
print(df_new)
```

Run

Output 1.18s

```
******************************
The transform data using get_dummies
   Planet_Earth  Planet_Moon  Planet_Sun  Planet_Venus
0             0            0           1             0
1             0            0           1             0
2             0            1           0             0
3             1            0           0             0
4             0            1           0             0
5             0            0           0             1
```