Handling Categorical Data

Data that can be categorized but lacks an inherent hierarchy or order is known as categorical data.

In other words, there is no mathematical connection between the categories. A person's gender (male/female), eye color (blue, green, brown, etc.), type of vehicle they drive (sedan, SUV, truck, etc.), or the kind of fruit they consume (apple, banana, orange, etc.) are examples of categorical data.

Categorical Dataset

Information is represented using two different forms of data: **categorical data and numeric data.**

- Data that may be categorized or grouped together is referred to as categorical data.

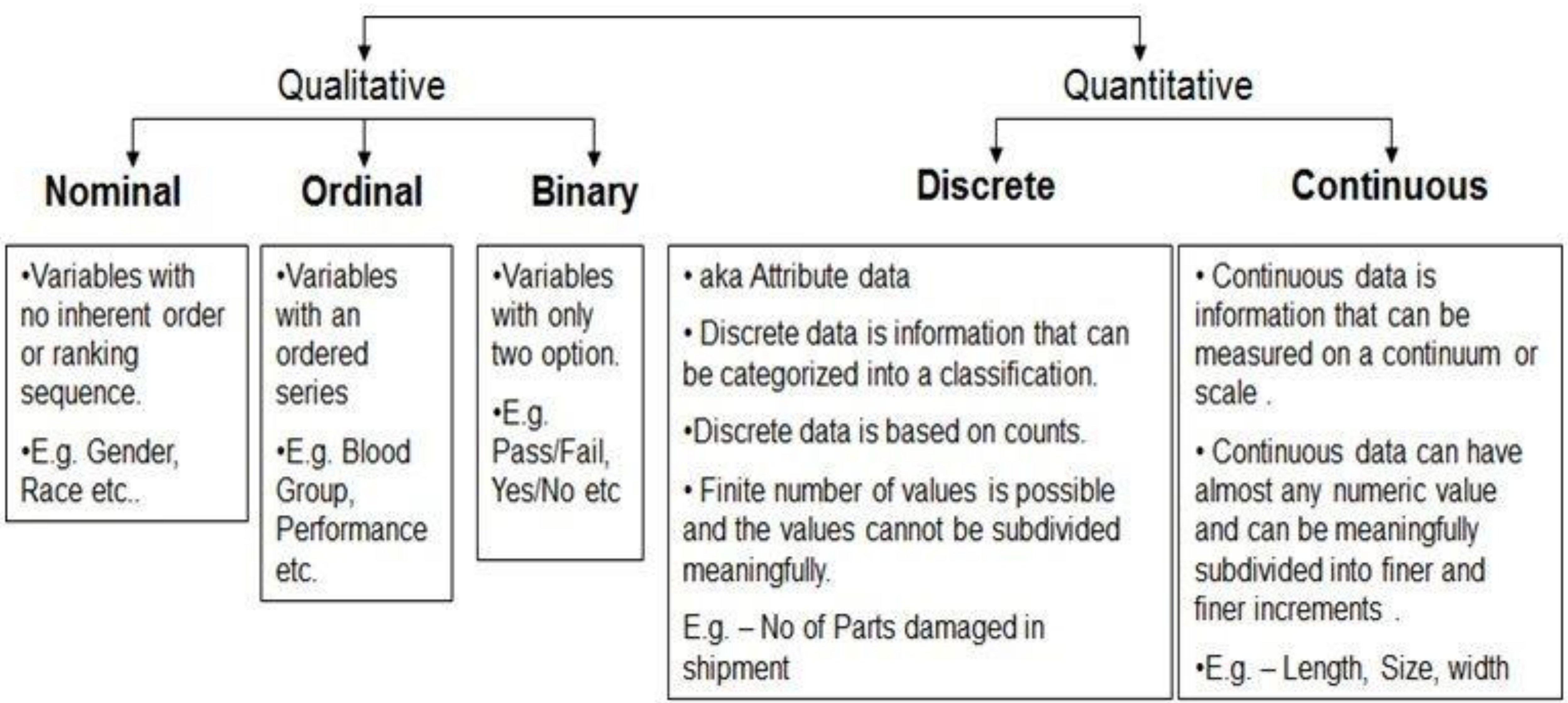
  Example: Men and women fall into the gender category, the colors red, green, and blue fall into the colors category, and the countries category might include the US, Canada, Mexico, etc.

- **Numeric data** refers to data that can be expressed as a number.

  Examples of numeric data include height, weight, and temperature.

- Majority of machine learning algorithms are created to operate with numerical data, categorical data is handled differently from numerical data in this field.

Before categorical data can be utilized as input to a machine learning model, it must first be transformed into numerical data. This process of converting categorical data into numeric representation is known as **encoding**.



Qualitative

Nominal

- Variables with no inherent order or ranking sequence.
- E.g. Gender, Race etc..

Ordinal

- Variables with an ordered series
- E.g. Blood Group, Performance etc.

Binary

- Variables with only two option.
- E.g. Pass/Fail, Yes/No etc

Quantitative

Discrete

- aka Attribute data
- Discrete data is information that can be categorized into a classification.
- Discrete data is based on counts.
- Finite number of values is possible and the values cannot be subdivided meaningfully.

E.g. – No of Parts damaged in shipment

Continuous

- Continuous data is information that can be measured on a continuum or scale .
- Continuous data can have almost any numeric value and can be meaningfully subdivided into finer and finer increments .
- E.g. – Length, Size, width

There are two types of categorical data: nominal and ordinal.

Nominal data

Nominal data is categorical data that may be divided into groups, but these groups lack any intrinsic hierarchy or order.

Examples of nominal data include brand names (Coca-Cola, Pepsi, Sprite), varieties of pizza toppings(pepperoni, mushrooms, onions), and hair color (blonde, brown, black, etc.).

Ordinal data

Ordinal data, on the other hand, describes information that can be categorized and has a distinct order or ranking.

Examples: Levels of education (high school, bachelor's, master's), levels of work satisfaction (extremely satisfied, satisfied, neutral, unsatisfied, very unsatisfied), and star ratings (1-star, 2-star, 3-star, 4-star, 5-star) are a few examples of ordinal data.

By giving each category a numerical value that reflects its order or ranking, ordinal data can be transformed into numerical data and used in machine learning. For algorithms that are sensitive to the size of the input data, this may be helpful.

Encoding Categorical Features

Categorical data cannot typically be directly handled by machine learning algorithms, as most algorithms are primarily designed to operate with numerical data only.

Therefore, before categorical features can be used as inputs to machine learning algorithms, they must be encoded as numerical values.

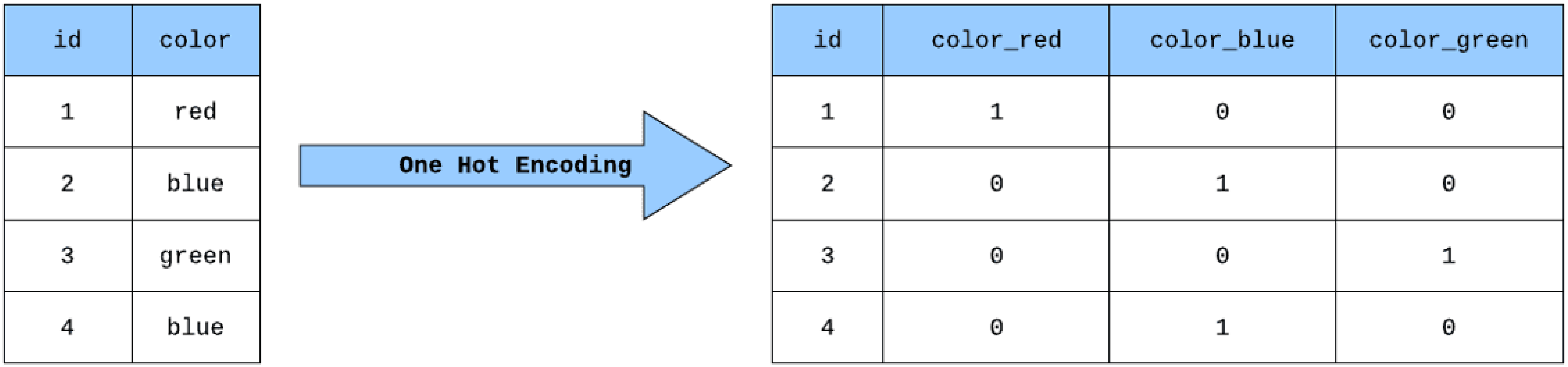
There are several techniques for encoding categorical features, including

- **One-hot encoding**
- **Label Encoding**
- **Ordinal encoding**
- **Target encoding.**
- **Frequency Encoding**

The choice of encoding technique depends on the specific characteristics of the data and the requirements of the machine learning algorithm being used.

One-hot encoding

One hot encoding is a process of representing categorical data as a set of binary values, where each category is mapped to a unique binary value. In this representation, only one bit is set to 1, and the rest are set to 0, hence the name "one



id	color
1	red
2	blue
3	green
4	blue

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

pandas categorical to numeric

One way to achieve this in pandas is by using the `pd.get_dummies()` method.

- It is a function in the Pandas library that can be used to perform one-hot encoding on categorical variables in a DataFrame.

- It takes a DataFrame and returns a new DataFrame with binary columns for each category.

Even though `pandas.get_dummies` is straightforward to use, a more common approach is to use `OneHotEncoder` from the sklearn library, especially when you are doing machine learning tasks.
The primary difference is `pandas.get_dummies` cannot learn encodings; it can only perform one-hot-encoding on the dataset you pass as an input.
On the other hand, `sklearn.OneHotEncoder` is a class that can be saved and used to transform other incoming datasets in the future.

Label encoding

Label encoding is a technique for encoding categorical variables as numeric values, with each category assigned a unique integer.

For example, suppose we have a categorical variable "color" with three categories: "red," "green," and "blue." We can encode these categories using label encoding as follows (red: 0, green: 1, blue: 2).

- Label encoding can be useful for some machine learning algorithms that require numeric inputs, as it allows categorical data to be represented in a way that the algorithms can understand.

- Label encoding introduces an arbitrary ordering of the categories, which may not necessarily reflect any meaningful relationship between them.

- In some cases, this can lead to problems in the analysis, especially if the ordering is interpreted as having some kind of ordinal relationship.

Label encoding

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

Label Encoded Data

Team	Points
0	25
0	12
1	15
1	14
1	19
1	23
2	25
2	29

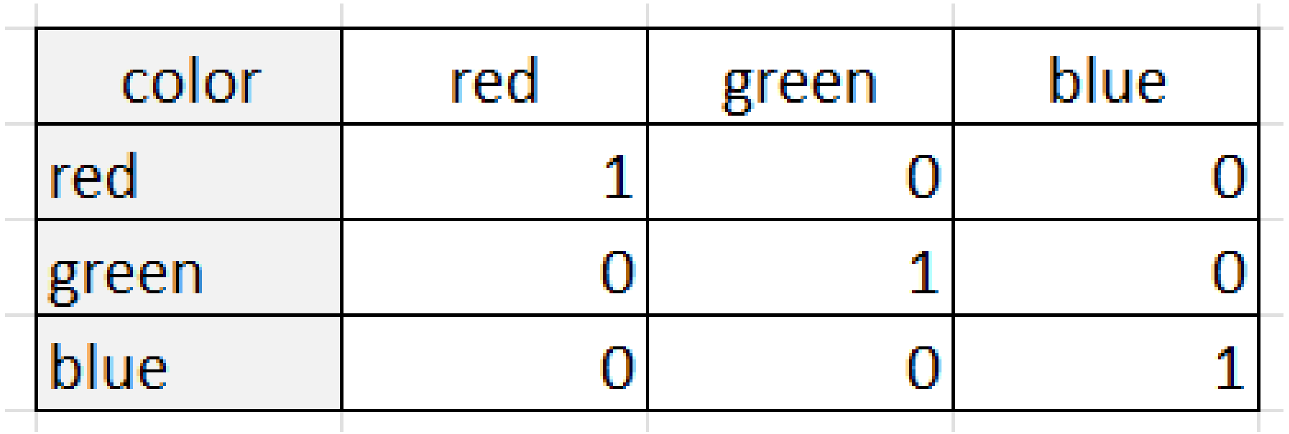
Comparison of One-hot and Label Encoding

- One-hot encoding and label encoding are both techniques for encoding categorical variables as numeric values, but they have different properties and are appropriate for different use cases.

- One-hot encoding represents each category as a binary column, with a 1 indicating the presence of the category and a 0 indicating its absence.

Example, suppose we have a categorical variable "color" with three categories: "red," "green," and "blue."

One-hot encoding would represent this variable as three binary columns:



color	red	green	blue
red	1	0	0
green	0	1	0
blue	0	0	1

Comparison of One-hot and Label Encoding

- One-hot encoding is appropriate when the categories do not have an intrinsic ordering or relationship with each other. This is because one-hot encoding treats each category as a separate entity with no relation to the other categories.

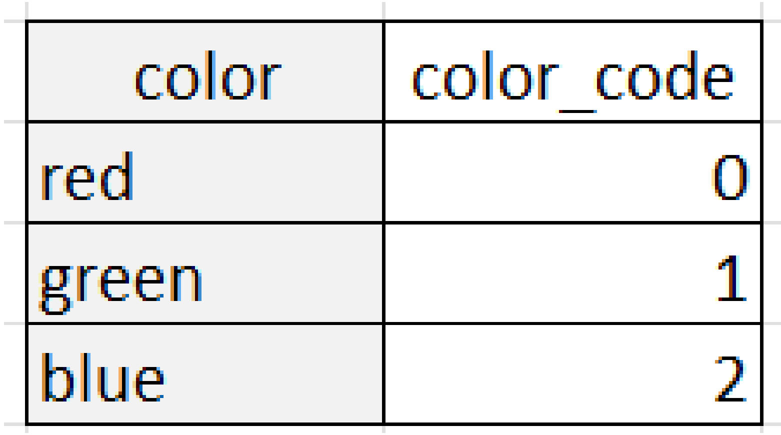
- One-hot encoding is also useful when the number of categories is relatively small, as the number of columns can become unwieldy for very large numbers of categories.

- Label encoding, on the other hand, represents each category as a unique integer.

For example, the "color" variable with three categories could be label-encoded as:

- Label encoding is appropriate when the categories have a natural ordering or relationship with each other, such as in the case of ordinal variables like "small," "medium," and "large." In these cases, the integer values assigned to the categories should reflect the ordering of the categories.

- Label encoding can also be useful when the number of categories is very large, as it reduces the dimensionality of the data.



color	color_code
red	0
green	1
blue	2

Comparison of One-hot and Label Encoding

In general, one-hot encoding is more commonly used in machine learning applications, as it is more flexible and avoids the problems of ambiguity and arbitrary ordering that can arise with label encoding.

However, label encoding can be useful in certain contexts where the categories have a natural ordering or when dealing with very large numbers of categories.

Dealing with High Cardinality Categorical Data

- High cardinality refers to a large number of unique categories in a categorical feature. Dealing with high cardinality is a common challenge in encoding categorical data for machine learning models.
- High cardinality can lead to sparse data representation and can have a negative impact on the performance of some machine learning models.

Techniques that can be used to deal with high cardinality in categorical features:

Rare Categories

Involves combining infrequent categories into a single category. This reduces the number of unique categories and also reduces the sparsity in the data representation.

Target Encoding

Target encoding replaces the categorical values with the mean target value of that category. It provides a more continuous representation of the categorical data and can help capture the relationship between the categorical feature and the target variable.

Target Encoding

workclass	target
State-gov	0
Self-emp-not-inc	1
Private	0
Private	0
Private	1

workclass	target mean
State-gov	0
Self-emp-not-inc	1
Private	1/3

workclass
0
1
1/3
1/3
1/3

Ordinal Encoding

- Ordinal encoding’s encoded variables retain the ordinal (ordered) nature of the variable.

- Looks similar to label encoding, the only difference being that label coding doesn't consider whether a variable is ordinal or not. It will then assign a sequence of integers.

Example: Ordinal encoding will assign values as Very Good(1) < Good(2) < Bad(3) < Worse(4)

First, we need to assign the original order of the variable through a dictionary.

Frequency Encoding

The category is assigned as per the frequency of values in its total lot.