

Label Encoder and **One-Hot Encoder** are both tools used in machine learning and data preprocessing to handle categorical data, which are variables that take on a limited, fixed number of possible values.

1. Label Encoder:

- **Purpose:** Label Encoder converts categorical labels into numerical labels.
- **How it works:** It assigns a unique integer (label) to each category in the categorical variable. For example, if you have categories like "red", "blue", and "green", Label Encoder might assign them 0, 1, and 2 respectively.
- **Usage:** Label Encoder is commonly used for ordinal data where there is a clear ranking or order among the categories. However, it should be used with caution because the numerical labels can be misinterpreted as having some kind of ordinal significance in models.

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
encoded_labels = encoder.fit_transform(['red', 'blue', 'green', 'red'])
print(encoded_labels)

#Output: [2, 0, 1, 2]
```

2. One-Hot Encoder:

- **Purpose:** One-Hot Encoder converts categorical integer features into binary vectors.
- **How it works:** It creates a binary column for each category and returns a matrix where each row corresponds to the original categorical variable, and each column corresponds to one of the possible categories.
- **Usage:** One-Hot Encoder is used when categorical variables are nominal (unordered) and when the model should not interpret them as having any ordinal relationship. It is essential for many machine learning algorithms that work with categorical data.

```
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder()
encoded_features = encoder.fit_transform([['red'], ['blue'], ['green'],
['red']]).toarray()
print(encoded_features)

#output:  [[0. 0. 1.]
#          [1. 0. 0.]
#          [0. 1. 0.]
#          [0. 0. 1.]
```

In summary, while Label Encoder assigns numerical labels to categories, One-Hot Encoder creates binary vectors representing each category separately, which is crucial for many machine learning algorithms to correctly interpret categorical data.