**Label Encoding** and **One-Hot Encoding** are two common techniques used to convert categorical variables into numerical representations for machine learning models.

It's important to choose the right encoding technique based on whether the data is **nominal** or **ordinal**

|  |  |
| --- | --- |
| **Nominal Data** | **Ordinal Data** |
| Categories with no inherent order | Categories with a natural order or ranking |
| One-Hot Encoding (preferred) | Label Encoding |
| Colors (Red, Blue, Green), Cities (Paris, NYC) | Education levels (Low, Medium, High) |
| pd.get\_dummies() | LabelEncoder() from sklearn.preprocessing |

**One-Hot Encoded Nominal Data:**

One-hot encoding transforms categorical data into a binary vector format that’s easier for ML models to comprehend. Each category is represented by a binary vector with a 1 at the point corresponding to the category and 0s everywhere else.

**Multicollinearity** is a situation where two or more independent variables are closely linked or correlated, making it hard to differentiate their separate effects.

To prevent multicollinearity, we remove one of the columns after/during the one-hot encoding.

The python pandas library includes a built-in function named get\_dummies() for implementing one-hot encoding.

By using the drop\_first parameter, we can eliminate the first dummy column.

------------------------------------------------------------

‘Color’ is original column and ‘Color\_Blue’, ‘Color\_Green’, ‘Color\_Red’ columns are dummies

Color\_Blue Color\_Green Color\_Red

0 0 1

0 1 0

1 0 0

0 1 0

0 0 1

Here one column is redundant which we can drop to remove **multicollinearity.**

**Label Encoded Ordinal Data:** [1 2 0 2 1]

Mapping of Ordinal Categories to Integers:

High -> 0

Low -> 1

Medium -> 2

======================================