## Encoding (In Machine Learning converting Categorical data into numeric):

In machine learning, encoding refers to the process of converting data into a suitable format for model training and inference. The types of encoding in machine learning can be broadly categorized into two main groups: categorical data encoding and numerical data encoding. Here are the common types of encoding used in machine learning:

### *1. Categorical Data Encoding*

Categorical data encoding is used to convert categorical (qualitative) data into numerical values so that machine learning algorithms can process them.

- **Label Encoding:**
    - Converts each unique category value to an integer.
    - Example: ['cat', 'dog', 'mouse'] becomes [0, 1, 2].
    - Suitable for ordinal data where there is a meaningful order.
- **One-Hot Encoding:**
    - Converts each category value into a new binary column.
    - Example: ['cat', 'dog', 'mouse'] becomes [[1, 0, 0], [0, 1, 0], [0, 0, 1]].
    - Suitable for nominal data where there is no ordinal relationship.
- **Binary Encoding:**
    - Converts categories into binary numbers and splits these binary digits into separate columns.
    - Example: ['A', 'B', 'C'] might become ['001', '010', '011'], which is then split into separate columns.
    - Useful when dealing with high cardinality categorical data.
- **Target Encoding (Mean Encoding):**
    - Replaces each category with the mean of the target variable for that category.
    - Example: For a target variable, categories like ['A', 'B', 'C'] might be encoded based on their mean target values [0.5, 0.3, 0.8].
- **Frequency Encoding:**
    - Replaces each category with its frequency or count.
    - Example: ['cat', 'cat', 'dog'] might become [2, 2, 1].
- **Ordinal Encoding:**
    - Assigns an integer value to each category, preserving the order.
    - Example: ['low', 'medium', 'high'] becomes [1, 2, 3].

### *2. Numerical Data Encoding*

Numerical data encoding involves transforming numerical features to make them more suitable for certain machine learning algorithms.

- **Normalization (Min-Max Scaling):**
    - Scales the data to a fixed range, usually [0, 1].
    - Formula: ( $x' = \frac{x - \text{min}(x)}{\text{max}(x) - \text{min}(x)}$ ).
- **Standardization (Z-Score Normalization):**
    - Scales the data to have a mean of 0 and a standard deviation of 1.
    - Formula: ( $x' = \frac{x - \mu}{\sigma}$ ), where ( $\mu$ ) is the mean and ( $\sigma$ ) is the standard deviation.
- **Binning (Discretization):**
    - Divides continuous values into discrete bins or intervals.
    - Example: Age can be binned into categories like ['0-10', '11-20', '21-30'].
- **Log Transformation:**
    - Applies the logarithm to data, often used to reduce skewness.

    - Formula: $x' = \log(x + 1)$.
- **Polynomial Features:**
    - Generates new features by taking powers of existing features.
    - Example: For a feature $x$, polynomial features could include $x^2$, $x^3$, etc.
- **Power Transformation:**
    - Stabilizes variance and makes the data more Gaussian-like.
    - Includes techniques like Box-Cox and Yeo-Johnson transformations.

### *3. Text Data Encoding*

Text data needs to be converted into numerical form for use in machine learning models.

- **Bag of Words (BoW):**
    - Converts text into a fixed-size vector based on word counts.
    - Example: The sentence "I love machine learning" might become a vector of word frequencies.
- **TF-IDF (Term Frequency-Inverse Document Frequency):**
    - Similar to BoW but adjusts word frequencies based on their importance across documents.
    - Formula: $\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log \left( \frac{N}{\text{DF}(t)} \right)$.
- **Word Embeddings (Word2Vec, GloVe):**
    - Converts words into dense vectors capturing semantic relationships.
    - Example: Words with similar meanings have vectors that are close in the vector space.
- **Sentence Embeddings:**
    - Converts entire sentences into dense vectors.
    - Example: Using models like BERT or Sentence-BERT to generate embeddings that capture sentence-level semantics.

These encoding techniques are crucial for preparing data for machine learning models, ensuring that the data is in a format that algorithms can effectively process and learn from. The choice of encoding method depends on the type of data and the specific requirements of the machine learning task.