Encoding Categorical Data

Machine learning models can't directly handle **categorical variables** (like "Red", "Blue", "Green"). So we **convert them into numbers** using **encoding techniques**.

Introduction:

- Categorical data = Non-numeric data
- **Examples:** Gender = ["Male", "Female"], City = ["Lahore", "Karachi", "Islamabad"]
- Encoding helps **transform strings → numbers** so ML models can understand them

---

Revision:

Quick recap of:

- **Types of data**: Numeric (int, float), Categorical (object/string)
- **Why encoding is needed**: Models can't process text
- Categorical variables are of two types:
    1. **Nominal** (no order): e.g., ["Apple", "Banana", "Mango"]
    2. **Ordinal** (has order): e.g., ["Low", "Medium", "High"]

What is Ordinal Data?

- Ordinal data has a **meaningful order** but not exact differences
- Examples:
    1. T-shirt Sizes: **Small < Medium < Large**
    2. Ratings: **Bad < Average < Good < Excellent**
- You **should not use Label Encoding on nominal data** (it imposes false order)

Label Encoding

- Assigns **integer labels** to each unique category
- Good for **ordinal data** (but misused sometimes on nominal)

Warning: For nominal data, the model might think **red > green > blue**, which is wrong!

How Ordinal Encoding Works?

- Use **OrdinalEncoder** when the categories **have a natural order**
- Must **manually define order** if needed

Summary

Encoding Type	Use For	Preserves Order?	Example Tool
Label Encoding	Ordinal (if no specific order is defined)	❌	LabelEncoder()
Ordinal Encoding	Ordered categories like ["Low", "Medium", "High"]	✅	OrdinalEncoder()
