Data Handling Techniques

Data types in ML

in Machine Learning (ML), the primary data types are numerical data (including integers and floats), categorical data, text data, and time series data; each with its own characteristics and usage depending on the problem you're trying to solve with your ML model

**Techniques to handle the data set   
Null values:**

1. Remove the column if data is 70% null
2. If data type is numeric and there are few null values, replace with mode.
3. If data type is continuous and there are few null values, replace with median

**Duplicates:**

The redundancy in data or the repetition of exact same data is called duplicated. We always have to remove it.

**IQR Z\_score:**

1. **IQR (Interquartile Range):**

* Measures the middle 50% spread of the data.
* **Formula:** IQR = Q3 - Q1 (where Q1 is the 25th percentile and Q3 is the 75th percentile).
* Outliers are typically outside:
  + Lower bound: Q1 - 1.5 × IQR
  + Upper bound: Q3 + 1.5 × IQR

1. **Z-score:**

* Measures how many standard deviations a data point is from the mean.
* **Formula:** Z = (X - μ) / σ (where μ = mean, σ = standard deviation).
* Outliers usually have |Z| > 3 (more than 3 standard deviations away).

**Anomaly Detection:**

Anomaly detection is a process that identifies data points that are different from what's normal or expected. It's also known as outlier detection.

**Why it's important**

* Helps businesses detect harmful outliers and protect data
* Can help identify fraud, security threats, and equipment malfunctions
* Can also help identify opportunities for improvement

**Box Plot**

A **Box Plot** is a statistical visualization tool used to display the **distribution, variability, and outliers** in a dataset. It summarizes data using five key values:

1. **Minimum** (Lowest value, excluding outliers)
2. **First Quartile (Q1)** – 25th percentile
3. **Median (Q2)** – 50th percentile (middle value)
4. **Third Quartile (Q3)** – 75th percentile
5. **Maximum** (Highest value, excluding outliers)

📌 **Outliers** are points beyond 1.5 × IQR and are plotted separately

**Box Plot Formula:**

**Interquartile Range (IQR)**

IQR=Q3−Q1IQR = Q3 - Q1IQR=Q3−Q1

Outliers are detected using:

* **Lower Bound:** Q1−1.5×IQRQ1 - 1.5 \times IQRQ1−1.5×IQR
* **Upper Bound:** Q3+1.5×IQRQ3 + 1.5 \times IQRQ3+1.5×IQR

**Advantages of Box Plot ✅**

✔ **Identifies Outliers** easily  
✔ **Displays Data Spread** (minimum to maximum)  
✔ **Shows Skewness** (left or right skew)  
✔ **Good for Comparing Multiple Datasets**  
✔ **Handles Large Datasets Efficiently**

**Disadvantages of Box Plot ❌**

✖ **Does Not Show Exact Distribution** (like a histogram)  
✖ **Limited Detail** on individual data points  
✖ **Cannot Display Multi-Modal Distributions**  
✖ **Sensitive to Skewed Data**

**Importance of Box Plot 🔥**

🔹 Used in **Anomaly Detection** (detects outliers)  
🔹 Helps in **Data Preprocessing** before ML models  
🔹 Provides a **quick summary** of data distribution  
🔹 Used in **Financial Analysis, Healthcare, and Research**

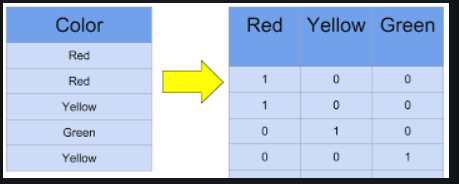
Box plots are a **powerful tool** in **exploratory data analysis (EDA)** and **machine learning** for understanding data behavior. 🚀

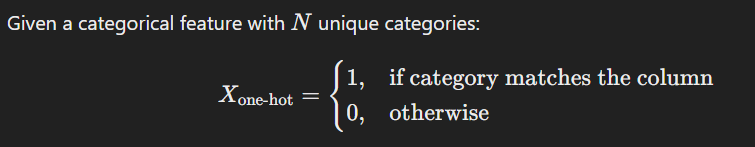
**One-Hot Encoding (OHE) 🔢**

**Definition:**

**One-Hot Encoding** is a technique used to convert **categorical data** into a **binary matrix** representation. It assigns a unique **binary vector** to each category, making it suitable for machine learning models that require numerical input.

**Formula for One-Hot Encoding:**





**Advantages of One-Hot Encoding ✅**

✔ **Makes Categorical Data Usable** for ML models  
✔ **Avoids Ordinal Relationships** (unlike Label Encoding)  
✔ **Works Well with Most ML Algorithms**

**Disadvantages of One-Hot Encoding ❌**

✖ **Increases Feature Space (High-Dimensionality)**  
✖ **Inefficient for Large Categorical Features** (e.g., 1000 categories → 1000 columns)  
✖ **Sparsity Issue** (many zero values, increasing computation cost)

**Importance of One-Hot Encoding 🔥**

🔹 **Essential for ML Models** that don’t handle categorical data  
🔹 Used in **Deep Learning, NLP, and Recommender Systems**  
🔹 **Prevents Incorrect Assumptions** about data relationships

One-Hot Encoding is a **crucial preprocessing step** for categorical data in ML models! 🚀

**Feature Engineering 🔍🚀**

**Definition:**

Feature Engineering is the process of **creating, transforming, or selecting features** to improve a machine learning model’s performance. It involves extracting the most useful information from raw data to enhance predictions.

**Types of Feature Engineering & Techniques:**

**1. Feature Creation**

* **Polynomial Features**: Create higher-order terms (e.g., X2,X3X^2, X^3)
* **Interaction Features**: Combine multiple features (e.g., X1×X2X\_1 \times X\_2)

**2. Feature Transformation**

* **Scaling**: Normalize or standardize numerical data
* **Encoding**: Convert categorical data (e.g., One-Hot Encoding, Label Encoding)
* **Log Transform**: Used for skewed data to make it normal

**3. Feature Selection**

* **Filter Methods**: Select features based on statistical scores (e.g., correlation)
* **Wrapper Methods**: Use models (e.g., RFE – Recursive Feature Elimination)
* **Embedded Methods**: Select features during training (e.g., LASSO Regression)

**4. Feature Extraction**

* **PCA (Principal Component Analysis)**: Reduces dimensionality while retaining variance
* **Autoencoders**: Extract features using deep learning
* **TF-IDF (Text Data)**: Converts text into numerical form

**Advantages of Feature Engineering ✅**

✔ **Improves Model Accuracy & Performance**  
✔ **Reduces Overfitting & Dimensionality**  
✔ **Enhances Interpretability**  
✔ **Helps ML Models Learn Better from Data**

**Disadvantages of Feature Engineering ❌**

✖ **Time-Consuming & Requires Expertise**  
✖ **Over-Engineering Can Lead to Overfitting**  
✖ **Not Always Generalizable to New Data**

**Importance of Feature Engineering 🔥**

🔹 **Transforms Raw Data into Meaningful Insights**  
🔹 **Reduces Computational Cost** of training models  
🔹 **Critical for Improving Model Generalization**

Feature Engineering is **one of the most powerful techniques** in **Machine Learning** for building better models! 🚀