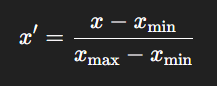
ML Questions

Q1. Difference between Normalization and Standardization with example and use case.

Ans.

| **Aspect** | **Normalization** | **Standardization** |
| --- | --- | --- |
| **Definition** | Rescales data to a fixed range, usually **0 to 1** (or -1 to 1). | Rescales data so **mean = 0** and **standard deviation = 1**. |
| **Result Range** | Fixed range (e.g., 0–1) | No fixed range (values can be negative or positive) |
| **Depends on** | Minimum and maximum values of data | Mean and standard deviation of data |
| **Effect of Outliers** | Highly affected (one extreme value can distort scaling) | Less affected (though still sensitive to extreme outliers) |
| **Best Used When** | You need all features in a fixed scale (especially for distance-based algorithms) | You need data centered and scaled for algorithms assuming Gaussian-like distribution |
| **Example** | Pixel values (0–255) in an image → Normalize to 0–1 for a neural network. | Heights of people for linear regression → Standardize to compare fairly with other features. |
| **Typical Algorithms** | K-Nearest Neighbors (KNN), Neural Networks, Image Processing | Linear Regression, Logistic Regression, SVM, PCA |
| **Why Use** | Prevents features with larger numbers from dominating the model’s calculations | Makes data comparable across features and improves convergence in gradient-based methods |

1. Normalization (Min–Max Scaling):-



Where –

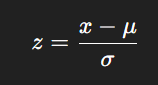
X = original value

x)min = smallest value in the dataset

x)max = largest value in the dataset

Result: Always between 0 and 1 (unless you choose another range).

1. Standardization (Z-score Scaling):-



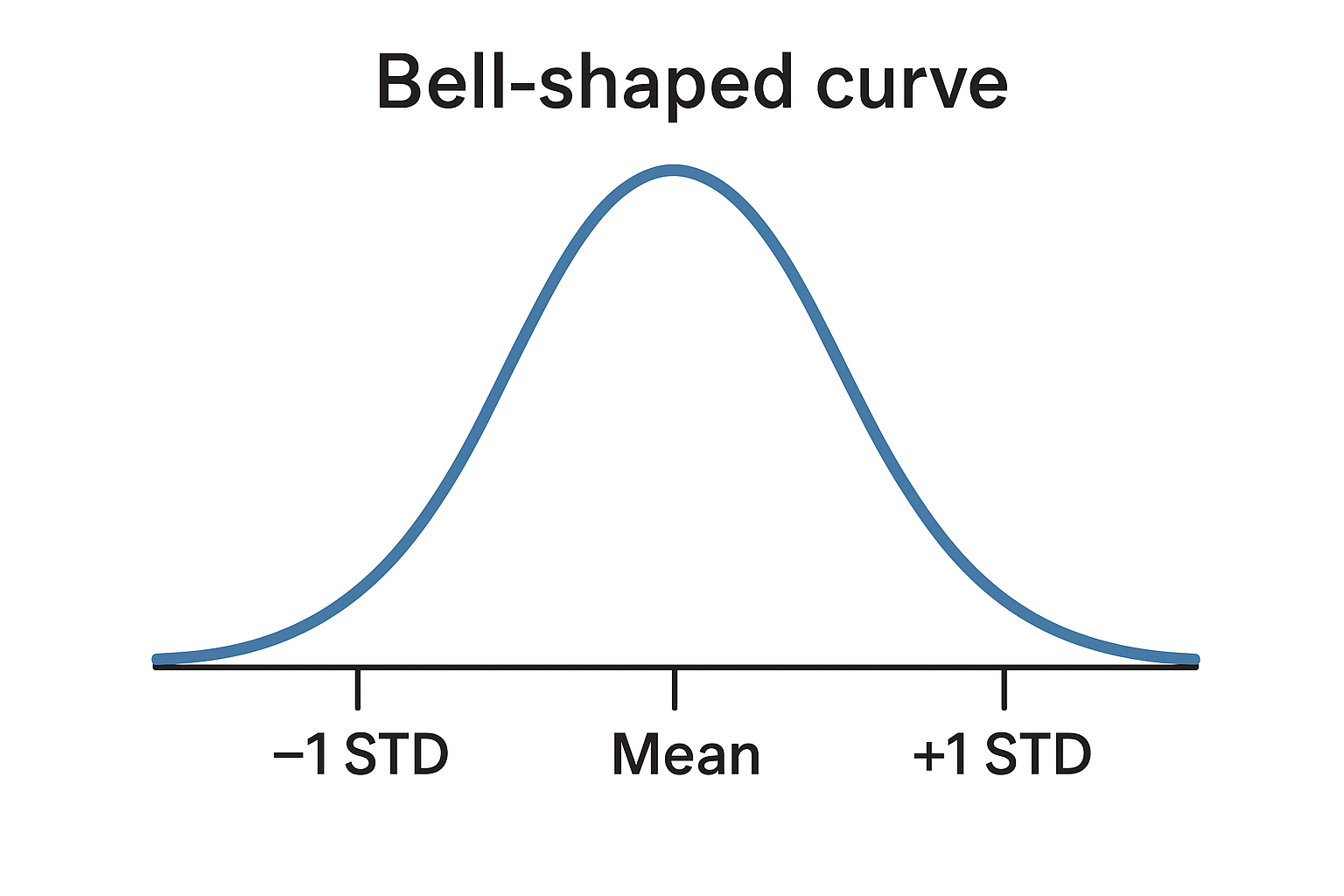
Where -

xxx = original value

μ = mean (average) of the dataset

σ = standard deviation of the dataset

Result: Mean = 0, Standard deviation = 1 (values can be negative or positive).



Q2. Explain the difference between one-hot encoding and label encoding.

Ans –

1. Label Encoding

What it does:  
Assigns each category a unique integer.

When to use:  
Works well for ordinal data (where order matters, like small < medium < large).

Danger:  
If the data is nominal (no order), algorithms might mistakenly think larger numbers mean “higher” or “better”.

| **Fruit** | **Encoded** |
| --- | --- |
| Apple | 0 |
| Banana | 1 |
| Cherry | 2 |

1. One-Hot Encoding

What it does:  
Creates a new column for each category, with 1 where that category is present and 0 elsewhere.

When to use:  
Works best for nominal data (no inherent order), so the model doesn’t get confused by numeric ranking.

Downside:  
Can make your dataset much larger if you have many categories.

| **Fruit** | **Apple** | **Banana** | **Cherry** |
| --- | --- | --- | --- |
| Apple | 1 | 0 | 0 |
| Banana | 0 | 1 | 0 |
| Cherry | 0 | 0 | 1 |

Q3. What is the use of fillna , dropna , and getdummies function in python?

Ans-

| **Function** | **What it does** | **Example use case** |
| --- | --- | --- |
| fillna() | Replace missing values | Fill missing ages with 0 or mean |
| dropna() | Remove rows/columns with missing values | Remove incomplete survey responses |
| get\_dummies() | One-hot encode categorical columns | Convert "Male/Female" to 0/1 columns |

Those fillna and dropna only work and replace or delete / remove missing values i.e. NaN other