1. Feature Engineering: Feature engineering is the process of transforming raw data into meaningful features that improve the performance of machine learning models.

Types of Feature Engineering-

1. Feature Creation: Creating new features from existing ones.

Ex.

From date: Extract year, month, weekday, is\_weekend

From name: Create features like name\_length, has\_title

From text: Compute word\_count, sentiment\_score

1. Feature Transformation: Modify feature to reveal better patterns

Ex.

Apply **log, square root, or exponential** to reduce skew.

Convert price to log(price) if the distribution is right-skewed.

Normalize features to make relationships linear.

1. Feature Extraction: Deriving new features from complex data like text, images, etc.

Ex.

**Text**: Use TF-IDF, embeddings, or topic models.

**Images**: Extract edges, textures, CNN-based embeddings.

**Audio**: Use MFCCs or frequency domain features.

1. Feature Encoding: Converting categorical data into numeric format.

Techniques:

**One-hot encoding**

**Label encoding**

**Target encoding**

**Frequency encoding**

1. Feature Binning/Discretization: Converts continuous variables into categorical bins.

Ex.

Convert age into bins like [0–18], [19–35], [36–60], [60+]

Useful in decision trees or for introducing nonlinearity

1. Interaction Feature: Combine multiple features to capture interactions.

Ex.

Multiply unit\_price × quantity to get total\_value

Concatenate text fields to capture joint semantics

1. Handling Missing Values: Replace missing values with mean/median/mode. Add a binary indicator: is\_missing

2. Confusion matrix: A confusion matrix is a 2x2 table used to evaluate the performance of a classification model.

|  | Predicted Positive | Predicted Negative |
| --- | --- | --- |
| Actual Positive | True Positive (TP) | False Negative (FP) |
| Actual Negative | False Positive (FP) | True Negative (TN) |

Metrics Derived:

**Accuracy** = (TP + TN) / (TP + TN + FP + FN)

**Precision** = TP / (TP + FP) → How many predicted positives are real

**Recall (Sensitivity)** = TP / (TP + FN) → How many actual positives are caught

**F1 Score** = Harmonic mean of Precision and Recall

3. Bais: Error due to overly simplistic assumptions in the model.

**High bias** leads to **underfitting** — the model can't capture the underlying patterns.

Ex.

A linear model trying to fit a nonlinear pattern.

4. Variance: Error due to model sensitivity to small fluctuations in the training data.

**High variance** leads to **overfitting** — the model learns noise and performs poorly on new data.

**Example**: A deep decision tree trained on small dataset.

5. Bias-Variance Tradeoff: You can't simultaneously minimize both bias and variance:

* **High Bias ↔ Low Variance**: Simple model, poor performance
* **Low Bias ↔ High Variance**: Complex model, unstable predictions
* **Goal**: Find a balance that minimizes total error (bias² + variance + irreducible noise)

6. Centre Limit Theorem: The sum of a **large enough sample size** from **any population** with a **finite mean and variance** will be **approximately normally distributed**, regardless of the population’s original distribution.

Properties: The **mean of the sampling distribution** = population mean (μ)

The **standard deviation of the sampling distribution** = σ /√n

where:

* σ = standard deviation of the population
* n = sample size

As **n increases**, the approximation gets better