Step 1: Define the Problem

- Identify what you want to predict (e.g., predicting house prices, classifying emails as spam or not).
- Decide on the type of machine learning: **Supervised** (labeled data) or **Unsupervised** (no labels).

Step 2: Collect and Prepare Data

- Gather relevant data (e.g., CSV files, databases, APIs).
- Clean the data (handle missing values, remove duplicates).
- Convert categorical data (e.g., "Male", "Female") into numerical form.

Step 3: Split the Data

- Divide the dataset into:
    - **Training set** (used to train the model) – 70-80% of data
    - **Test set** (used to evaluate the model) – 20-30% of data

Step 4: Choose a Model

- Select an algorithm based on your problem type:
    - **Regression** (predicting numbers): Linear Regression
    - **Classification** (predicting categories): Decision Tree, Random Forest, Logistic Regression
    - **Clustering** (grouping similar data): K-Means
    - **Deep Learning** (complex problems like image recognition): Neural Networks

Step 5: Train the Model

- Feed the training data into the model.

- Adjust parameters to minimize errors.
- Use techniques like **Gradient Descent** to improve performance.

Step 6: Evaluate the Model

- Test the model on unseen data (test set).
- Measure performance using metrics:
    - **Accuracy** (for classification)
    - **Mean Squared Error (MSE)** (for regression)
    - **Precision & Recall** (for imbalanced data)

Step 7: Improve the Model

- Tune hyperparameters (e.g., learning rate, number of trees in a Random Forest).
- Use feature engineering (create new meaningful features).
- Apply cross-validation (train the model on different data splits).

Step 8: Deploy the Model

- Save the trained model.
- Deploy it in an application (e.g., a website, mobile app).
- Continuously monitor and update the model with new data.

Splitting the Data in Machine Learning

When building a machine learning model, we need to **split the dataset** into two parts:

1. **Training Set** (70-80% of the data)
    - Used to train the model
    - The model learns patterns and relationships from this data.
2. **Test Set** (20-30% of the data)
    - Used to evaluate how well the model performs on unseen data.
    - Ensures the model does not just memorize the training data (avoids overfitting).

Why Do We Split the Data?

1. **Prevent Overfitting**: If we train on 100% of the data, the model might just memorize the data instead of learning patterns.
2. **Assess Performance**: The test set helps us measure how well the model generalizes to new, unseen data.
3. **Avoid Data Leakage**: If we evaluate the model on the same data it was trained on, we get misleadingly high accuracy.

How to Split Data in Python

use **train_test_split** from the sklearn.model_selection module.

How Does the Split Work?

- train_test_split(X, y, test_size=0.2, random_state=42)
    - **X** → Features (independent variables).

- **y** → Target (dependent variable).
- **test_size=0.2** → 20% of data is for testing.
- **random_state=42** → Ensures the same split every time for reproducibility.

Training set → Helps the model learn.
Test set → Evaluates model performance.
Use train_test_split to split data easily.
Keep random_state fixed for consistent results.

Variable	**Meaning**
X_train	Training data (features) – used to train the model (80%)
X_test	Testing data (features) – used to test the model (20%)
y_train	Training labels (actual answers for training data)
y_test	Testing labels (actual answers for testing data)