Machine Learning Workflow

By Anshum Banga; www.github.com/anshumbanga; www.linkedin.com/in/anshum-banga

Step	Stage	Explanation	Example
1	**Problem Understanding**	Clearly define the business or real-world problem, objectives, and success criteria. Decide whether the task is regression, classification, or clustering.	Predict whether a customer will churn (Yes/No) → Classification problem
2	**Data Collection**	Gather relevant data from databases, APIs, files, sensors, or surveys. The quality of data directly affects model performance.	Customer data collected from CRM, Excel files, or SQL database
3	**Data Cleaning**	Handle missing values, incorrect entries, duplicates, and outliers to make data reliable.	Filling missing age values with mean, removing duplicate rows
4	**Exploratory Data Analysis (EDA)**	Analyze data using statistics and visualizations to understand patterns, trends, and relationships.	Using histograms to study age distribution, heatmap for correlations
5	**Feature Engineering**	Select, transform, or create new features to improve model learning.	Converting “Gender” to numeric, scaling income, creating Total_Spend
6	**Train–Test Split**	Split data into training and testing sets to evaluate model performance on unseen data.	80% data for training, 20% for testing
7	**Model Training**	Train the selected machine learning algorithm on training data so it can learn patterns.	Training Logistic Regression or Random Forest model
8	**Model Evaluation**	Evaluate model performance using appropriate metrics based on problem type.	Accuracy, Precision, Recall for classification
9	**Hyperparameter Tuning**	Optimize model performance by tuning hyperparameters using techniques like GridSearch or RandomSearch.	Finding best depth for Decision Tree
10	Final Model	Select the best performing model and prepare it for deployment or real-world use.	Deploy churn prediction model in production

Key Concepts

Features & Labels

Features (X) — Independent Variables

Features are the **input variables** used by a machine learning model to make predictions.

- Also called: **Independent variables**, **predictors**, or **inputs**
- Represent the **cause or influencing factors**
- Written as **X**

Why "independent"?
Because they **do not depend on the output**.
They exist on their own and help explain or predict the output.

Example:

House Price Prediction

Feature (X)	Meaning
Area	Size of the house
BHK	Number of rooms
Location	Area of city
Parking	Availability

Label (y) — Dependent Variable

Label is the **output variable** that the model is trying to predict.

- Also called: **Dependent variable**, **target**, or **output**
- Written as **y**
- Depends on features (X)

Why "dependent"?
Because its value **depends on the input features**.

Example:

- **House Price** → Label (y)
- **Spam / Not Spam** → Label (y)

Relationship

```
scss

X (Features)  ——→  y (Label)
Cause              Effect
```

What is Train–Test Split?

Train–Test Split is the process of dividing the dataset into **two separate parts**:

1. **Training Data** → Used to train the model
2. **Testing Data** → Used to evaluate model performance

Key Idea:

A model must be tested on **unseen data** to check if it generalizes well.

Training Data

Purpose:

- Used by the algorithm to **learn patterns**
- Model adjusts its internal parameters here

Components:

- **X_train** → Features used for training
- **y_train** → Labels used for training

📌 Example:

- Model learns how *area, location, BHK* affect *house price*

Testing Data

Purpose:

- Used to **evaluate performance**
- Model has **never seen this data before**

Components:

- **X_test** → Features for testing
- **y_test** → True labels for comparison

📌 We compare:

```
Predicted y  vs  Actual y_test
```

Why Do We Split Data?

Without split ❌

- Model memorizes data
- High accuracy but poor real-world performance
- Overfitting risk

With split ✅

- Fair evaluation
- Checks **generalization**
- Mimics real-world usage

Common Split Ratios — WHY 70–30 or 80–20?

80–20 Split

- 80% → Training
- 20% → Testing

Used when:

- Dataset is medium or large
- Need strong learning + reliable evaluation

70–30 Split

- 70% → Training
- 30% → Testing

Used when:

- Dataset is small
- More testing data needed for evaluation confidence

Logic Behind Ratios

- **More training data** → Better learning
- **Enough testing data** → Reliable performance check

Final Structure After Split

Data Type	Features	Labels
Training Data	X_train	y_train
Testing Data	X_test	y_test