CROSS VALIDATION

Cross-validation is a **model evaluation technique** used to estimate how well a machine learning model will generalize to unseen data.

Instead of training the model once and testing on a single hold-out set, CV splits the dataset into **multiple training–testing cycles** and averages the results.

Reduces **overfitting risk** by checking performance on multiple unseen splits.

Why Cross-Validation is Needed

Without CV, you might:

- Overestimate accuracy if you test on data that's too similar to training data.
- Underestimate accuracy if you test on an unlucky "bad" split.

Cross-validation ensures:

- **Stability** of performance estimate.
- Fair comparison between models.

Common Types of Cross-Validation

The three steps involved in cross-validation are as follows:

- Reserve some portion of sample data-set.
- Using the rest data-set train the model.

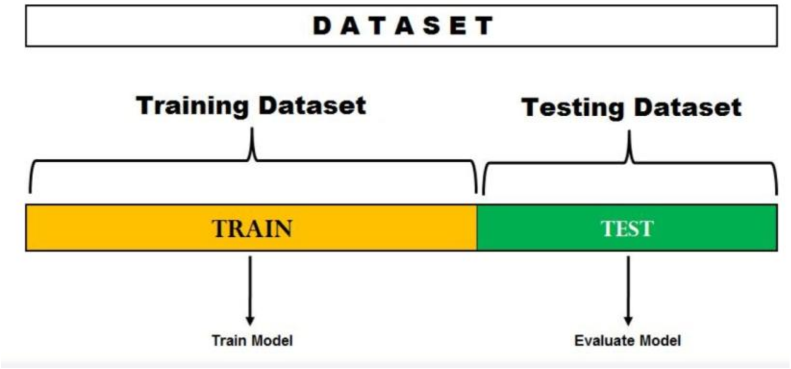
Test the model using the reserve portion of the data-set.

Hold out method

Here the entire dataset(population) is divided into 2 sets – train set and test set.

The data can be divided into two sets with any proportion depending on the use case.

The proportion of training data has to be larger than the test data.



Benefits of Holdout Method

□ **Simplicity & Speed**

- Easiest to implement.
- Fast → only one model training cycle.

□ **Less Computational Cost**

Drawbacks

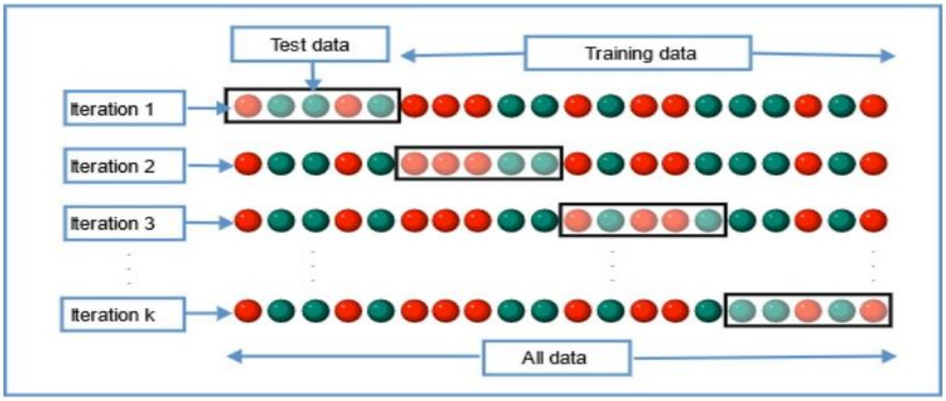
1. **High Variance**
   - Performance depends heavily on how the data is split.
   - Different splits → different results.
2. **Inefficient Use of Data**
   - Test data isn't used for training at all.
   - In small datasets, this wastes valuable data.
3. **Risk of Bias**
   - If split isn't random or balanced, the test set may not represent the real-world distribution.
4. **Overfitting Risk**
   - If you tune hyperparameters repeatedly on the test set → you "leak" information and overfit to test data.

When to Use Holdout Method

- ✓ When you have a **very large dataset** (millions of rows).
- ✓ When you need a **quick, rough estimate** of performance.

(a) k-Fold Cross-Validation

- **Process:**
    1. Split data into k roughly equal folds (e.g., k=5).
    2. Train on k-1 folds, test on the remaining fold.
    3. Repeat k times (each fold is test once).
    4. Average the scores.



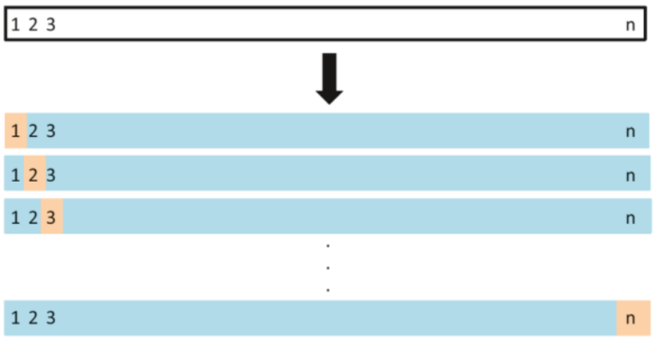
- **When to use:**
    - Default choice for most models & datasets.
    - Works well if data is **not time-dependent**.
- **Example:** 5-fold CV → 80% train, 20% test each iteration.

(b) Stratified k-Fold Cross-Validation

- **Process:** Same as k-fold, **but keeps class proportions the same** in each fold.
- **When to use:**
    - For classification problems with **imbalanced classes**.
    - Prevents bias toward majority class.

(c) Leave-One-Out Cross-Validation (LOOCV)

- **Process:**
    - k = number of samples in dataset.
    - Train on all but 1 sample, test on that 1 sample. Repeat for all samples.



- **When to use:**
  - Very small datasets (e.g., <100 samples).
- **Downside:** Very slow for large datasets.

(d) Leave-p-Out Cross-Validation

- **Process:** Similar to LOOCV but leaves **p samples out** for testing each time.
- **When to use:**
  - Theoretical analysis or very small datasets.
- **Rarely used** in practice (computationally heavy).

How to Choose Which Cross-Validation Type

Situation	Recommended CV Type
General tabular data	k-Fold CV
Imbalanced classification	Stratified k-Fold CV
Very small dataset	LOOCV or Repeated k-Fold CV
Time series data	Time Series CV

Best Practices

- Always perform **data preprocessing inside each fold** to avoid leakage.
- For imbalanced datasets → **use stratification**.
- For time series → **never shuffle data**; use sequential CV.
- Choose k=5 or k=10 for most cases (balance between bias & variance).
- Use **Repeated k-Fold** if you want more confidence in your score.

1. Holdout Method

📌 Scenario:

You are building a movie recommendation system using a dataset of **10 million user ratings**. You try a **70% training / 30% test split**.

Question:

Would the holdout method be appropriate here? Why or why not?

Expected Reasoning:

- Dataset is very large → 30% (3 million ratings) is still representative.
- Holdout method is efficient and less computationally expensive → **Yes, appropriate.**

---

2. Holdout vs Cross-Validation

📌 Scenario:

You are training a model on a **medical dataset with only 500 patient records**. You use a **70/30 holdout split** and get 92% accuracy.

Question:

Should you trust this accuracy estimate? What would be a better validation technique?

Expected Reasoning:

- Small dataset → holdout wastes valuable data & high variance results.
- Better: **k-Fold CV** (like 5-fold or 10-fold) for a more stable estimate.