Introduction to Machine Learning

- Prof. Balaraman Ravindran | IIT Madras

Problem Solving Session (Week-7)

Shreya Bansal

PMRF PhD Scholar
IIT Ropar

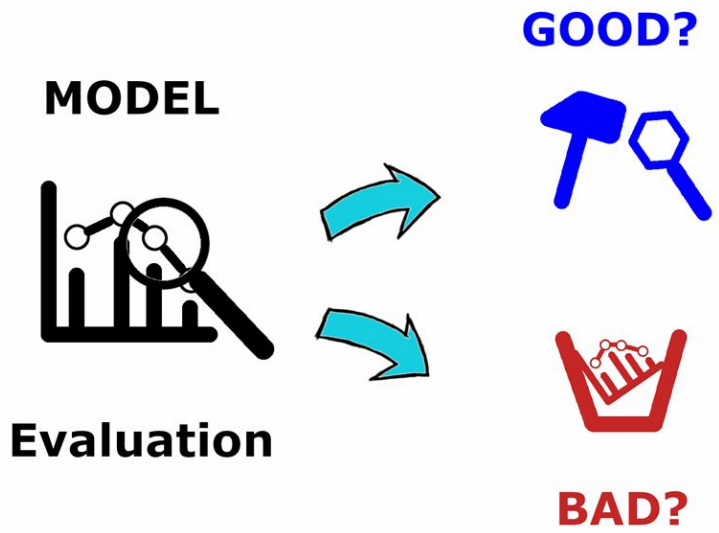
Week-7 Contents

---

1. Evaluation Measures
2. Cross Validation
3. ROC Curve
4. Ensemble Methods

Evaluation and Evaluation Measures

- Measuring how well a model performs on given data.
- Ensures the model generalizes well beyond training data.



Common Classification Evaluation Measures

- **Misclassification Error (0-1 Loss)**

  Fraction of incorrect predictions.

- **Cross-Entropy Loss**

  Measures how different the predicted probabilities are from actual labels.

- **Accuracy**

  (Correct Predictions) / (Total Predictions).

Regression Evaluation Measures

- **Mean Squared Error (MSE)**

  Penalizes larger errors more.

- **Root Mean Squared Error (RMSE)**

  Square root of MSE, interpretable in original units.

- **Mean Absolute Error (MAE)**

  Average of absolute differences between predictions and actual values.

Evaluating the Classifier’s Performance

- **Goal: Measure performance on the entire data distribution.**
- **We don’t know the full data distribution, only have samples.**
- **Use training and test data to estimate true performance.**

Train-Test Split and Bias

- **Train-Test Split:**
  - **Training set: Used for learning model parameters.**
  - **Test set: Used for final evaluation.**
- **Potential Issues:**
  - **Test data may not be independent of training data.**
  - **Training data may be biased (not representative of real-world distribution).**

Cross-Validation

- Single train-test split may not generalize well.
- Instead, use k-Fold Cross-Validation:
  - Split data into k subsets.
  - Train on k-1 subsets, test on the remaining subset.
  - Repeat for all subsets and average results.

Overfitting and Generalization

- **Overfitting: Model performs well on training data but poorly on unseen data.**
- **Regularization techniques:**
    - **L1/L2 Regularization.**
    - **Data augmentation.**
    - **Dropout (for deep learning).**

Active Learning

- **When Training Data is Not Representative:**
    - **Sample additional data from regions with high uncertainty.**
    - **Model actively queries for more samples in critical areas.**
    - **Helps reduce bias in data.**

Ensemble Learning & Weak Classifiers

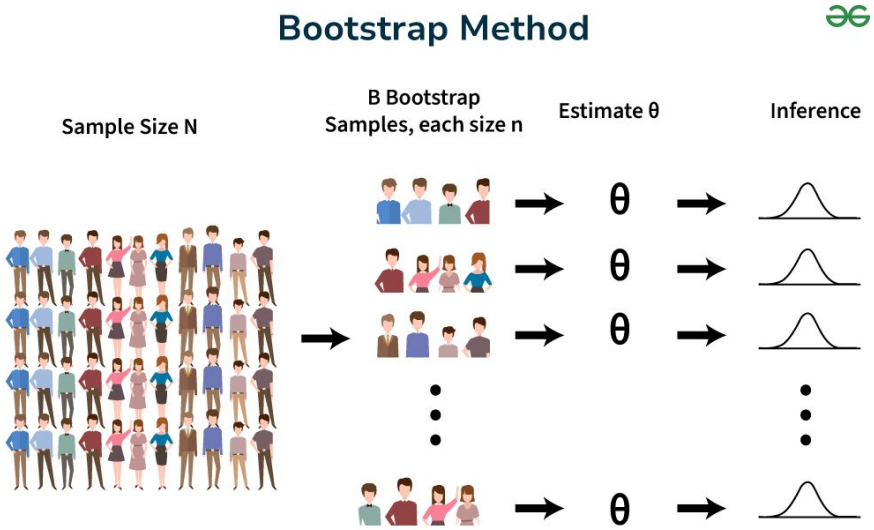
- **Weak Classifiers:**
    - **Classifiers slightly better than random (e.g., 51% accuracy in a binary task).**
- **Boosting (e.g., AdaBoost):**
    - **Combines weak classifiers to form a strong model.**
- **Bagging (e.g., Random Forests):**
    - **Reduces variance by averaging multiple models.**

Bootstrap and Cross-Validation

- Why do we need sampling methods?
- The goal: Estimating performance metrics accurately
- Two popular techniques: Bootstrap & Cross-Validation

What is Bootstrap?

- A simple but powerful resampling technique
- Key Idea: Sample from the dataset with replacement
- Helps estimate variance and classification error
- Generates multiple bootstrap samples (S′, S″, S‴, ...)



Bootstrap Sampling Process

- Start with a dataset of size N
- Sample N elements with replacement → S′
- Repeat the process L times to create multiple bootstrap samples
- Use S′ for training, and the remaining data for testing
- Average the results → Bootstrap estimate

Why Use Bootstrap?

- Reduces variance in error estimation
- Works well with large datasets
- Provides a better approximation of the underlying distribution
- Used for performance estimation, variance reduction, and parameter estimation

Bootstrap Error Estimation

- Train on S', test on $S - S'$
- Repeat for multiple bootstrap samples
- Take the average error estimate
- Lower variance compared to random train-test splits

The 0.632 Bootstrap Method

- On average, 63.2% of data appears in any given bootstrap sample
- The remaining 36.8% is left out
- Weighted estimate:

$$E_{0.632} = 0.632 \times E_{out} + 0.368 \times E_{in}$$

The 0.632 Bootstrap Method

Why 0.632?

- Each data point has a probability of $\frac{1}{N}$ of being selected in one draw.
- The probability of NOT being selected in a single draw is $1 - \frac{1}{N}$.
- Over N draws, the probability of a data point never being selected is:

$$\left(1 - \frac{1}{N}\right)^N$$

As $N \to \infty$, this approaches $e^{-1} \approx 0.368$, meaning about **36.8%** of data points are left out (OOB), and **63.2%** are included.

What is Cross-Validation?

- A method to evaluate model performance
- Used when data is limited
- K-Fold Cross-Validation:
    - Split data into K equal parts
    - Train on K-1 parts, test on the remaining 1 part
    - Repeat for all K parts and average results

Comparing Bootstrap & Cross-Validation

Feature	Bootstrap	Cross-Validation
Sampling	With replacement	Without replacement
Data Size	Needed Large dataset	Works well for small datasets
Variance Reduction	Yes	Yes, depends on K
Use Case	Error estimation, variance estimation	Model evaluation, hyperparameter tuning

Leave-One-Out Cross-Validation (LOO-CV)

- Special case of K-Fold CV where K = N
- Train on N-1 samples, test on the left-out sample
- Pros: Uses all data for training
- Cons: Computationally expensive

Challenges & Best Practices

- **Bootstrap:**
  - **Requires a large sample size**
  - **Can overestimate variance for small datasets**
- **Cross-Validation:**
  - **Choose K = 5 or 10 for optimal balance**
  - **Ensure stratified sampling to maintain class distributions**

2-Class Evaluation Measures

- Classification is a fundamental task in machine learning.
- We focus on two-class (binary) classification problems.
- Many multiclass metrics are extensions of these.

Confusion Matrix

- **Definition: A table summarizing predictions vs. actual values.**
- **Helps understand model performance.**

Actual → Predicted ↓	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Accuracy & Misclassification Error

- **Accuracy = (TP + TN) / (Total samples)**
- **Misclassification Error = 1 - Accuracy**
- **Works well when class distribution is balanced.**
- **Fails in highly imbalanced datasets.**

Actual → Predicted ↓	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Precision

- **Measures correctness of positive predictions.**
- **Formula: Precision = TP / (TP + FP)**
- **High precision → Fewer false positives.**
- **Important when false positives are costly (e.g., medical diagnosis).**

Actual → Predicted ↓	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Recall (Sensitivity)

- **Measures how many actual positives are correctly identified.**
- **Formula: Recall = TP / (TP + FN)**
- **High recall → Fewer false negatives.**
- **Critical when missing positives is costly (e.g., cancer detection).**

Actual → Predicted ↓	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Precision-Recall Tradeoff

High Precision → Lower Recall

High Recall → Lower Precision

Example: Spam detection vs. fraud detection.

F1-Score (Harmonic Mean of Precision & Recall)

- **Formula: F1 = 2 × (Precision × Recall) / (Precision + Recall)**
- **Balances precision and recall.**
- **Useful for imbalanced datasets.**

Why Accuracy Fails in Imbalanced Data

- Example: 99% negative class, 1% positive class.
- Predicting all negatives → 99% accuracy but useless model.
- Use precision, recall, and F1-score instead.

Summary

Confusion Matrix: Core performance table.

Accuracy: Good for balanced classes.

Precision & Recall: Useful for imbalanced classes.

F1-Score: Tradeoff metric.

Choose the right metric for your application!

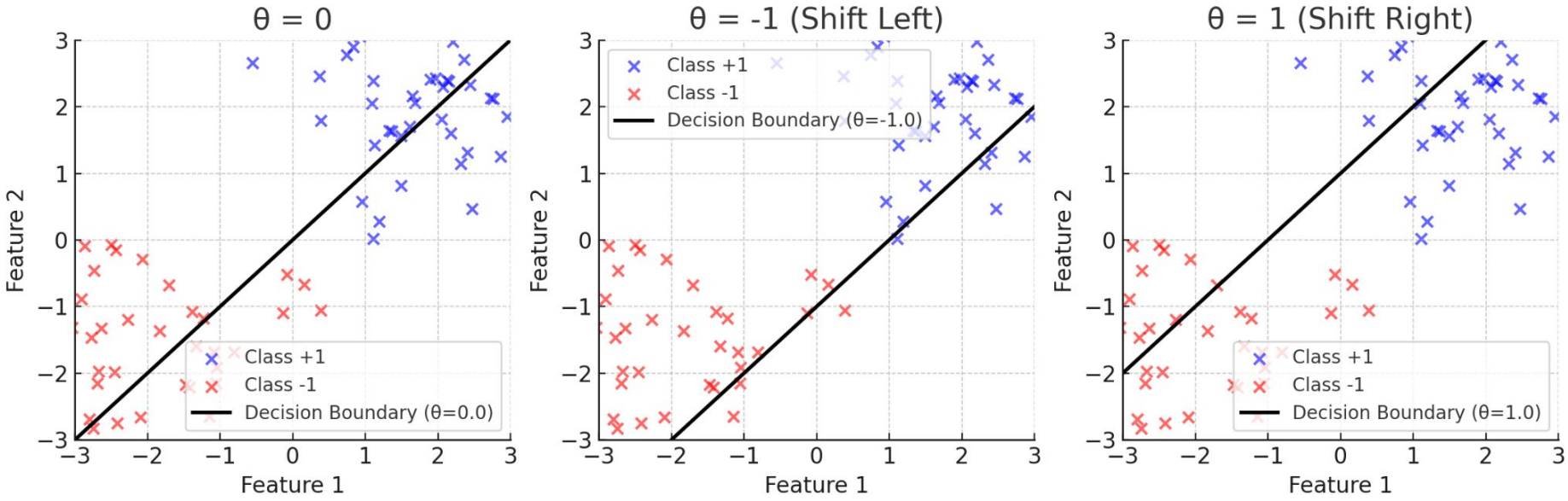
Classifier Decision Boundaries

- Classifiers assign labels based on a discriminant function $\Delta(x)$
- For a two-class problem:
- $\Delta(x)<0 \rightarrow$ Class A
- $\Delta(x)>0 \rightarrow$ Class B
- Instead of using 0 as a threshold, we introduce θ to shift decision boundaries.

Effect of Changing θ

- Adjusting θ shifts the decision boundary.
- This impacts classification:
    - Increases or decreases false positives and false negatives.
    - Allows tuning of classifier performance for specific goals (precision, recall, etc.).

Visualizing Decision Boundaries



Evaluating Classifier Performance

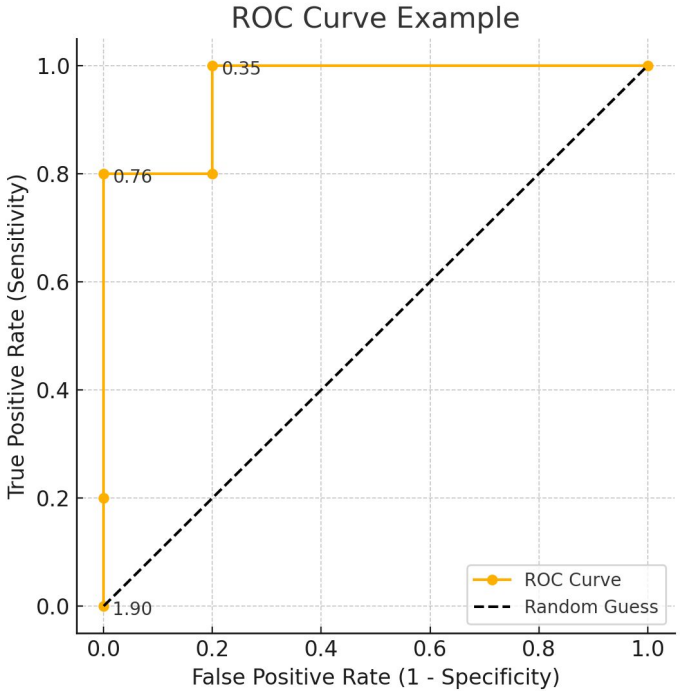
- **Need a systematic way to evaluate classifier performance.**
- **Common metrics:**
  - **True Positive Rate (TPR) = True Positives/Total Positives**
  - **False Positive Rate (FPR) = False Positives/Total Negatives**
- **These metrics are used to plot the ROC Curve.**

Actual → Predicted ↓	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

ROC Curve Introduction

- **Receiver Operating Characteristic (ROC) curve plots:**
    - **X-axis: False Positive Rate (FPR)**
    - **Y-axis: True Positive Rate (TPR)**
- **Shows the trade-off between sensitivity (TPR) and specificity (1 - FPR).**

ROC Curve Example



Interpreting the ROC Curve

- Ideal classifier: Steep rise towards (0,1), then flattens.
- Random classifier: Diagonal line (random guessing).
- Higher ROC curve = better classifier.
- ROC curves help choose optimal decision thresholds.

Area Under Curve (AUC)

- AUC (Area Under the Curve) quantifies classifier performance.
- Ranges from 0.5 (random) to 1.0 (perfect).
- Higher AUC = better classifier performance.

Practical Use of ROC & AUC

- ROC and AUC help compare multiple classifiers.
- Used in medical diagnosis, fraud detection, image classification, etc.
- Key takeaway: Choose the right threshold θ based on application needs.

Example

Consider a binary classification task with **20 samples**:

Sample	True Label (y)	Predicted Label (ŷ)	Probability (ŷ_prob)
1	1	1	0.95
2	1	1	0.90
3	1	1	0.85
4	1	0	0.45
5	1	0	0.40
6	1	1	0.92
7	1	1	0.88
8	1	0	0.30
9	1	0	0.25
10	0	1	0.70
11	0	1	0.60
12	0	1	0.55
13	0	0	0.40
14	0	0	0.35
15	0	0	0.30
16	0	0	0.25
17	0	0	0.20
18	0	0	0.15
19	0	0	0.10
20	0	0	0.05

Example

Step 1: Compute Confusion Matrix

We count **True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN).**

	Predicted Positive ($\hat{y}=1$)	Predicted Negative ($\hat{y}=0$)
Actual Positive (y=1)	TP = 5	FN = 4
Actual Negative (y=0)	FP = 3	TN = 8

- **TP = 5** → Correctly classified as positive
- **FN = 4** → Incorrectly classified as negative
- **FP = 3** → Incorrectly classified as positive
- **TN = 8** → Correctly classified as negative

Example

Step 2: Compute Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{5 + 8}{5 + 8 + 3 + 4} = \frac{13}{20} = 0.65$$

Step 3: Compute Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$= \frac{5}{5 + 3} = \frac{5}{8} = 0.625$$

Example

Step 4: Compute Recall (Sensitivity)

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$= \frac{5}{5+4} = \frac{5}{9} = 0.556$$

Step 5: Compute F1-score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$= 2 \times \frac{0.625 \times 0.556}{0.625 + 0.556}$$

$$= 2 \times \frac{0.347}{1.181} = 0.588$$

Example

Step 6: Compute ROC Curve

The **ROC Curve** plots the **True Positive Rate (TPR) vs. False Positive Rate (FPR)** for different probability thresholds.

Formulas

$$\text{TPR} = \frac{TP}{TP + FN} = \text{Recall}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

Sorting by predicted probability:

Threshold	TP	FN	FP	TN	TPR (Recall)	FPR
0.95	1	8	0	9	0.111	0.00
0.92	2	7	0	9	0.222	0.00
0.90	3	6	0	9	0.333	0.00
0.88	4	5	0	9	0.444	0.00
0.85	5	4	0	9	0.556	0.00
0.70	5	4	1	8	0.556	0.111
0.60	5	4	2	7	0.556	0.222
0.55	5	4	3	6	0.556	0.333

Consider a binary classification task with **20 samples**:

Sample	True Label (y)	Predicted Label (ŷ)	Probability (ŷ_prob)
1	1	1	0.95
2	1	1	0.90
3	1	1	0.85
4	1	0	0.45
5	1	0	0.40
6	1	1	0.92
7	1	1	0.88
8	1	0	0.30
9	1	0	0.25
10	0	1	0.70
11	0	1	0.60
12	0	1	0.55
13	0	0	0.40
14	0	0	0.35
15	0	0	0.30
16	0	0	0.25
17	0	0	0.20
18	0	0	0.15
19	0	0	0.10
20	0	0	0.05

Example

Consider a binary classification task with **20 samples**:

Sample	True Label (y)	Predicted Label (ŷ)	Probability (ŷ_prob)
1	1	1	0.95
2	1	1	0.90
3	1	1	0.85
4	1	0	0.45
5	1	0	0.40
6	1	1	0.92
7	1	1	0.88
8	1	0	0.30
9	1	0	0.25
10	0	1	0.70
11	0	1	0.60
12	0	1	0.55
13	0	0	0.40
14	0	0	0.35
15	0	0	0.30
16	0	0	0.25
17	0	0	0.20
18	0	0	0.15
19	0	0	0.10
20	0	0	0.05

Step 2: Sort the Predictions by Probability

We sort the samples by their predicted probability (`ŷ_prob`) in **descending** order.

Sample	True Label (y)	Predicted Probability (ŷ_prob)
1	1	0.95
6	1	0.92
2	1	0.90
7	1	0.88
3	1	0.85
10	0	0.70
11	0	0.60
12	0	0.55
4	1	0.45
5	1	0.40
13	0	0.40
14	0	0.35
8	1	0.30
15	0	0.30
16	0	0.25
9	1	0.25
17	0	0.20
18	0	0.15
19	0	0.10
20	0	0.05

Formula for Classification at a Threshold `t`:

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{y}_{\text{prob}} \geq t \\ 0 & \text{otherwise} \end{cases}$$

Example

- If ŷ_prob ≥ **threshold**, predict **1 (positive)**.
- If ŷ_prob < **threshold**, predict **0 (negative)**.

Step 2: Sort the Predictions by Probability

We sort the samples by their predicted probability (`ŷ_prob`) in **descending** order.

Sample	True Label (y)	Predicted Probability (ŷ_prob)
1	1	0.95
6	1	0.92
2	1	0.90
7	1	0.88
3	1	0.85
10	0	0.70
11	0	0.60
12	0	0.55
4	1	0.45
5	1	0.40
13	0	0.40
14	0	0.35
8	1	0.30
15	0	0.30
16	0	0.25
9	1	0.25
17	0	0.20
18	0	0.15
19	0	0.10
20	0	0.05

Step 2: Compute TPR and FPR at Each Threshold

We go through the list and count **True Positives (TP)**, **False Positives (FP)**, **False Negatives (FN)**, and **True Negatives (TN)**.

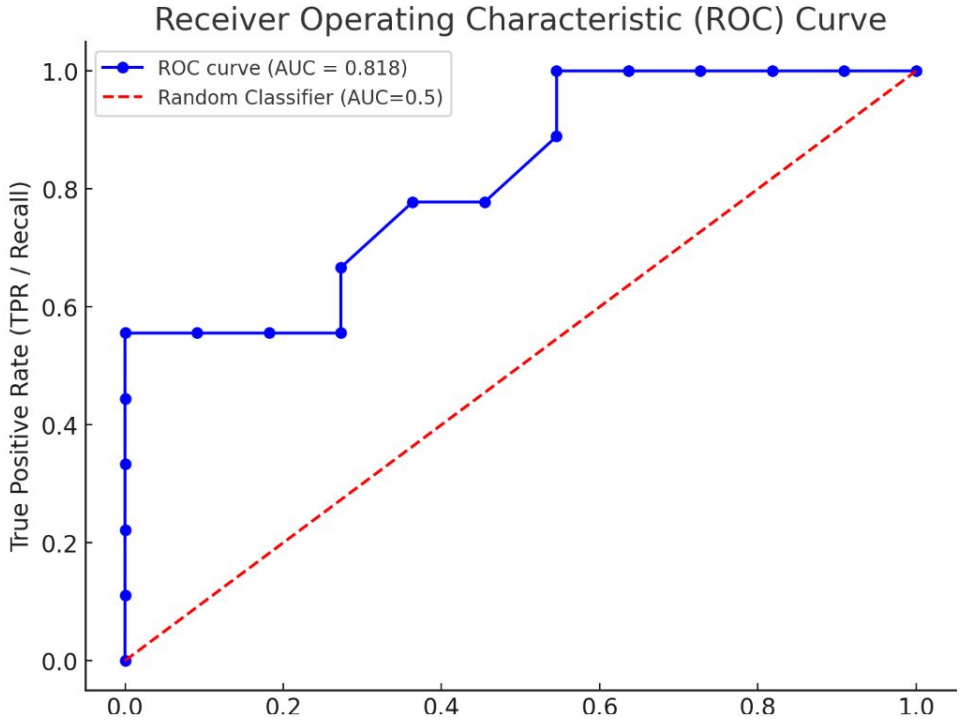
Threshold	TP	FN	FP	TN	TPR (Recall)	FPR
1.00	0	9	0	11	$\frac{0}{9} - 0.000$	$\frac{0}{11} - 0.000$
0.95	1	8	0	11	$\frac{1}{9} - 0.111$	$\frac{0}{11} - 0.000$
0.92	2	7	0	11	$\frac{2}{9} - 0.222$	$\frac{0}{11} - 0.000$
0.90	3	6	0	11	$\frac{3}{9} - 0.333$	$\frac{0}{11} - 0.000$
0.88	4	5	0	11	$\frac{4}{9} - 0.444$	$\frac{0}{11} - 0.000$
0.85	5	4	0	11	$\frac{5}{9} - 0.556$	$\frac{0}{11} - 0.000$
0.70	5	4	1	10	$\frac{5}{9} - 0.556$	$\frac{1}{11} - 0.091$
0.60	5	4	2	9	$\frac{5}{9} - 0.556$	$\frac{2}{11} - 0.182$
0.55	5	4	3	8	$\frac{5}{9} - 0.556$	$\frac{3}{11} - 0.273$
0.45	6	3	3	8	$\frac{6}{9} - 0.667$	$\frac{3}{11} - 0.273$
0.40	7	2	4	7	$\frac{7}{9} - 0.778$	$\frac{4}{11} - 0.364$
0.35	7	2	5	6	$\frac{7}{9} - 0.778$	$\frac{5}{11} - 0.455$
0.30	8	1	6	5	$\frac{8}{9} - 0.889$	$\frac{6}{11} - 0.545$
0.25	9	0	6	5	$\frac{9}{9} - 1.000$	$\frac{6}{11} - 0.545$
0.20	9	0	7	4	$\frac{9}{9} - 1.000$	$\frac{7}{11} - 0.636$
0.15	9	0	8	3	$\frac{9}{9} - 1.000$	$\frac{8}{11} - 0.727$
0.10	9	0	9	2	$\frac{9}{9} - 1.000$	$\frac{9}{11} - 0.818$
0.05	9	0	10	1	$\frac{9}{9} - 1.000$	$\frac{10}{11} - 0.909$
0.00	9	0	11	0	$\frac{9}{9} - 1.000$	$\frac{11}{11} - 1.000$

Example

Step 4: Interpret Results

- **AUC Interpretation:**
  - If **AUC = 1**, the classifier perfectly distinguishes between classes.
  - If **AUC = 0.5**, the classifier is no better than random guessing.
  - If **AUC > 0.9**, it's an excellent classifier.
  - If **AUC between 0.7 and 0.9**, it's a fair classifier.
  - If **AUC < 0.7**, it's a poor classifier.
- **Understanding ROC:**
  - A **higher TPR and lower FPR** indicate better performance.
  - A **steep ROC curve** towards the top-left corner is ideal.
  - If the curve is close to the **diagonal line (y = x)**, the classifier performs poorly.

Example


Receiver Operating Characteristic (ROC) Curve
ROC curve (AUC = 0.818)
Random Classifier (AUC=0.5)
True Positive Rate (TPR / Recall)
0.0
0.2
0.4
0.6
0.8
1.0

Example

Manual Computation of AUC Using the Trapezoidal Rule

AUC is computed as the area under the ROC curve using the **Trapezoidal Rule**:

$$AUC = \sum_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \times \frac{(TPR_{i+1} + TPR_i)}{2}$$

Step 1: Given ROC Points

From our previous table:

Threshold	FPR	TPR
1.00	0.000	0.000
0.95	0.000	0.111
0.92	0.000	0.222
0.90	0.000	0.333
0.88	0.000	0.444
0.85	0.000	0.556
0.70	0.111	0.556
0.60	0.222	0.556
0.55	0.333	0.556
0.00	1.000	1.000

Example

Given Data (TPR vs. FPR):

Threshold	TPR	FPR
1.00	0.000	0.000
0.95	0.111	0.000
0.92	0.222	0.000
0.90	0.333	0.000
0.88	0.444	0.000
0.85	0.556	0.000
0.70	0.556	0.091
0.60	0.556	0.182
0.55	0.556	0.273
0.45	0.667	0.273
0.40	0.778	0.364
0.35	0.778	0.455
0.30	0.889	0.545
0.25	1.000	0.545
0.20	1.000	0.636
0.15	1.000	0.727
0.10	1.000	0.818
0.05	1.000	0.909
0.00	1.000	1.000

Step 1: Apply Trapezoidal Rule

The AUC is computed as:

$$AUC = \sum \frac{(x_{i+1} - x_i) \times (y_{i+1} + y_i)}{2}$$

where x values are **FPRs**, and y values are **TPRs**.

$$AUC = \sum_{i=0}^{n-1} \frac{(FPR_{i+1} - FPR_i) \times (TPR_{i+1} + TPR_i)}{2}$$

Computing each trapezoidal segment:

$$AUC = \frac{(0.000 - 0.000) \times (0.111 + 0.000)}{2} + \frac{(0.000 - 0.000) \times (0.222 + 0.111)}{2} + \dots + \frac{(1.000 - 0.909) \times (1.000 + 1.000)}{2}$$

After summing up all values:

$$AUC = 0.818$$

Introduction to MDL

- The Minimum Description Length (MDL) principle is based on the idea that the best model is the one that requires the fewest bits to describe while maintaining good performance.
- It provides a way to balance model complexity and error rate.

MDL and Model Complexity

- Complex models (e.g., deep neural networks, decision trees with many branches, SVMs with many support vectors) require more bits to describe.
- Simpler models require fewer bits but may have higher error rates.
- The goal is to trade-off between model complexity and classification error.

Example – Support Vector Machines (SVMs)

- In SVM, a model is defined by its support vectors and α coefficients.
- The more support vectors, the more bits needed to describe the classifier.
- Kernel-based SVMs add additional complexity because they involve inner product calculations.

The Trade-off in MDL

- More complex models → More bits to describe, fewer errors
- Simpler models → Fewer bits to describe, more errors
- The optimal model minimizes the total description length (model + errors).

MDL as a Bayesian Approach

- MDL is closely related to Bayesian learning.
- It can be derived from Maximum A Posteriori (MAP) estimates.
- Used for complexity and performance evaluation in machine learning.

Introduction to Exploratory Data Analysis (EDA)

- EDA is the first step in machine learning before model selection.
- Helps in understanding:
  - Data distribution
  - Feature importance
  - Missing values & outliers

Why is EDA Important?

Helps avoid blindly applying models.

Ensures data is clean and ready for training.

Prevents unexpected biases and errors in modeling.

Key Techniques in EDA

Descriptive Statistics: Mean, Median, Variance

Visualizations: Histograms, Boxplots, Scatter Plots

Feature Correlation: Identifying relationships between variables

Outlier Detection: Finding extreme values

Experimental Approaches in ML

- **Observation-based experiments: Identify correlations and associations.**
- **Manipulation experiments: Test causal hypotheses by controlling variables.**
- **Example: Comparing two ML models (Algorithm A vs. Algorithm B) under different conditions.**

Bagging (Bootstrap Aggregating)

Uses bootstrap sampling: multiple training sets created by sampling with replacement.

Trains multiple classifiers (same type) on different bootstrap samples.

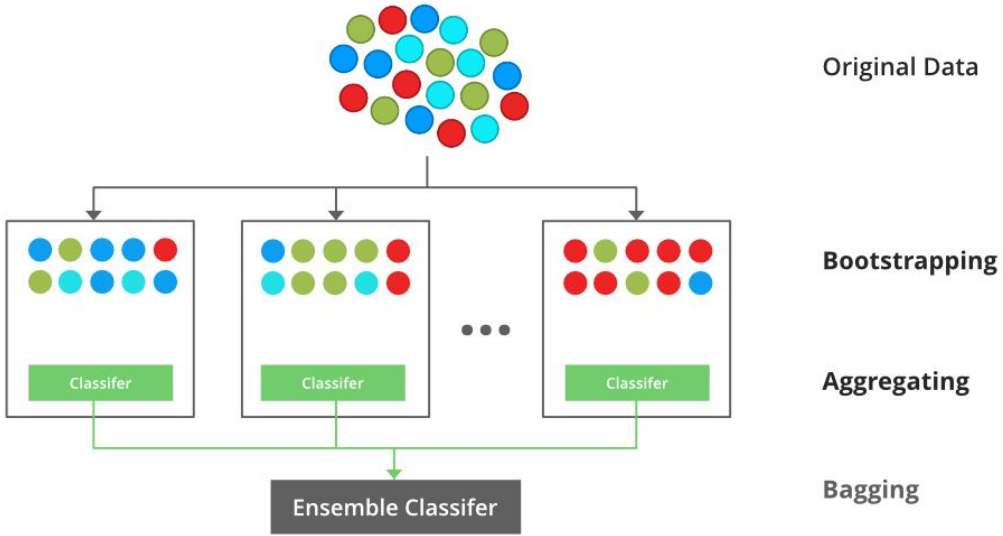
Reduces variance and stabilizes predictions.

Combines results via:

- Majority voting (for classification).
- Averaging predictions (for regression).

Works well with unstable models like decision trees (e.g., Random Forests).

Bagging (Bootstrap Aggregating)



Committee Machines

Uses different classifiers (SVM, neural networks, decision trees, etc.).

Trains them on the same dataset.

Averages predictions or assigns equal weights to each model.

Simple but lacks optimal weighting.

Stacking (Stacked Generalization)

- Instead of equal weights, learns the best way to combine predictions.
- Uses a meta-learner (e.g., logistic regression, another neural network).
- Steps:
    - Train multiple classifiers on the training data.
    - Use their predictions as features for a new model.
    - Train a meta-classifier to learn the optimal way to combine outputs.

How Stacking Works

First Level Classifiers (Base Models)

You train multiple classifiers (e.g., f1, f2, ..., fm) on the dataset.

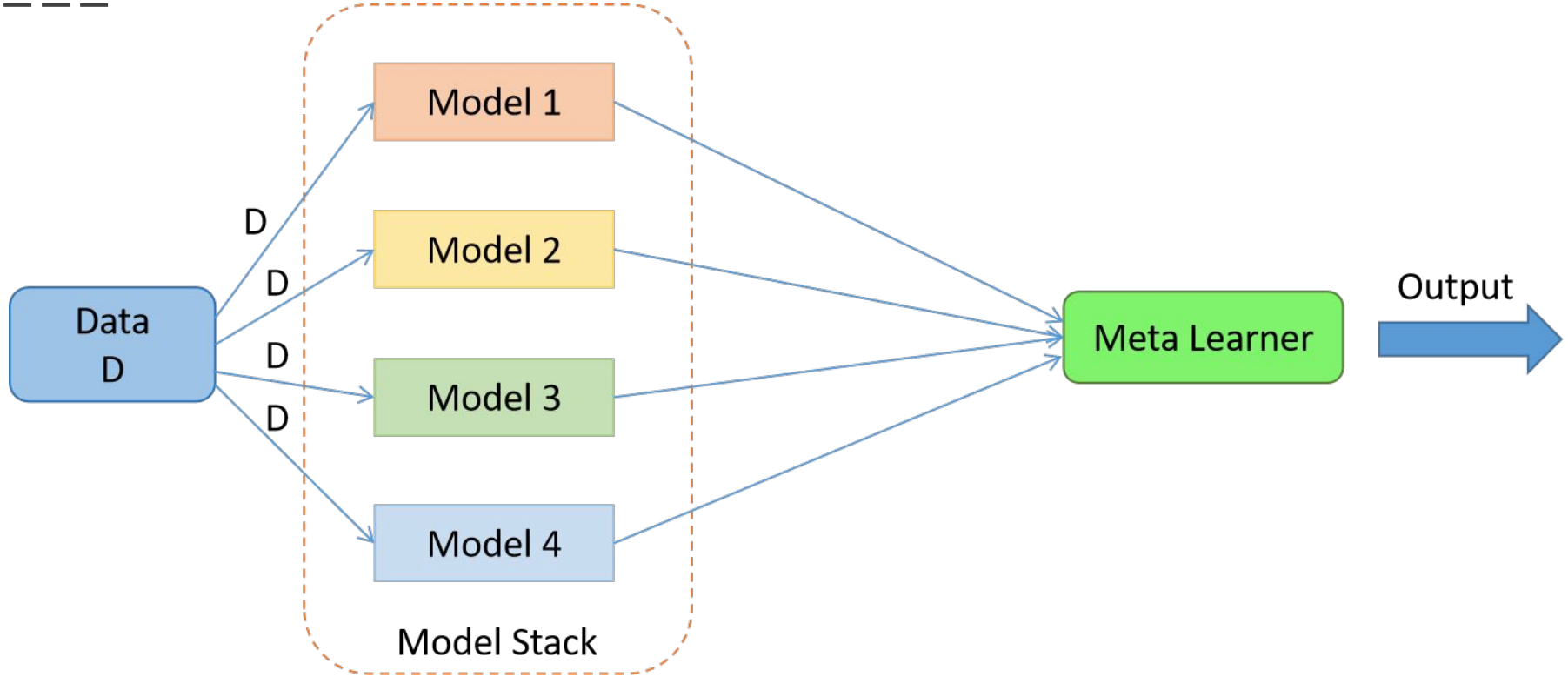
Each classifier outputs either a class label or a probability distribution over the classes.

Second Level Model (Meta-Learner h)

The outputs of the first-level classifiers become inputs to the second-level model.

This meta-classifier learns to combine the base model outputs to make the final prediction.

Example


Model 1
Model 2
Model 3
Model 4
D
D
D
D
Data
D
Model Stack
Meta Learner
Output

Example

Suppose we classify an image using three different models:

Classifier 1 → predicts "Class 1"

Classifier 2 → predicts "Class 2"

Classifier 3 → predicts "Class 1"

Instead of choosing one arbitrarily, we pass [Class 1, Class 2, Class 1] as input to a meta-classifier (h) that learns how to weigh these predictions.

Why Stacking?

- **It captures diverse classifier biases**
  - Different classifiers learn different decision boundaries.
  - Stacking helps by leveraging their strengths and reducing their weaknesses.
- **It can model complex functions better**
  - If the decision function is complex, a single classifier may not be enough.
  - Stacking allows simpler classifiers to be combined for more powerful predictions.
- **It can improve generalization**
  - Instead of overfitting on a single model, stacking learns a robust combination.

Choosing the Meta-Learner

The function h that combines classifiers′ outputs can be:

- Linear (weighted sum of classifier outputs)
- Non-linear (neural networks, decision trees, etc.)
- Bayesian (assigning probabilities based on likelihood)

Extending Stacking

Dynamic Weights: Instead of fixed weights, the meta-classifier can learn to trust certain classifiers more in specific regions of input space.

Multi-level Stacking: Instead of a single meta-layer, multiple layers of classifiers can be stacked.

Real-World Uses

Kaggle competitions often use stacking to improve model performance.

Financial forecasting, medical diagnosis, and image classification often benefit from stacked models.

Boosting

Boosting was initially studied in theoretical computer science, rather than empirical machine learning.

The fundamental idea is that if you have a weak classifier (one that is slightly better than random, say accuracy $0.5 + \varepsilon$), you can combine many such classifiers to form a strong classifier with accuracy arbitrarily close to 1.

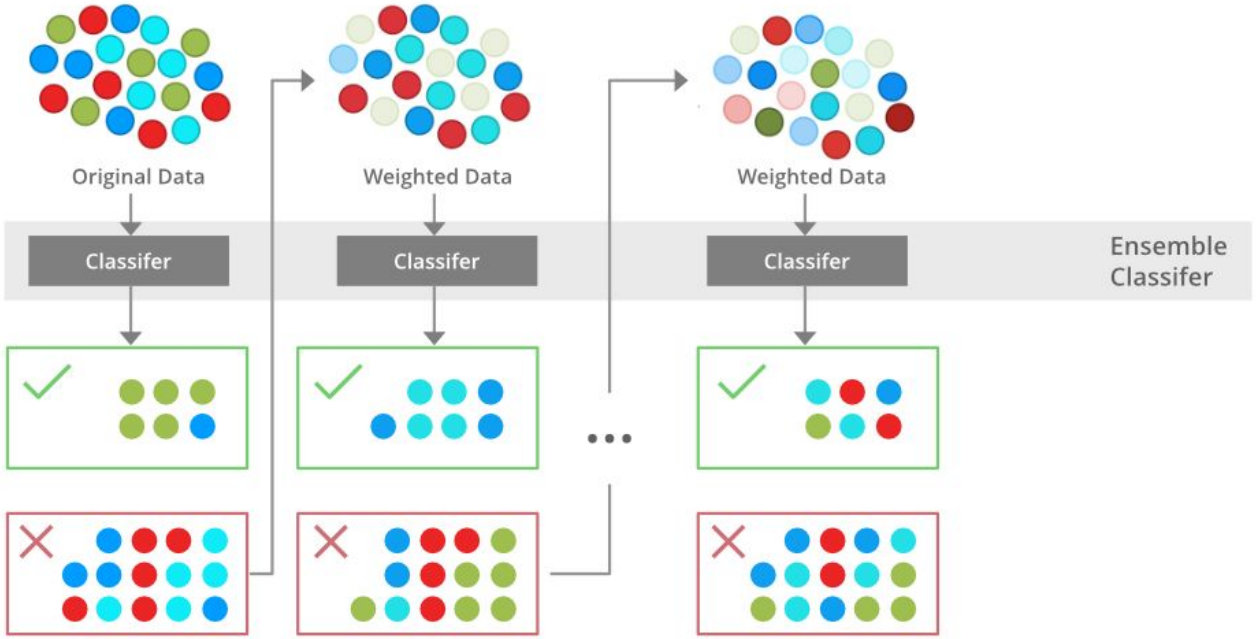
Boosting is a Sequential Process

Unlike traditional ensemble methods (like bagging, where classifiers are trained independently), boosting builds classifiers incrementally.

At each stage, a new classifier is added to correct errors made by the previous ones.

The goal is not just to add models but to ensure each new model reduces the error from previous models.

Boosting


Original Data
Weighted Data
Weighted Data
Classifer
Classifer
Classifer
Ensemble
Classifer

How AdaBoost Works

AdaBoost assigns weights to training samples.

Initially, all samples have equal weight.

After training each weak classifier, misclassified samples get higher weights, so the next classifier focuses more on them.

Final prediction is a weighted sum of all weak classifiers, where better classifiers get higher weights.

Loss Function in AdaBoost

The loss function used in AdaBoost is the exponential loss:

$\sum e^{-yif(xi)}$ where y_i is the true label and $f(x_i)$ is the model's prediction.

If a sample is correctly classified, its contribution to the loss decreases.

If it is misclassified, its contribution increases, causing future classifiers to focus more on it.

Training Weak Classifiers

Each weak classifier is trained on the weighted dataset.

The classifier minimizes the weighted error.

The final classifier is a weighted sum of all weak classifiers.

Key Insight

Boosting focuses on hard-to-classify samples.

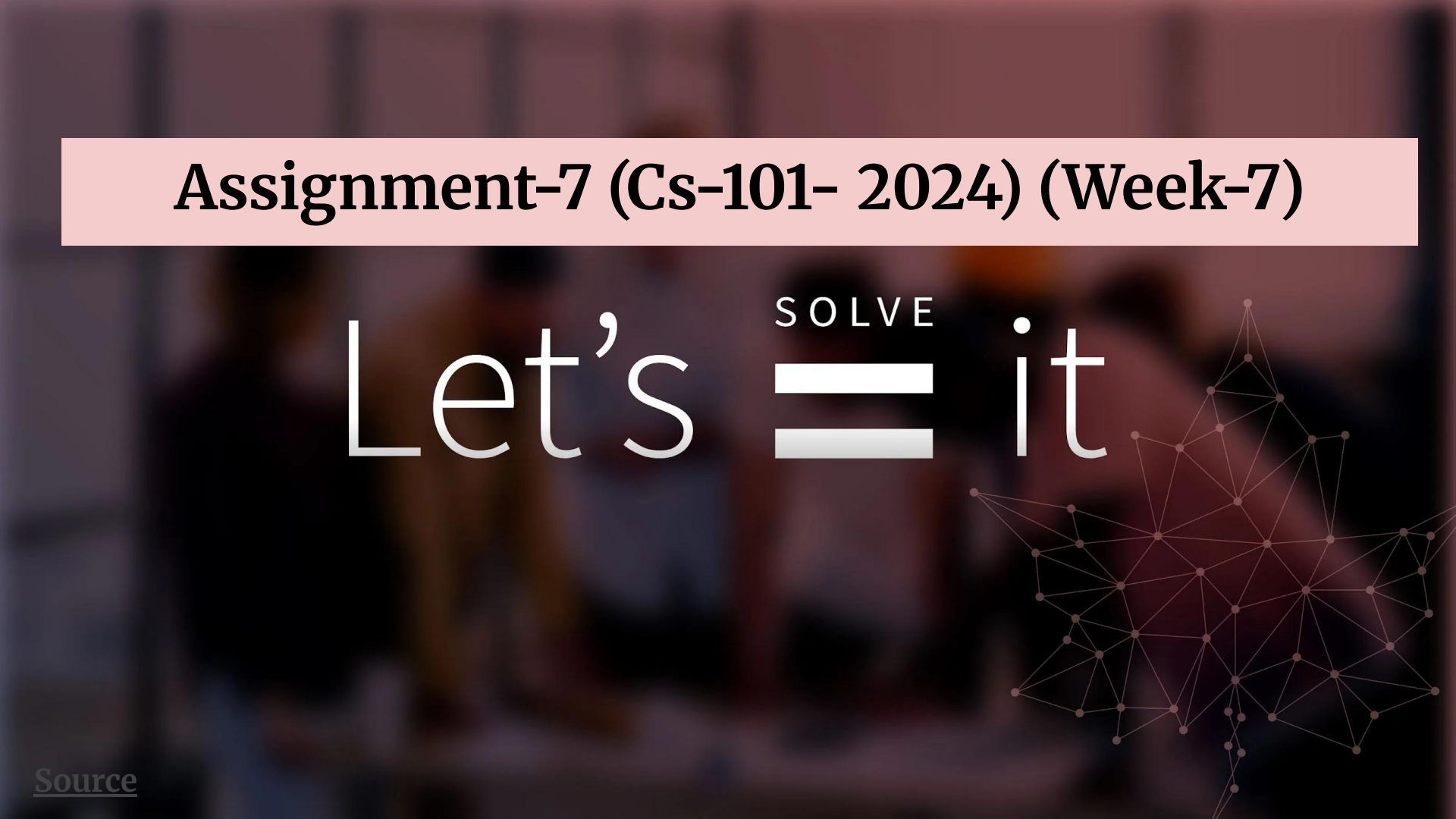
It does not just train multiple models independently—it adapts based on previous mistakes.

The final ensemble can achieve very high accuracy even though individual classifiers are weak.

Comparison with Other Ensemble Methods

Method	How It Works	Pros	Cons
Bagging	Train multiple models on random subsets (e.g., Random Forest)	Reduces variance	Needs diverse base learners
Boosting	Train models sequentially, fixing previous errors (e.g., AdaBoost, XGBoost)	Reduces bias, improves accuracy	Prone to overfitting
Stacking	Train multiple models and learn a meta-classifier to combine them	Leverages diverse models	Computationally expensive

Assignment-7 (Cs-101- 2024) (Week-7)



Source

Question-1

01:00

Define active learning

a) A learning approach where the algorithm passively receives all training data at once
b) A technique where the model learns from its own predictions without human intervention
c) An iterative learning process where the model selects the most informative data points for labeling
d) A method where the model randomly selects data points for training to reduce bias

Question-1- Correct answer

Define active learning

a) A learning approach where the algorithm passively receives all training data at once
b) A technique where the model learns from its own predictions without human intervention
c) An iterative learning process where the model selects the most informative data points for labeling
d) A method where the model randomly selects data points for training to reduce bias

Correct options: (c)

Question-2

01:00

Given 100 distinct data points, if you sample 100 times with replacement, what is the expected number of distinct points you will obtain?

a) Approximately 50

b) Approximately 63

c) Exactly 100

d) Approximately 37

Question-2- Correct answer

Given 100 distinct data points, if you sample 100 times with replacement, what is the expected number of distinct points you will obtain?

a) Approximately 50

b) Approximately 63 – 0.632 Bootstrap Method

c) Exactly 100

d) Approximately 37

Correct options: (d)

Question-3

01:00

What is the key difference between bootstrapping and cross-validation?

a) Bootstrapping uses the entire dataset for training, while cross-validation splits the data into subsets
b) Cross-validation allows replacement, while bootstrapping does not
c) Bootstrapping creates multiple samples with replacement, while cross-validation creates subsets without replacement
d) Cross-validation is used for model selection, while bootstrapping is only used for uncertainty estimation

Question-3 - Correct answer

What is the key difference between bootstrapping and cross-validation?

a) Bootstrapping uses the entire dataset for training, while cross-validation splits the data into subsets
b) Cross-validation allows replacement, while bootstrapping does not
c) Bootstrapping creates multiple samples with replacement, while cross-validation creates subsets without replacement
d) Cross-validation is used for model selection, while bootstrapping is only used for uncertainty estimation

Correct options: (c)

Question-4

03:00

Consider the following confusion matrix for a binary classification problem. What are the precision, recall, and accuracy of this classifier?

	Predicted Positive	Predicted Negative
Actual Positive	85	15
Actual Negative	20	80

a) Precision: 0.81, Recall: 0.85, Accuracy: 0.83

b) Precision: 0.85, Recall: 0.81, Accuracy: 0.85

c) Precision: 0.80, Recall: 0.85, Accuracy: 0.82

d) Precision: 0.85, Recall: 0.85, Accuracy: 0.80

Question-4 - Correct answer

Consider the following confusion matrix for a binary classification problem. What are the precision, recall, and accuracy of this classifier?

	Predicted Positive	Predicted Negative
Actual Positive	85	15
Actual Negative	20	80

a) **Precision: 0.81, Recall: 0.85, Accuracy: 0.83**
b) **Precision: 0.85, Recall: 0.81, Accuracy: 0.85**
c) **Precision: 0.80, Recall: 0.85, Accuracy: 0.82**
d) **Precision: 0.85, Recall: 0.85, Accuracy: 0.80**

Correct options: (a)

Question-5

01:00

AUC for your newly trained model is 0.5. Is your model prediction completely random?

a) Yes

b) No

c) ROC curve is needed to derive this conclusion

d) Cannot be determined even with ROC

Question-5- Explanation

01:00

AUC for your newly trained model is 0.5. Is your model prediction completely random?

An AUC of 0.5 indicates one of the following:

Random Predictions

- The model is guessing without any meaningful pattern.
- The ROC curve will be a diagonal line from (0,0) to (1,1), meaning TPR ≈ FPR at all thresholds.

Predicting a Single Class

- If the model predicts only one class (e.g., always "negative"), then the ROC curve will be flat at 0 or 1, leading to an AUC of 0.5.
- This can happen in imbalanced datasets where the model learns to favor the majority class.

Systematically Incorrect Predictions (Invertible Model)

- If the model inverts the predictions (e.g., predicting positives as negatives and vice versa), the ROC curve will be flipped around the diagonal.
- In this case, simply reversing the model's predictions (taking 1 minus the predicted probability) could improve performance.

What to Check?

Confusion Matrix: See if the model predicts only one class.

ROC Curve Shape: If it follows the diagonal, it's rando
if it's flipped, inverting the predictions might help.

Class Distribution: If the dataset is highly imbalanced,
AUC may not reflect actual performance.

Question-5 - Correct answer

AUC for your newly trained model is 0.5. Is your model prediction completely random?

a) Yes

b) No

c) ROC curve is needed to derive this conclusion

d) Cannot be determined even with ROC

Correct options: (c)

Question-6

01:00

You are building a model to detect cancer. Which metric will you prefer for evaluating your model?

a) Accuracy
b) Sensitivity
c) Specificity
d) MSE

Question-6 - Correct answer

You are building a model to detect cancer. Which metric will you prefer for evaluating your model?

a) **Accuracy**
b) **Sensitivity —-In medical application, FP is the most important (which sensitivity captures)**
c) **Specificity**
d) **MSE**

Correct options: (b)

Question-7

03:00

You have 2 binary classifiers A and B. A has accuracy=0% and B has accuracy=50%. Which classifier is more useful?

a) Both are good

b) B

c) A

d) Cannot say

Question-7 - Correct answer

You have 2 binary classifiers A and B. A has accuracy=0% and B has accuracy=50%.

Which classifier is more useful?

a) Both are good
b) B
c) A
d) Cannot say

Correct options: (c)

Question-8

03:00

You have a special case where your data has 10 classes and is sorted according to target labels. You attempt 5-fold cross validation by selecting the folds sequentially. What can you say about your resulting model?

a) It will have 100% accuracy.

b) It will have 0% accuracy.

c) It will have close to perfect accuracy.

d) Accuracy will depend on the compute power available for training

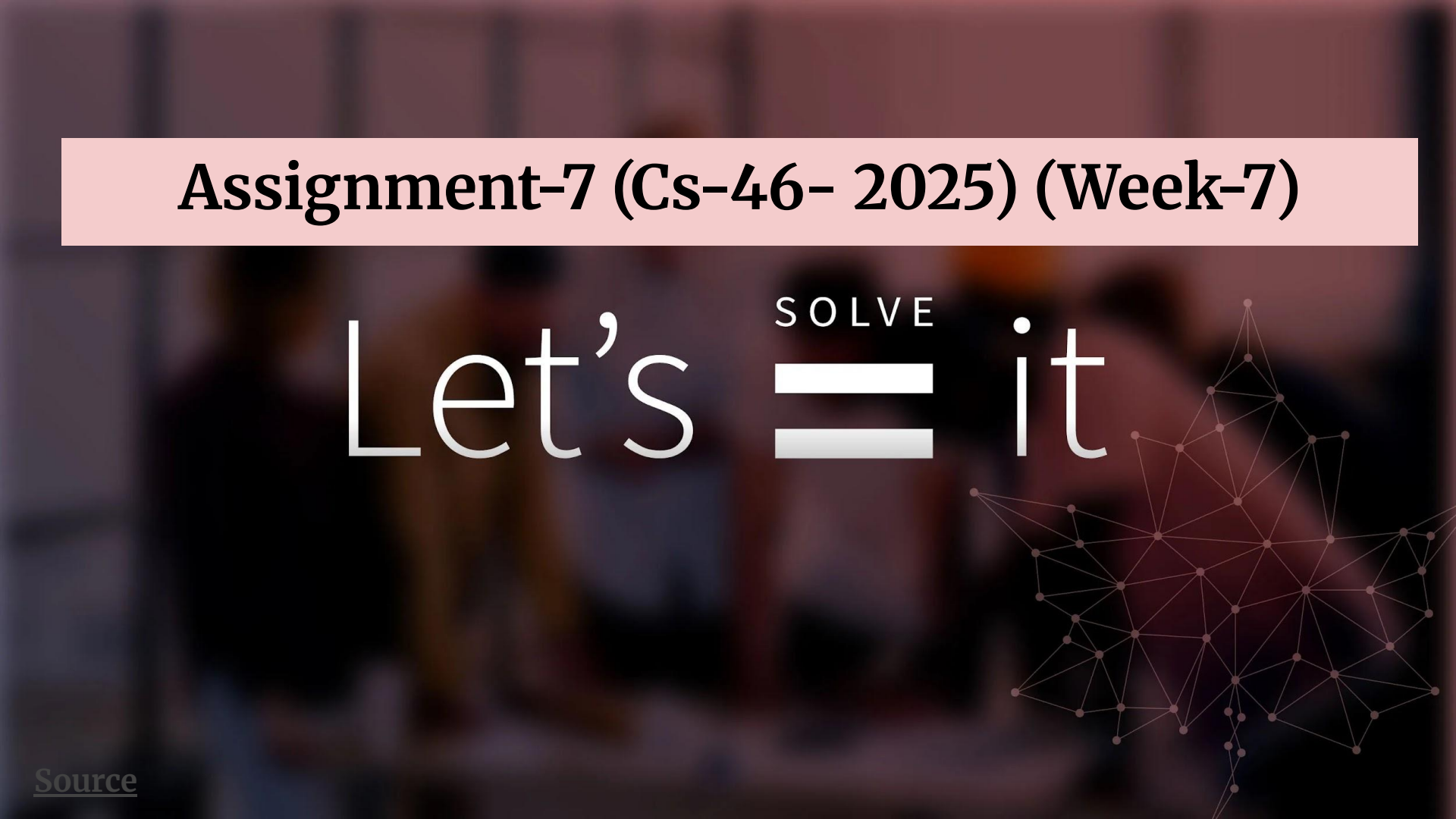
Question-8 - Correct answer

You have a special case where your data has 10 classes and is sorted according to target labels. You attempt 5-fold cross validation by selecting the folds sequentially. What can you say about your resulting model?

a) It will have 100% accuracy.
b) It will have 0% accuracy.
c) It will have close to perfect accuracy.
d) Accuracy will depend on the compute power available for training

Correct options: (b)

Assignment-7 (Cs-46- 2025) (Week-7)



Source

Question-1

01:00

Which of the following statement(s) regarding the evaluation of Machine Learning models is/are true?

a) A model with a lower training loss will perform better on a validation dataset.

b) A model with a higher training accuracy will perform better on a validation dataset.

c) The train and validation datasets can be drawn from different distributions

d) The train and validation datasets must accurately represent the real distribution of data

Question-1- Correct answer

Which of the following statement(s) regarding the evaluation of Machine Learning models is/are true?

a) A model with a lower training loss will perform better on a validation dataset.

b) A model with a higher training accuracy will perform better on a validation dataset.

c) The train and validation datasets can be drawn from different distributions

d) The train and validation datasets must accurately represent the real distribution of data

Correct options: (d)

Question-2

01:00

Suppose we have a classification dataset comprising of 2 classes A and B with 200 and 40 samples respectively. Suppose we use stratified sampling to split the data into train and test sets. Which of the following train-test splits would be appropriate?

a) Train-{A:50samples,B:10samples},Test-{A:150samples,B:30samples}

b) Train-{A:50samples,B:30samples}, Test- {A:150samples,B:10samples}

c) Train- {A:150samples,B:30samples}, Test- {A:50samples,B:10samples}

d) Train- {A:150samples,B:10samples}, Test- {A:50samples,B:30samples}

e) No, the answer is incorrect.

Question-2- Correct answer

Suppose we have a classification dataset comprising of 2 classes A and B with 200 and 40 samples respectively. Suppose we use stratified sampling to split the data into train and test sets. Which of the following train-test splits would be appropriate?

a) Train-{A:50samples,B:10samples},Test-{A:150samples,B:30samples}
b) Train-{A:50samples,B:30samples}, Test- {A:150samples,B:10samples}
c) Train- {A:150samples,B:30samples}, Test- {A:50samples,B:10samples}
d) Train- {A:150samples,B:10samples}, Test- {A:50samples,B:30samples}
e) No, the answer is incorrect.

Correct options: (c)

Question-3

01:00

Suppose we are performing cross-validation on a multiclass classification dataset with N data points. Which of the following statement(s) is/are correct?

a) In k -fold cross-validation, we train k-1 different models and evaluate them on the same test set
b) In k-fold cross-validation, we train k different models and evaluate them on different test sets
c) In k-fold cross-validation, each fold should have a class-wise proportion similar to the given dataset.
d) In LOOCV (Leave-One-Out Cross Validation), we train N different models, using N-1 data points for training each model

Question-3 - Correct answer

Suppose we are performing cross-validation on a multiclass classification dataset with N data points. Which of the following statement(s) is/are correct?

a) In k -fold cross-validation, we train k-1 different models and evaluate them on the same test set
b) In k-fold cross-validation, we train k different models and evaluate them on different test sets
c) In k-fold cross-validation, each fold should have a class-wise proportion similar to the given dataset.
d) In LOOCV (Leave-One-Out Cross Validation), we train N different models, using N-1 data points for training each model

Correct options: (b)(c)(d)

Question-4-7

(Qns 4 to 7) For a binary classification problem we train classifiers and evaluate them to obtain confusion matrices in the following format:

	Predicted Positive	Predicted Negative
Actual Positive		
Actual Negative		

Question-4

03:00

Which of the following classifiers should be chosen to maximize the recall?

a)

4	6
13	77

b)

8	2
40	60

c)

5	5
9	81

d)

7	3
0	90

Question-4 - Correct answer

Which of the following classifiers should be chosen to maximize the recall?

a)

4	6
13	77

b)

8	2
40	60

c)

5	5
9	81

d)

7	3
0	90

Correct options: (b)

Question-5

01:00

For the confusion matrices described in Q4, which of the following classifiers should be chosen to minimize the False Positive Rate?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Question-5 - Correct answer

For the confusion matrices described in Q4, which of the following classifiers should be chosen to minimize the False Positive Rate?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Correct options: (c)

Question-6

01:00

For the confusion matrices described in Q4, which of the following classifiers should be chosen to maximize the precision?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Question-6 - Correct answer

For the confusion matrices described in Q4, which of the following classifiers should be chosen to maximize the precision?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Correct options: (d)

Question-7

01:00

For the confusion matrices described in Q4, which of the following classifiers should be chosen to maximize the F1-score?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Question-7 - Correct answer

For the confusion matrices described in Q4, which of the following classifiers should be chosen to maximize the F1-score?

a)

4	6
6	84

b)

8	2
13	77

c)

1	9
2	88

d)

10	0
4	86

Correct options: (d)

Question-8

03:00

Which of the following statement(s) regarding boosting is/are correct?

a) Boosting is an example of an ensemble method
b) Boosting assigns equal weights to the predictions of all the weak classifiers
c) Boosting may assign unequal weights to the predictions of all the weak classifiers
d) The individual classifiers in boosting can be trained parallelly
e) The individual classifiers in boosting cannot be trained parallelly

Question-8 - Correct answer

Which of the following statement(s) regarding boosting is/are correct?

a) Boosting is an example of an ensemble method
b) Boosting assigns equal weights to the predictions of all the weak classifiers
c) Boosting may assign unequal weights to the predictions of all the weak classifiers
d) The individual classifiers in boosting can be trained parallelly
e) The individual classifiers in boosting cannot be trained parallelly

Correct options: (a)(c)(e)

Question-9

03:00

Which of the following statement(s) about bagging is/are correct?

a) Bagging is an example of an ensemble method
b) The individual classifiers in bagging can be trained in parallel
c) Training sets are constructed from the original dataset by sampling with replacement
d) Training sets are constructed from the original dataset by sampling without replacement
e) Bagging increases the variance of an unstable classifier.

Question-9 - Correct answer

Which of the following statement(s) about bagging is/are correct?

a) Bagging is an example of an ensemble method
b) The individual classifiers in bagging can be trained in parallel
c) Training sets are constructed from the original dataset by sampling with replacement
d) Training sets are constructed from the original dataset by sampling without replacement
e) Bagging increases the variance of an unstable classifier.

Correct options: (a)(b)(c)

Question-10

03:00

Which of the following statement(s) about ensemble methods is/are correct?

a) Ensemble aggregation methods like bagging aim to reduce overfitting and variance
b) Committee machines may consist of different types of classifiers
c) Weak learners are models that perform slightly worse than random guessing
d) Stacking involves training multiple models and stacking their predictions into new training data

Question-10 - Correct answer

Which of the following statement(s) about ensemble methods is/are correct?

a) Ensemble aggregation methods like bagging aim to reduce overfitting and variance
b) Committee machines may consist of different types of classifiers
c) Weak learners are models that perform slightly worse than random guessing
d) Stacking involves training multiple models and stacking their predictions into new training data

Correct options: (a)(b)(d)


THANK YOU

Suggestions and Feedback

Next Session:

Tuesday:
14-Sept-2025
3:00 - 5:00 PM