Introduction to Machine Learning

- Prof. Balaraman Ravindran | IIT Madras

Problem Solving Session (Week-6)

Shreya Bansal

PMRF PhD Scholar
IIT Ropar

Week-5 Contents

---

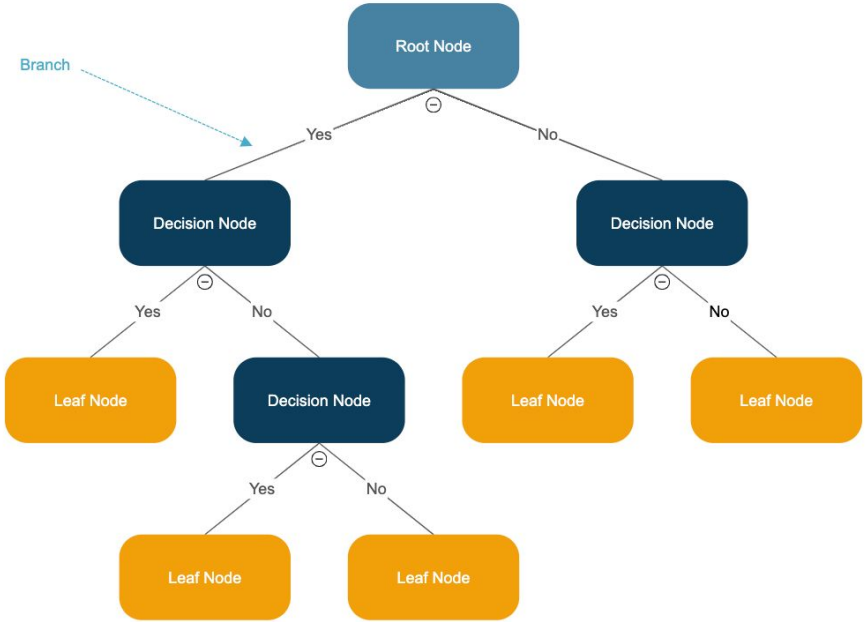
1. Decision Tree
2. Regression Trees
3. Pruning
4. Measures to split

Decision Trees in Machine Learning

- What is a Decision Tree?
- Used for classification & regression tasks
- Hierarchical structure with nodes representing decisions

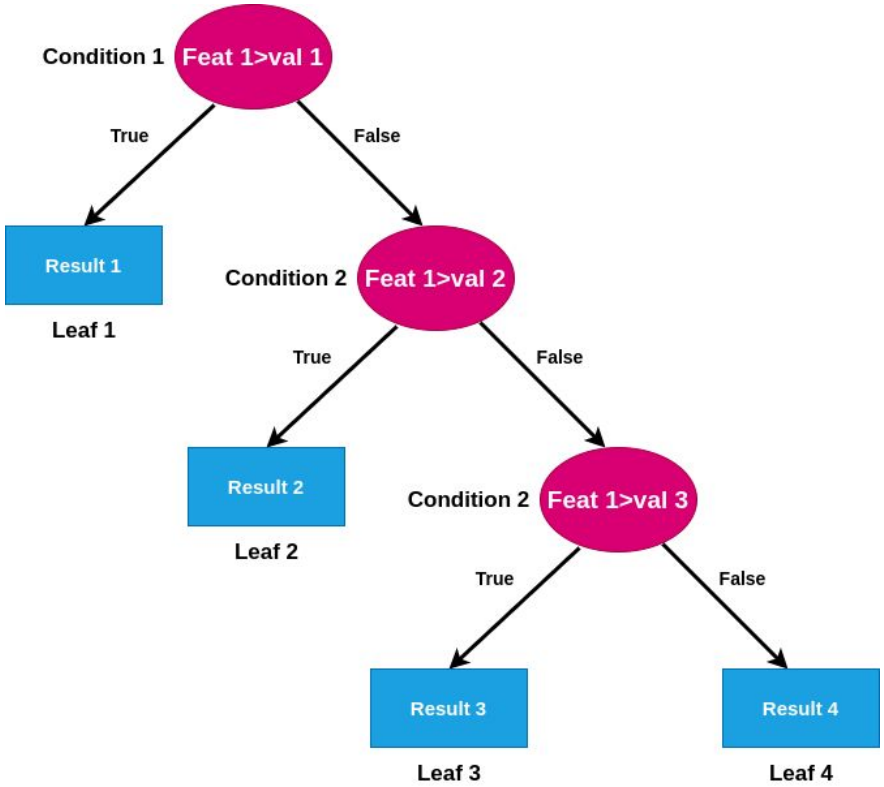
Components of a Decision Tree

- Root Node: Represents the entire dataset
- Decision Nodes: Intermediate splits based on features
- Leaf Nodes: Final output/classification



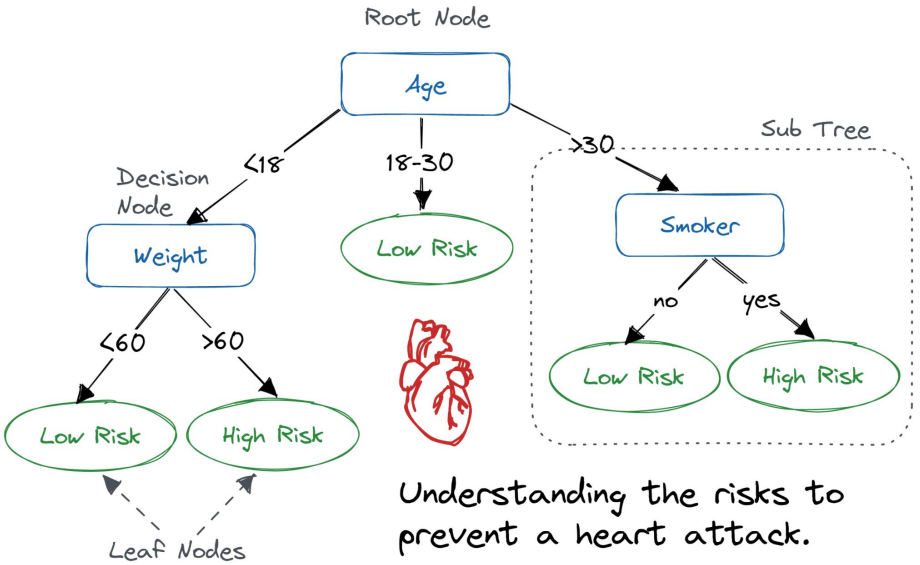
How Decision Trees Work

- Recursive partitioning of data
- Splitting criteria: Gini Index, Entropy, or Variance Reduction
- Stops when purity is achieved or a stopping condition is met



Splitting Criteria

- Gini Index (used in CART)
- Entropy (used in ID3 and C4.5)
- Mean Squared Error (MSE) (for regression trees)



Advantages of Decision Trees

- Easy to interpret
- Handles both numerical & categorical data
- Requires minimal data preparation

Limitations of Decision Trees

- Overfitting on small datasets
- Sensitive to noisy data
- Biased toward dominant classes

Pruning Techniques

- Pre-Pruning: Stops tree growth early
- Post-Pruning: Removes weak branches after full growth

Decision Tree Algorithms

- ID3
- C4.5
- CART
- Random Forest (ensemble method)

Decision Trees vs. Other ML Models

Feature	Decision Trees	SVM	Neural Networks
Interpretability	High	Medium	Low
Computation Time	Fast	Slow	Moderate
Handles Missing Data	Yes	No	No

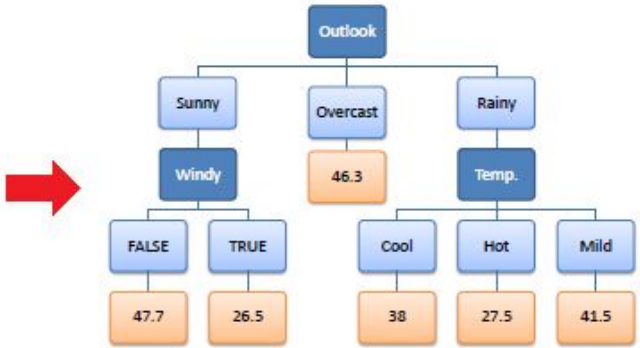
Decision Tree Regression

- What is regression?
- How does decision tree regression differ from linear regression?
- Key idea: Partitioning input space into regions

How Decision Tree Regression Works

- Splitting data into smaller regions
- Assigning a prediction to each region (constant or linear model)
- Example of a simple decision tree split

Predictors				Target
Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Piecewise Constant Approximation

- Each region has a constant predicted value (mean of data points in the region)
- Example illustration with a dataset

Piecewise Linear Approximation

- Instead of a constant, each region can have a linear function
- More complex but can capture trends within regions
- Example illustration

Parametric vs. Nonparametric Models

- Parametric models (e.g., Linear Regression) assume a fixed function form
- Decision trees are nonparametric: they adapt to the data

Finding Optimal Partitions

- The challenge of finding the best split points
- Exhaustive search is impractical for large datasets
- Need for efficient greedy algorithms

Greedy Algorithms for Tree Splitting

- Step 1: Select the best feature and split point (minimizing error)
- Step 2: Recursively apply the same process
- Splitting criteria (Mean Squared Error, Mean Absolute Error)

Greedy Algorithms for Tree Splitting

- Step 1: Find the Best Split
- Possible split points: (between 900, 1200, 1350, etc.)
- Suppose we try splitting at 1350 sqft
- Left Group (size ≤ 1350): [120, 150, 180, 200] → Mean = 162.5
- Right Group (size > 1350): [250, 270, 300, 350] → Mean = 292.5
- Compute MSE for both sides and select the split that minimizes the total error.

House Size (sqft)	Price ($1000s)
850	120
900	150
1200	180
1350	200
1600	250
1800	270
2200	300
2500	350

Greedy Algorithms for Tree Splitting

Advantages & Disadvantages

- **Pros: Handles nonlinear relationships, interpretable, works with small data**
- **Cons: Prone to overfitting, sensitive to small changes in data**

Early Stopping & Pruning in Decision Tree

- Decision trees split data recursively based on feature values.
- Used for both classification and regression tasks.
- Overfitting is a common issue due to deep trees.

What is Overfitting in Decision Trees?

- **Trees can become too complex, fitting noise instead of patterns.**
- **Leads to high variance and poor generalization to new data.**
- **Solution: Control tree depth using early stopping or pruning.**

Early Stopping in Decision Trees

- Stops tree growth before it fully fits the training data.
- Common stopping criteria:
  - Minimum samples per leaf
  - Maximum depth
  - Minimum information gain

Pruning in Decision Trees

- Post-pruning: Grow a full tree, then remove unnecessary nodes.
- Pre-pruning (early stopping): Stop growing the tree earlier using heuristics.
- Reduces model complexity and prevents overfitting.

Post-Pruning Techniques

- **Cost Complexity Pruning (CCP): Uses a penalty term to balance complexity.**
- **Reduced Error Pruning: Removes branches with little contribution to accuracy.**
- **Usually applied after full tree construction.**

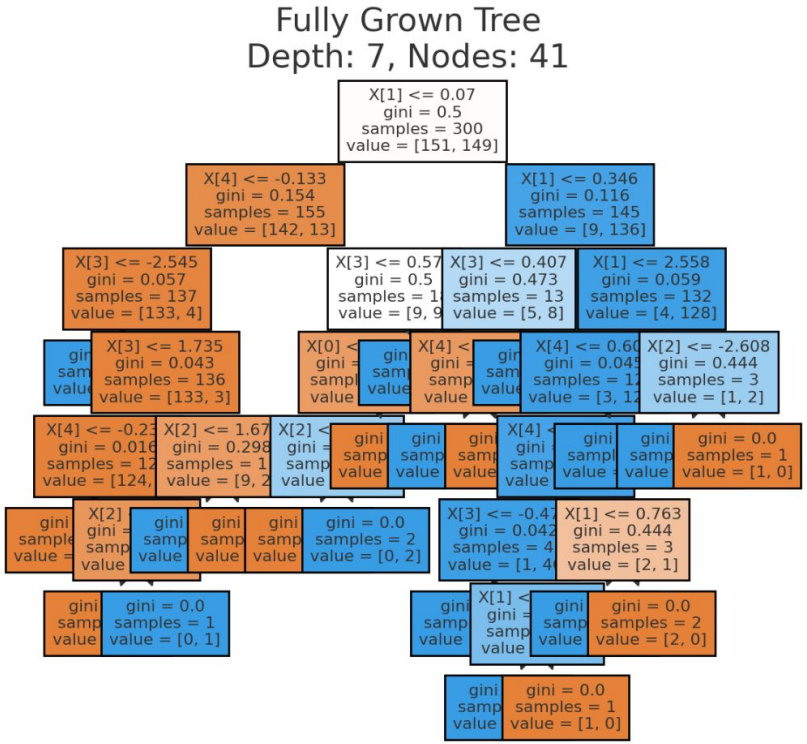
Pre-Pruning (Early Stopping) Techniques

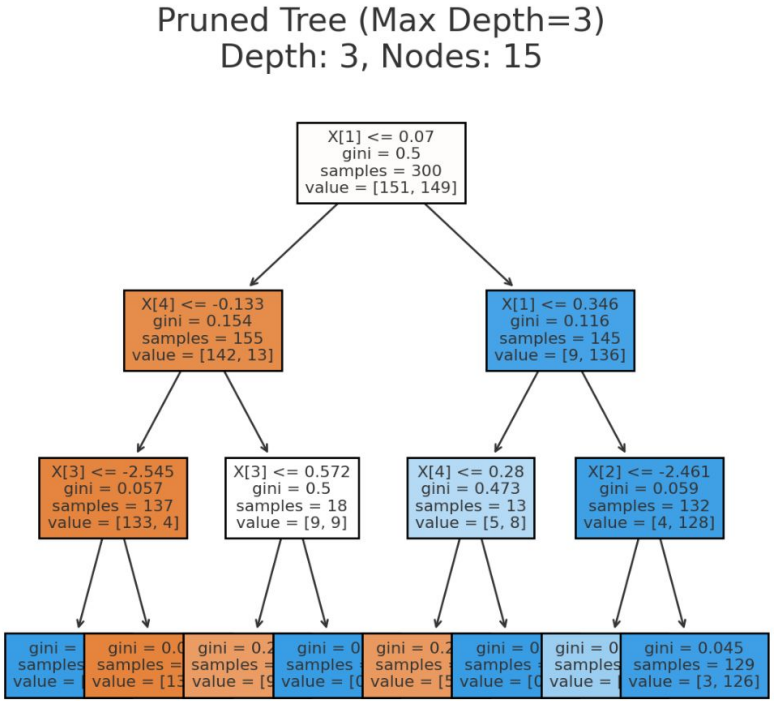
- **Max Depth Constraint: Limits the number of levels in the tree.**
- **Min Samples Split: Minimum number of samples required to split a node.**
- **Min Samples Leaf: Ensures each leaf has enough data points.**

Comparing Early Stopping vs. Pruning

Feature	Early Stopping	Pruning
Applied When?	During training	After full tree is grown
Controls Overfitting?	Yes	Yes
Commonly Used In?	Decision trees, Random Forests	Decision trees, CART
Computational Cost	Lower	Higher (requires re-evaluation)

Example – Decision Tree with Early Stopping





Impact of Pruning on Model Performance

- Reduced model complexity.
- Lower risk of overfitting.
- Improved generalization to unseen data.

Estimating Class Probability in a Region (P_{MK})

- The probability $P_{M,K}$ of a data point in region M belonging to class K is estimated as:
- $$P_{M,K} = \frac{\text{\# of data points of class } K \text{ in } M}{\text{Total \# of data points in M}}$$

Using Misclassification Error

- Assign the most frequent class $K(M)$ to each region M.
- Misclassification error for region M:
  - $1 - P_{M,K(M)}$

- When splitting, choose the split that minimizes the sum of misclassification errors across regions.

Classification Error Rate

Problem: A model classifies 100 samples, and the confusion matrix is:

Actual \ Predicted	0	1
0	40	10
1	15	35

Compute the classification error rate.

Solution: Error Rate=Misclassified samples/Total samples

Misclassified samples = 10+15=25

Total samples = 40+10+15+35=100

Error Rate= 25/100=0.25

Other Splitting Criteria

- Gini Index: Measures impurity. If all samples in a region belong to one class, Gini Index is 0.

$$Gini = \sum_k P_{M,k}(1 - P_{M,k})$$

- Cross-Entropy (or Information Gain):
- $H = -\sum_k P_{M,k} \log P_{M,k}$
- This relates to Shannon's entropy and represents the information needed to encode class labels.
- Information gain is the reduction in entropy when splitting a region.

Entropy Calculation

- Problem:A dataset contains 3 classes:
- Class A: 40 samples Class B: 30 samples Class C: 30 samples
- Compute the entropy.

- Solution:$H=-(p_A \log_2 p_A + p_B \log_2 p_B + p_C \log_2 p_C)$
- $p_A = 40/100=0.4$, $p_B=30/100=0.3$, $p_C=0.3$
- $H=-(0.4\log_2 0.4+0.3\log_2 0.3+0.3\log_2 0.3)$
-
- Using $\log_2 0.4 \approx -1.322$, $\log_2 0.3 \approx -1.737$
-
- $H=-[0.4(-1.322)+0.3(-1.737)+0.3(-1.737)]$
- $=-(-0.5288-0.5211-0.5211)=1.57$

Decision Tree Construction

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Compute entropy of Y:

$H(Y) = -p_1 \log_2 p_1 - p_0 \log_2 p_0$

$p1 = ¾, \quad p0 = 1/4$

$H(Y) = -(¾ \log_2 ¾ + ¼ \log_2 ¼)$

$= -(0.75 \times (-0.415) + 0.25 \times (-2)) = 0.811$

Compute Information Gain for X1 and X2, choose the best split.

After calculating, X1 gives the highest information gain, so we split on X1 first.

Continue splitting until leaf nodes are pure.

Information Gain for X1 and X2

Step 2: Compute Entropy for Splitting on X_1

Subsets when splitting on X_1:

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Group 1: $X_1 = 0$ (2 samples: Y = {0,1})

$$H(Y|X_1 = 0) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)$$

$$= -\left(0.5 \times -1 + 0.5 \times -1\right) = 1.0$$

- Group 2: $X_1 = 1$ (2 samples: Y = {1,1})

$$H(Y|X_1 = 1) = -\left(1\log_2 1 + 0\log_2 0\right) = 0$$

Since all samples belong to the same class, entropy = 0.

Weighted Entropy After Split on X_1:

$$H(Y|X_1) = \frac{2}{4}(1) + \frac{2}{4}(0) = 0.5$$

Information Gain for X_1:

$$IG(X_1) = H(Y) - H(Y|X_1)$$

$$= 0.811 - 0.5 = 0.311$$

Step 3: Compute Entropy for Splitting on X_2

Subsets when splitting on X_2:

X_1	X_2	Y
0	0	0
1	0	1
0	1	1
1	1	1

- Group 1: $X_2 = 0$ (2 samples: Y = {0,1})

$$H(Y|X_2 = 0) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

- Group 2: $X_2 = 1$ (2 samples: Y = {1,1})

$$H(Y|X_2 = 1) = -\left(1\log_2 1 + 0\log_2 0\right) = 0$$

Weighted Entropy After Split on X_2:

$$H(Y|X_2) = \frac{2}{4}(1) + \frac{2}{4}(0) = 0.5$$

Information Gain for X_2:

$$IG(X_2) = H(Y) - H(Y|X_2)$$

$$= 0.811 - 0.5 = 0.311$$

Choose Best Split

- Since $IG(X1)=0.311$ and $IG(X2)=0.311$, both attributes provide the same information gain. You can choose either $X1$ or $X2$ for the first split.
- Final Decision Tree Structure:
- Choose $X1$ as root (or $X2$, since they are equal).
- If $X1=0$, split further based on $X2$.
- If $X1=1$, the outcome is already pure ($Y=1$).

Gini Index Calculation

- **Problem:**
- **A dataset contains 80 instances. 50 belong to Class A, and 30 belong to Class B. Compute the Gini Index.**
- **Solution:**

$$Gini = 1 - (p_A^2 + p_B^2)$$

- $p_A = 50/80 = 0.625$, $p_B = 30/80 = 0.375$

- $Gini = 1 - (0.625^2 + 0.375^2) = 1 - (0.3906 + 0.1406) = 1 - 0.5312 = 0.4688$

Example

Problem

You're given a toy dataset with two candidate features—**Color** and **Size**—and a binary label **Buy?** (Yes=1, No=0). Choose the better first split using the **Gini index**.

ID	Color	Size	Buy?
1	Red	Small	1
2	Red	Small	1
3	Red	Large	1
4	Red	Large	0
5	Blue	Small	0
6	Blue	Large	0
7	Blue	Large	1
8	Green	Small	1
9	Green	Small	1
10	Green	Large	0

Totals: 6 positives (1), 4 negatives (0), 10 samples.

Example

Step 1: Parent node Gini

$$p_1 = \frac{6}{10} = 0.6, \quad p_0 = \frac{4}{10} = 0.4$$

$$Gini_{parent} = 1 - (p_1^2 + p_0^2) = 1 - (0.6^2 + 0.4^2) = 1 - (0.36 + 0.16) = 0.48$$

Step 2: Split by Color (multiway: Red/Blue/Green)

Red (4 samples): 3 pos, 1 neg

$$Gini_{Red} = 1 - \left(\left(\tfrac{3}{4}\right)^2 + \left(\tfrac{1}{4}\right)^2\right) = 1 - (0.5625 + 0.0625) = 0.375$$

Blue (3 samples): 1 pos, 2 neg

$$Gini_{Blue} = 1 - \left(\left(\tfrac{1}{3}\right)^2 + \left(\tfrac{2}{3}\right)^2\right) = 1 - \left(\tfrac{1}{9} + \tfrac{4}{9}\right) = \tfrac{4}{9} \approx 0.4444$$

Green (3 samples): 2 pos, 1 neg

$$Gini_{Green} = 1 - \left(\left(\tfrac{2}{3}\right)^2 + \left(\tfrac{1}{3}\right)^2\right) = 1 - \left(\tfrac{4}{9} + \tfrac{1}{9}\right) = \tfrac{4}{9} \approx 0.4444$$

Weighted Gini after split (Color):

$$Gini_{Color} = \frac{4}{10} \cdot 0.375 + \frac{3}{10} \cdot 0.4444 + \frac{3}{10} \cdot 0.4444 = 0.15 + 0.1333 + 0.1333 \approx 0.4167$$

Gini gain (Color):

$$0.48 - 0.4167 = 0.0633$$

Step 3: Split by Size (binary: Small/Large)

Small (5 samples): 4 pos, 1 neg

$$Gini_{Small} = 1 - \left(0.8^2 + 0.2^2\right) = 1 - (0.64 + 0.04) = 0.32$$

Large (5 samples): 2 pos, 3 neg

$$Gini_{Large} = 1 - \left(0.4^2 + 0.6^2\right) = 1 - (0.16 + 0.36) = 0.48$$

Weighted Gini after split (Size):

$$Gini_{Size} = \frac{5}{10} \cdot 0.32 + \frac{5}{10} \cdot 0.48 = 0.16 + 0.24 = 0.40$$

Gini gain (Size):

$$0.48 - 0.40 = 0.08$$

Conclusion

- Weighted Gini after split by **Color** ≈ **0.4167** (gain ≈ 0.0633)
- Weighted Gini after split by **Size** = **0.40** (gain = 0.08)

Since **lower is better**, splitting on **Size** is the better first split (it yields the lower post-split Gini and higher Gini gain).

Example

1) Parent node entropy

$p_1 = 6/10 = 0.6,\ p_0 = 0.4.$

Entropy formula (binary):

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p).$$

Compute:

$$H_{parent} = -0.6\log_2(0.6) - 0.4\log_2(0.4) \approx 0.9709505945 \text{ bits}$$

---

2) Split by Color (Red / Blue / Green)

Counts per color:

- **Red:** 4 samples → 3 pos, 1 neg. $p_{1|Red} = 3/4 = 0.75.$
  $H_{Red} = -0.75\log_2(0.75) - 0.25\log_2(0.25) \approx 0.8112781245.$
- **Blue:** 3 samples → 1 pos, 2 neg. $p_{1|Blue} = 1/3 \approx 0.3333333.$
  $H_{Blue} = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) \approx 0.9182958341.$
- **Green:** 3 samples → 2 pos, 1 neg. $p_{1|Green} = 2/3 \approx 0.6666667.$
  $H_{Green} \approx 0.9182958341$ (same as Blue because symmetric).

Weighted child entropy for Color:

$$H_{Color} = \frac{4}{10}H_{Red} + \frac{3}{10}H_{Blue} + \frac{3}{10}H_{Green}$$

$$H_{Color} \approx 0.4 \cdot 0.8112781245 + 0.3 \cdot 0.9182958341 + 0.3 \cdot 0.9182958341 \approx 0.8754887502$$

Information gain (Color):

$$IG_{Color} = H_{parent} - H_{Color} \approx 0.9709505945 - 0.8754887502 = 0.0954618442$$

3) Split by Size (Small / Large)

Counts per size:

- **Small:** 5 samples → 4 pos, 1 neg. $p_{1|Small} = 4/5 = 0.8.$
  $H_{Small} = -0.8\log_2(0.8) - 0.2\log_2(0.2) \approx 0.7219280949.$
- **Large:** 5 samples → 2 pos, 3 neg. $p_{1|Large} = 2/5 = 0.4.$
  $H_{Large} = -0.4\log_2(0.4) - 0.6\log_2(0.6) \approx 0.9709505945.$

Weighted child entropy for Size:

$$H_{Size} = \frac{5}{10}H_{Small} + \frac{5}{10}H_{Large} = 0.5 \cdot 0.7219280949 + 0.5 \cdot 0.9709505945 \approx 0.8464393447$$

Information gain (Size):

$$IG_{Size} = H_{parent} - H_{Size} \approx 0.9709505945 - 0.8464393447 = 0.1245112498$$

Conclusion

- $IG_{Color} \approx \mathbf{0.09546}$ bits
- $IG_{Size} \approx \mathbf{0.12451}$ bits

Splitting on *Size* yields the larger information gain, so Size is the better first split.

Decision Tree Splitting Strategy

- For each feature j, find the best split point s that optimizes Gini index, cross-entropy, or misclassification error.
- When combining multiple regions, use weighted combinations of their impurity scores.

Pruning and Final Evaluation

- Grow the tree using Gini index or entropy.
- Prune the tree using misclassification error, as the final goal is classification accuracy.

Splitting Attributes & Split Points

- Decision trees typically assume continuous attributes.
- For continuous attributes:
- Choose a numerical split point (e.g., threshold on age).
- Optimize split point based on impurity measures (Gini index, entropy, etc.).
- What happens if attributes are categorical?

Handling Categorical Attributes

- Categorical attributes take discrete values (e.g., color: Red, Blue, Green).
- Some categorical attributes may have an inherent ordering (e.g., Age categories: Young, Middle-aged, Old).
- If unordered, defining a split becomes challenging.

The Challenge with Categorical Attributes

- For q categorical values, possible subsets = 2^(q-1) - 1.
- Example: If q = 5 (Red, Blue, Green, Yellow, Magenta), then 31 possible subsets!
- Evaluating all possible splits is computationally expensive.

Strategies for Categorical Splitting

- **Binary Classification Trick**
  - **Compute probability of each category belonging to class 1.**
  - **Sort categories by this probability.**
  - **Treat as an ordered attribute and find the best split.**
- **Grouping Categorical Values**
  - **Try different subsets and optimize impurity reduction.**
  - **Use heuristics to reduce search space.**

Example of Binary Classification Trick

- Assume categorical values: Red, Blue, Green, Yellow, Magenta.
- Compute fraction of class 1 instances per category:
    - Red → 0.2
    - Blue → 0.3
    - Green → 0.4
    - Yellow → 0.45
    - Magenta → 0.55
- Sort and split at optimal point based on impurity measure.

Choosing the Best Split

- Possible splits:
    - {Red} | {Blue, Green, Yellow, Magenta}
    - {Red, Blue} | {Green, Yellow, Magenta}
    - {Red, Blue, Green} | {Yellow, Magenta}
    - {Red, Blue, Green, Yellow} | {Magenta}
- Use Gini index, entropy, or misclassification error to determine the best split.

Advantages of This Approach

- **Reduces complexity from 2^(q-1) - 1 to q-1 comparisons.**
- **Ensures optimal splits for binary classification tasks.**
- **Similar strategy can be adapted for multi-class classification with modifications.**

Multi-class categorical attributes in decision trees

Example: Predicting Shirt Popularity Based on Color and Price

We have a dataset of shirts with attributes **Color** and **Price**, and the target label is **Popularity** (Low, Medium, High).

Color	Price	Popularity
Red	15	High
Blue	10	Medium
Yellow	20	High
Magenta	25	Low
Green	18	Medium
Red	22	Low
Blue	17	Medium
Yellow	12	High

Multi-class categorical attributes in decision trees

---

1. Decision Tree with Multi-Way Splits

- If we split on **Color**, we create **5 branches** (Red, Blue, Yellow, Magenta, Green).
- This creates a problem:
  - Some branches have very few data points (e.g., Magenta only has 1 row).
  - Leads to **overfitting** and sparse data issues.

Alternative Approach: Binning (Grouping Similar Values)

Instead of treating each color separately, we **cluster colors** into categories like:

- **Warm Colors**: Red, Yellow
- **Cool Colors**: Blue, Green
- **Rare Colors**: Magenta

Now, we only need **3 branches**, making the decision tree more robust.

Multi-class categorical attributes in decision trees

2. One-Hot Encoding + Dimensionality Reduction

Instead of using colors directly, we convert them into **indicator variables** (One-Hot Encoding):

Red	Blue	Yellow	Magenta	Green	Price	Popularity
1	0	0	0	0	15	High
0	1	0	0	0	10	Medium
0	0	1	0	0	20	High
0	0	0	1	0	25	Low

Now, instead of making a decision on **5 separate colors**, we can use **Principal Component Analysis (PCA)** or another method to reduce these 5 dimensions into **1 continuous value** and use that for splitting.

Multi-class categorical attributes in decision trees

---

3. Binary Splitting Instead of Multi-Way Splitting

Instead of splitting into 5 groups at once, we **split iteratively**:

1. First, split into **(Red, Yellow) vs. (Blue, Green, Magenta)**
2. Then, further split these groups.

This prevents **overfitting** while still maintaining useful information.

Sources of Missing Data

No response in surveys

Noise removal

Malfunctioning sensors

Data not being recorded due to practical reasons

Handling Missing Values

Removing attributes when they are missing in too many samples

Removing rows if missing values are few but critical

Using imputation techniques:

- Mean imputation
- Class-conditioned mean imputation
- Regression-based imputation
- Multiple imputation (sampling from a probability distribution)

Why Imputation Matters

- Helps retain correlations in data
- Prevents loss of valuable information
- Reduces variance when multiple imputation is used

Alternative Approach: Treating Missingness as a Feature

Instead of filling in missing values, introduce a new category ("missing")

Helps when missing values have a pattern or meaning

Surrogate Splits in Decision Trees

Used when a primary split attribute has missing values

Finds another attribute that splits the data in a similar way

Ensures decision trees handle missing values effectively during both training and testing

Instability in Decision Trees

- Decision trees are highly unstable.
- Definition of instability: Small changes in training data can lead to large changes in the tree structure.
- Example: A feature that was at the root node may move deeper in the tree with minor dataset modifications.
- Impact: Higher variance when the dataset is small.

Regularization Techniques

Pruning: Reduces tree complexity by removing less important branches.

Feature Selection: Limits the number of features used.

Limiting Tree Depth: Prevents excessive growth and overfitting.

Drawback: These methods only help to some extent, and variance remains high.

Bagging for Stability

Definition: Bootstrap Aggregating (Bagging) reduces variance by training multiple models on different random subsets of the data.

Process:

Train multiple decision trees on different 70% random subsets of data.

Combine predictions (e.g., majority vote for classification, averaging for regression).

Advantage: Reduces instability and improves accuracy.

Disadvantage: Loss of interpretability due to multiple trees.

Example

Example: Play Tennis Decision Tree

We have the following dataset with attributes:

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rainy	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

The target variable is **Play Tennis** (Yes/No).

Example

Step 1: Compute Entropy of the Dataset

Entropy formula:

$$H(S) = -\sum p_i \log_2 p_i$$

We have **9 "Yes"** and **5 "No"** in the dataset:

$$H(S) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right)$$

$$H(S) = -(0.643 \times -0.466) - (0.357 \times -0.807)$$

$$H(S) = 0.940$$

Example

Step 2: Compute Information Gain for Each Attribute

We calculate entropy for each attribute's splits and compute **Information Gain**:

(a) Outlook Attribute

Outlook	Play Tennis (Yes)	Play Tennis (No)	Total
Sunny	2	3	5
Overcast	4	0	4
Rainy	3	2	5

$$H(Sunny) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.971$$

$$H(Overcast) = 0 \quad \text{(since all are Yes)}$$

$$H(Rainy) = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.971$$

Weighted entropy:

$$H_{\text{Outlook}} = \left(\frac{5}{14} \times 0.971\right) + \left(\frac{4}{14} \times 0\right) + \left(\frac{5}{14} \times 0.971\right)$$

$$H_{\text{Outlook}} = 0.693$$

$$IG_{\text{Outlook}} = 0.940 - 0.693 = 0.247$$

Example

Similarly, we compute **Information Gain** for **Temperature, Humidity, and Wind** and find the attribute with the highest **IG**.

After computing, we get:

Attribute	Information Gain
Outlook	**0.247**
Temperature	0.029
Humidity	0.151
Wind	0.048

Since **Outlook has the highest IG**, we choose it as the root node.

Step 3: Split on "Outlook" and Repeat the Process

- If **Outlook = Overcast**, Play Tennis = **Yes** (pure, stop here).
- If **Outlook = Sunny or Rainy**, we continue splitting.

For **Sunny**, we calculate IG for **Humidity, Wind, and Temperature** and split accordingly.

For **Rainy**, we compute IG for **Humidity, Wind, and Temperature** and split accordingly.

This process continues until all leaf nodes are **pure** (contain only Yes or No).

Example

Final Decision Tree

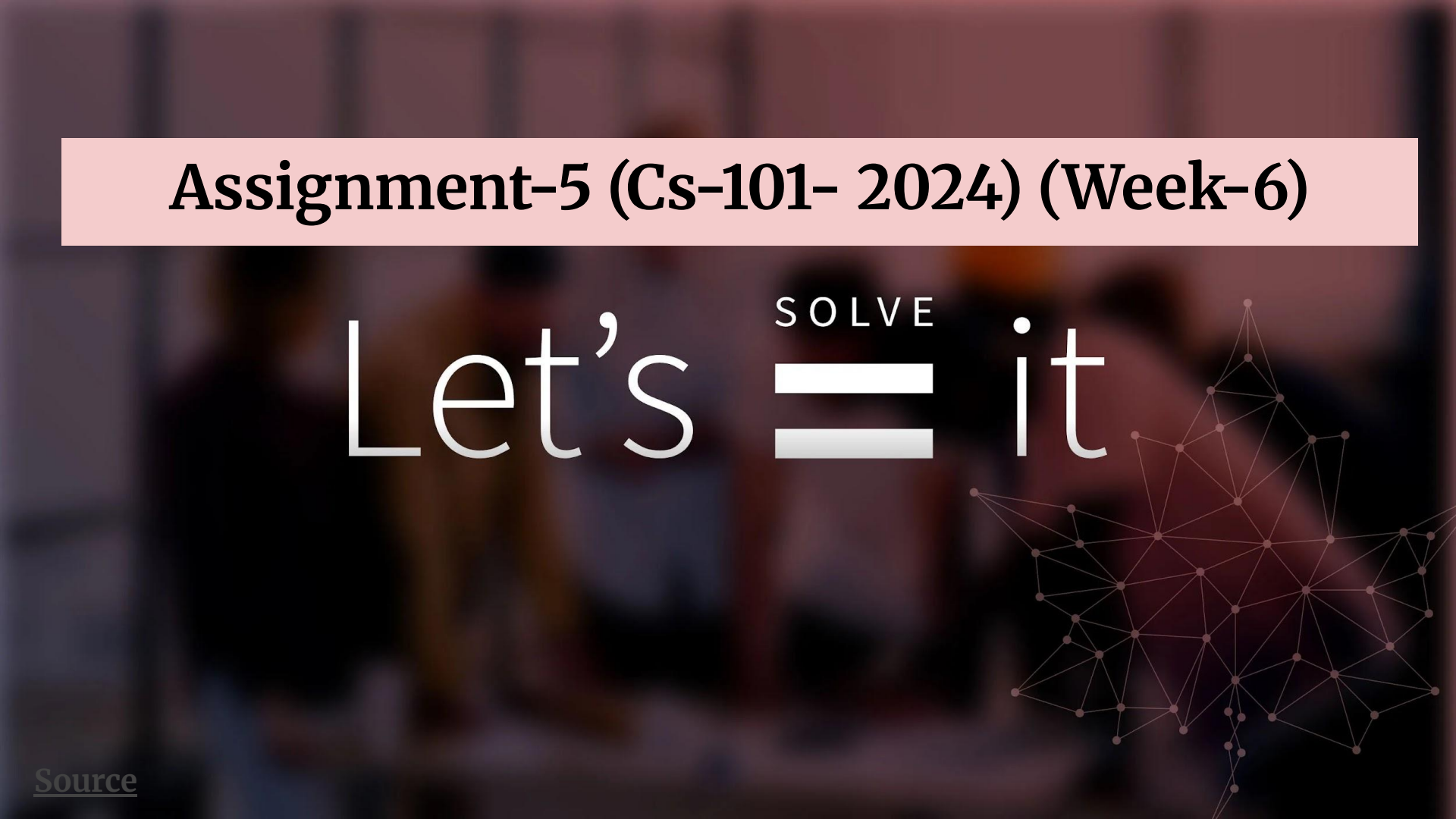
yaml

Copy Edit

```
          Outlook
         /   |   \
      Sunny Overcast Rainy
      /   \    |    /    \
  Humidity  Yes  Wind  Humidity
  /     \        /   \  /    \
High  Normal Weak Strong High Normal
 No      Yes   Yes    No   No   Yes
```

This decision tree can now be used to classify new examples!

Assignment-5 (Cs-101- 2024) (Week-6)



Source

Question-1

01:00

Entropy for a 90-10 split between two classes is:

a) 0.469
b) 0.195
c) 0.204
d) None

Question-1- Correct answer

Entropy for a 90-10 split between two classes is:

a) 0.469
b) 0.195
c) 0.204
d) None

Entropy (H) is calculated using the formula:

$$H = -\sum p_i \log_2 p_i$$

For a 90-10 split between two classes:

- $p_1 = 0.9$
- $p_2 = 0.1$

$$H = -(0.9 \log_2 0.9 + 0.1 \log_2 0.1)$$

Approximating the logarithm values:

$$\log_2 0.9 \approx -0.152$$

$$\log_2 0.1 \approx -3.322$$

$$H \approx -(0.9 \times -0.152 + 0.1 \times -3.322)$$

$$H \approx -(-0.1368 - 0.3322)$$

$$H \approx 0.469$$

So, the entropy for a 90-10 split is **approximately 0.469 bits.**

Correct options: (a)

Question-2

01:00

Consider a dataset with only one attribute(categorical). Suppose, there are 8 unordered values in this attribute, how many possible combinations are needed to find the best split-point for building the decision tree classifier?

a) 511

b) 1023

c) 512

d) 127

Question-2- Correct answer

Consider a dataset with only one attribute(categorical). Suppose, there are 8 unordered values in this attribute, how many possible combinations are needed to find the best split-point for building the decision tree classifier?

a) 511

b) 1023

c) 512

d) 127 —--- $2^{(8-1)}-1$

Correct options: (d)

Question-3

01:00

Having built a decision tree, we are using reduced error pruning to reduce the size of the tree. We select a node to collapse. For this particular node, on the left branch, there are three training data points with the following outputs: 5, 7, 9.6, and for the right branch, there are four training data points with the following outputs: 8.7, 9.8, 10.5, 11. The average value of the outputs of data points denotes the response of a branch. The original responses for data points along the two branches (left & right respectively) were response-left and, response-right and the new response aftercollapsing the node is response-new. What are the values for response-left, response-right and response-new (numbers in the option are given in the same order)?

a) 9.6, 11, 10.4

b) 7.2; 10; 8.8

c) 5, 10.5, 15

d) depends on the tree height.

Question-3 - Correct answer

Having built a decision tree, we are using reduced error pruning to reduce the size of the tree. We select a node to collapse. For this particular node, on the left branch, there are three training data points with the following outputs: 5, 7, 9.6, and for the right branch, there are four training data points with the following outputs: 8.7, 9.8, 10.5, 11. The average value of the outputs of data points denotes the response of a branch. The original responses for data points along the two branches (left & right respectively) were response-left and, response-right and the new response aftercollapsing the node is response-new. What are the values for response-left, response-right and response-new (numbers in the option are given in the same order)?

a) 9.6, 11, 10.4

b) 7.2; 10; 8.8

c) 5, 10.5, 15

d) depends on the tree height.

Correct options: (b)

Question-4

03:00

Which of the following is a good strategy for reducing the variance in a decision tree?

a) If improvement of taking any split is very small, don't make a split. (Early Stopping)
b) Stop splitting a leaf when the number of points is less than a set threshold K.
c) Stop splitting all leaves in the decision tree when any one leaf has less than a set threshold K points.
d) None of the Above.

Question-4 - Correct answer

Which of the following is a good strategy for reducing the variance in a decision tree?

a) If improvement of taking any split is very small, don't make a split. (Early Stopping)
b) Stop splitting a leaf when the number of points is less than a set threshold K.
c) Stop splitting all leaves in the decision tree when any one leaf has less than a set threshold K points.
d) None of the Above.

Correct options: (b)

Question-5

01:00

Which of the following statements about multiway splits in decision trees with categorical features is correct?

a) They always result in deeper trees compared to binary splits
b) They always provide better interpretability than binary splits
c) They can lead to overfitting when dealing with high-cardinality categorical features
d) They are computationally less expensive than binary splits for all categorical features

Question-5 - Correct answer

Which of the following statements about multiway splits in decision trees with categorical features is correct?

a) They always result in deeper trees compared to binary splits
b) They always provide better interpretability than binary splits
c) They can lead to overfitting when dealing with high-cardinality categorical features
d) They are computationally less expensive than binary splits for all categorical features

Correct options: (c)

Question-6

01:00

Which of the following statements about imputation in data preprocessing is most accurate?

a) Mean imputation is always the best method for handling missing numerical data
b) Imputation should always be performed after splitting the data into training and test sets
c) Missing data is best handled by simply removing all rows with any missing values
d) Multiple imputation typically produces less biased estimates than single imputation methods

Question-6 - Correct answer

Which of the following statements about imputation in data preprocessing is most accurate?

a) Mean imputation is always the best method for handling missing numerical data
b) Imputation should always be performed after splitting the data into training and test sets
c) Missing data is best handled by simply removing all rows with any missing values
d) Multiple imputation typically produces less biased estimates than single imputation methods

Correct options: (d)

Question-7

03:00

Consider the following dataset:
Which among the following split-points for feature2 would give the best split according to the misclassification error?

a) 186.5

b) 188.6

c) 189.2

d) 198.1

feature1	*feature2*	*output*
18.3	187.6	a
14.7	184.9	a
19.4	193.3	a
17.9	180.5	a
19.1	189.1	a
17.6	191.9	b
19.9	190.2	b
17.3	198.6	b
18.7	182.6	b
15.2	187.3	b

Question-7 - Correct answer

Consider the following dataset:

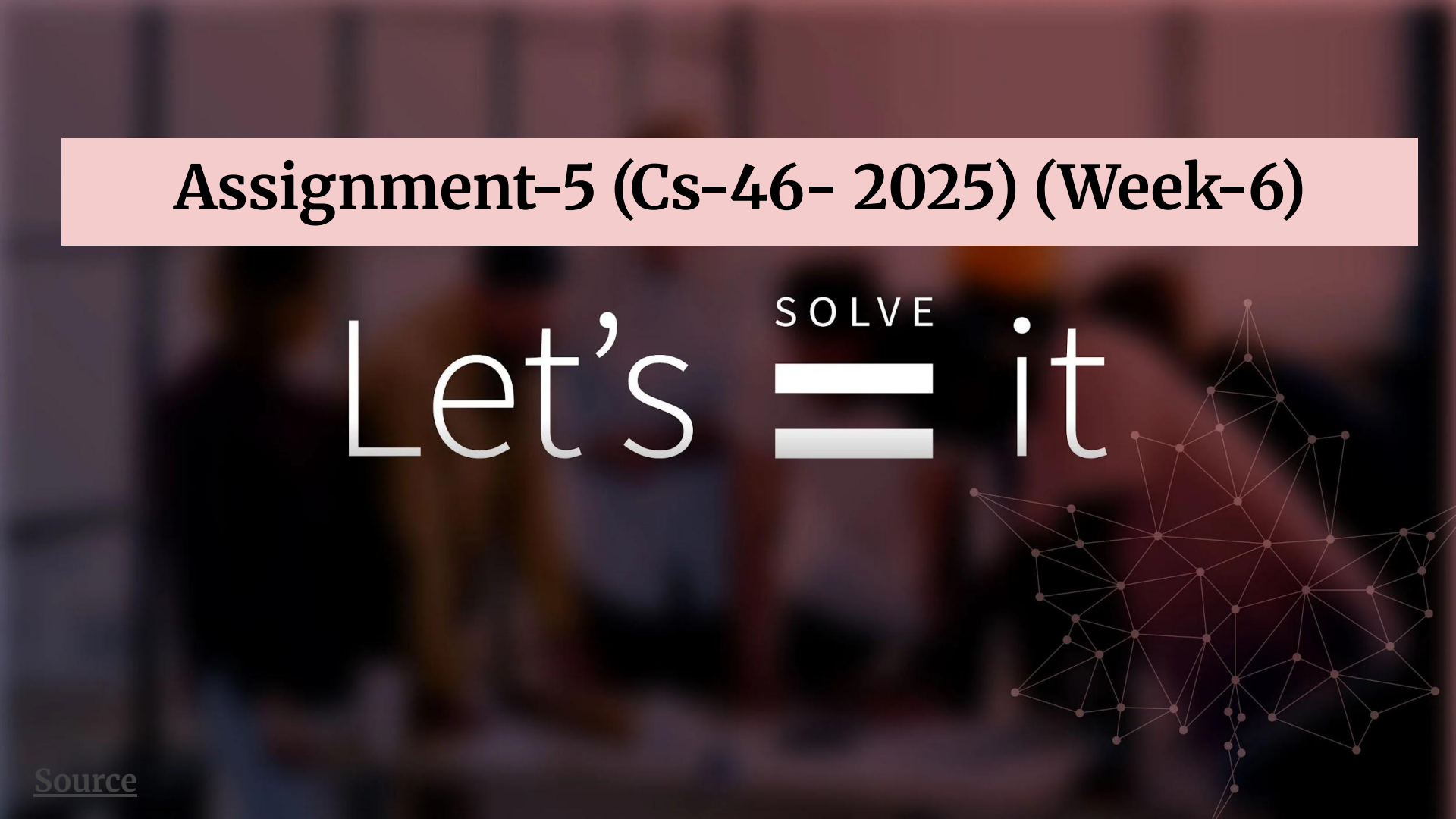
Which among the following split-points for feature2 would give the best split according to the misclassification error?

a) 186.5
b) 188.6
c) 189.2
d) 198.1

feature1	*feature2*	*output*
18.3	187.6	a
14.7	184.9	a
19.4	193.3	a
17.9	180.5	a
19.1	189.1	a
17.6	191.9	b
19.9	190.2	b
17.3	198.6	b
18.7	182.6	b
15.2	187.3	b

Correct options: (c)

Assignment-5 (Cs-46- 2025) (Week-6)



Source

Question-1

01:00

Statement: Decision Tree is an unsupervised learning algorithm.

Reason: The splitting criterion use only the features of the data to calculate their respective measures

a) Statement is True. Reason is True.
b) Statement is True. Reason is False
c) Statement is False. Reason is True
d) Statement is False. Reason is False

Question-1- Correct answer

Statement: Decision Tree is an unsupervised learning algorithm.

Reason: The splitting criterion use only the features of the data to calculate their respective measures

a) **Statement is True. Reason is True.**

b) **Statement is True. Reason is False**

c) **Statement is False. Reason is True**

d) **Statement is False. Reason is False**

Correct options: (d)

Question-2

01:00

Increasing the pruning strength in a decision tree by reducing the maximum depth:

a) Will always result in improved validation accuracy.
b) Will lead to more overfitting
c) Might lead to underfitting if set too aggressively
d) Will have no impact on the tree's performance.
e) Will eliminate the need for validation data.

Question-2- Correct answer

Increasing the pruning strength in a decision tree by reducing the maximum depth:

a) Will always result in improved validation accuracy.
b) Will lead to more overfitting
c) Might lead to underfitting if set too aggressively
d) Will have no impact on the tree's performance.
e) Will eliminate the need for validation data.

Correct options: (c)

Question-3

01:00

What is a common indicator of overfitting in a decision tree?

a) The training accuracy is high while the validation accuracy is low.
b) The tree is shallow.
c) The tree has only a few leaf nodes.
d) The tree's depth matches the number of attributes in the dataset.
e) The tree's predictions are consistently biased.

Question-3 - Correct answer

What is a common indicator of overfitting in a decision tree?

a) The training accuracy is high while the validation accuracy is low.
b) The tree is shallow.
c) The tree has only a few leaf nodes.
d) The tree's depth matches the number of attributes in the dataset.
e) The tree's predictions are consistently biased.

Correct options: (a)

Question-4

03:00

Consider the following statements:

Statement 1: Decision Trees are linear non-parametric models.

Statement 2: A decision tree may be used to explain the complex function learned by a neural network.

a) Both the statements are True.

b) Statement 1 is True, but Statement 2 is False.

c) Statement 1 is False, but Statement 2 is True.

d) Both the statements are False.

Question-4 - Correct answer

Consider the following statements:

Statement 1: Decision Trees are linear non-parametric models.

Statement 2: A decision tree may be used to explain the complex function learned by a neural network.

a) **Both the statements are True.**
b) **Statement 1 is True, but Statement 2 is False.**
c) **Statement 1 is False, but Statement 2 is True.**
d) **Both the statements are False.**

Correct options: (c)

Question-5

01:00

Entropy for a 50-50 split between two classes is:

a) 0

b) 0.5

c) 1

d) None of the above

Question-5 - Correct answer

Entropy for a 50-50 split between two classes is:

a) 0

b) 0.5

c) 1

d) None of the above

Correct options: (c)

Question-6

01:00

Consider a dataset with only one attribute(categorical). Suppose, there are 10 unordered values in this attribute, how many possible combinations are needed to find the best split-point for building the decision tree classifier?

a) 1024
b) 511
c) 1023
d) 512

Question-6 - Correct answer

Consider a dataset with only one attribute(categorical). Suppose, there are 10 unordered values in this attribute, how many possible combinations are needed to find the best split-point for building the decision tree classifier?

a) 1024
b) 511
c) 1023
d) 512

Correct options: (b)

Question-7

03:00

Consider the following dataset:
What is the initial entropy of Malignant?

Age	Vaccination	Tumor Size	Tumor Site	Malignant
5	1	Small	Shoulder	0
9	1	Small	Knee	0
6	0	Small	Marrow	0
6	1	Medium	Chest	0
7	0	Medium	Shoulder	0
8	1	Large	Shoulder	0
5	1	Large	Liver	0
9	0	Small	Liver	1
8	0	Medium	Shoulder	1
8	0	Medium	Shoulder	1
6	0	Small	Marrow	1
7	0	Small	Chest	1

a) 0.543

b) 0.9798

c) 0.8732

d) 1

Question-7 - Correct answer

Consider the following dataset:

What is the initial entropy of Malignant?

Age	Vaccination	Tumor Size	Tumor Site	Malignant
5	1	Small	Shoulder	0
9	1	Small	Knee	0
6	0	Small	Marrow	0
6	1	Medium	Chest	0
7	0	Medium	Shoulder	0
8	1	Large	Shoulder	0
5	1	Large	Liver	0
9	0	Small	Liver	1
8	0	Medium	Shoulder	1
8	0	Medium	Shoulder	1
6	0	Small	Marrow	1
7	0	Small	Chest	1

a) 0.543
b) 0.9798
c) 0.8732
d) 1

Correct options: (b)

Question-8

03:00

Consider the following dataset:
What is the info gain of Vaccination?

Age	Vaccination	Tumor Size	Tumor Site	Malignant
5	1	Small	Shoulder	0
9	1	Small	Knee	0
6	0	Small	Marrow	0
6	1	Medium	Chest	0
7	0	Medium	Shoulder	0
8	1	Large	Shoulder	0
5	1	Large	Liver	0
9	0	Small	Liver	1
8	0	Medium	Shoulder	1
8	0	Medium	Shoulder	1
6	0	Small	Marrow	1
7	0	Small	Chest	1

a) 0.4763

b) 0.2102

c) 0.1134

d) 0.9355

Question-8 - Correct answer

Consider the following dataset:

What is the info gain of Vaccination?

Age	Vaccination	Tumor Size	Tumor Site	Malignant
5	1	Small	Shoulder	0
9	1	Small	Knee	0
6	0	Small	Marrow	0
6	1	Medium	Chest	0
7	0	Medium	Shoulder	0
8	1	Large	Shoulder	0
5	1	Large	Liver	0
9	0	Small	Liver	1
8	0	Medium	Shoulder	1
8	0	Medium	Shoulder	1
6	0	Small	Marrow	1
7	0	Small	Chest	1

a) 0.4763
b) 0.2102
c) 0.1134
d) 0.9355

Correct options: (a)


THANK YOU

Suggestions and Feedback

Next Session:

Tuesday:
11-Mar-2025
6:00 - 8:00 PM