1) Regression Problem

Goal: Split to Reduce Impurity (Variance)

In **regression trees**, unlike classification (which uses **entropy** or **Gini impurity**), we use **variance** as the impurity measure. We aim to **split the data so that child nodes have less variance** in the target values.

We want to split dataset D into:

- D_l **(left)** and D_r **(right)** by choosing the best **feature** X(i) and split **value** v.

The objective is to **maximize reduction in impurity (variance)**

Score=$|D| \cdot Var(D) - |D_L| \cdot Var(D_L) - |D_R| \cdot Var(D_R)$

|D| = number of samples in parent node

- Var(D) = variance of target values in parent
- Var(D_L), Var(D_R) = variances in the left and right child nodes

$$\mathrm{Var}(D) = \frac{1}{|D|} \sum_{i \in D} (y_i - \bar{y}_D)^2$$

We **maximize this score** — i.e., the **reduction in total variance** — to find the best split.

Example: Predicting House Prices Based on Size

Goal: Predict the price of a house based on its size (in square feet).

Here's a small dataset of recent home sales:

House	Size (sqft)	Price ($1000s)
A	600	150
B	800	180
C	1000	200
D	1200	240
E	1500	300
F	1800	330

Let's Build a Regression Tree

Step 1: Try Splitting on Size < 1000

- **Left group** (small houses): 600, 800 → Prices: [150, 180]
  - Mean = 165, Variance = 225
- **Right group** (bigger houses): 1000, 1200, 1500, 1800 → Prices: [200, 240, 300, 330]
  - Mean = 267.5, Variance = 2431.25
- **Parent group** (all prices):
  - Mean = 233.33, Variance = 4030.56

Now calculate the **variance reduction**:

$\text{Score} = 6 \cdot 4030.56 - 2 \cdot 225 - 4 \cdot 2431.25 = 24183.36 - 450 - 9725 = \mathbf{14008.36}$

Step 2: Try Splitting on Size < 1200

- **Left group**: 600, 800, 1000 → Prices: [150, 180, 200]
    - Mean = 176.67, Variance = 422.22
- **Right group**: 1200, 1500, 1800 → Prices: [240, 300, 330]
    - Mean = 290, Variance = 1388.89

Score=6·4030.56−3·422.22−3·1388.89=24183.36−1266.67−4166.67=**18750.02**

✅ This split is **better** because it reduces variance **more**. So, we split on **Size < 1200**.

Regression tree (first level only)

```
      [Size < 1200?]
         /      \
       Yes        No
    Predict: 177   Predict: 290
```

Left side: Sizes = [600, 800, 1000] → Prices = [150, 180, 200]
Right side: Sizes = [1200, 1500, 1800] → Prices = [240, 300, 330]

Step 2: Split the Left Subtree (600, 800, 1000)

Let's try **Size < 800** as a possible split within that left side:

- **Left-Left (LL)**: 600 → Price = [150] → Variance = 0
- **Left-Right (LR)**: 800, 1000 → Prices = [180, 200] → Mean = 190, Variance = 100

Score=3·422.22−1·0−2·100=1266.67−0−200=1066.67

Let's try another split: **Size < 1000**

- **LL**: 600, 800 → [150, 180] → Mean = 165, Var = 225
- **LR**: 1000 → [200] → Var = 0

Score=3·422.22−2·225−1·0=1266.67−450=816.67

✅ **Best split is Size < 800** (higher score).

Step 3: Split the Right Subtree (1200, 1500, 1800)

Try **Size < 1500**

- **Right-Left (RL)**: 1200 → Price = 240 → Var = 0
- **Right-Right (RR)**: 1500, 1800 → Prices: [300, 330] → Mean = 315, Var = 225

$\text{Score} = 3 \cdot 1388.89 - 1 \cdot 0 - 2 \cdot 225 = 4166.67 - 450 = 3716.67$

This is the only real split here (we only have 3 points), but it helps!

Updated Regression Tree (Two Levels Deep)

```
                  [Size < 1200?]
                   /                    \
         [Size < 800?]           [Size < 1500?]
          /        \                      /         \
 Predict:150  Predict:190  Predict:240  Predict:315
```

Final Predictions:

Size	Prediction
600	150
800	190
1000	190
1200	240
1500	315
1800	315

2) Classification example:

Information Gain Explanation and Decision Tree Construction

Dataset: (6 emails)

Email	Contains "Free" (X1)	Has Attachment (X2)	Sender in Contacts (X3)	Is Spam (Y)
1	Yes	No	No	Yes
2	Yes	Yes	No	Yes
3	Yes	No	No	Yes
4	No	Yes	Yes	No
5	No	No	Yes	No
6	No	No	Yes	No

Step 1: Calculate Entropy of the target (Y) before split

Entropy measures impurity (uncertainty). For binary classification:

$H(Y) = -p_{yes} \log_2(p_{yes}) - p_{no} \log_2(p_{no})$

Count spam (Yes) and not spam (No):
- Spam (Yes): 3 (Emails 1, 2, 3)
- Not Spam (No): 3 (Emails 4, 5, 6)

$p_{yes} = 3/6 = 0.5$, $p_{no} = 3/6 = 0.5$

Entropy:
$H(Y) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

Entropy before split = 1 (maximum uncertainty).

Step 2: Calculate Entropy after splitting by each feature

Calculate expected entropy after splitting on each feature, then compute Information Gain

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

Feature X1: "Contains Free"

- For X1=Yes: Emails 1, 2, 3 (All spam) ⇒ Entropy = 0
- For X1=No: Emails 4, 5, 6 (All not spam) ⇒ Entropy = 0

Weighted entropy after split:
$H(Y|X1) = (3/6) * 0 + (3/6) * 0 = 0$
Information Gain:
$IG(X1) = 1 - 0 = 1$

Feature X2: "Has Attachment"

- For X2=Yes: Emails 2, 4 (1 spam, 1 not spam) ⇒ Entropy = 1
- For X2=No: Emails 1, 3, 5, 6 (2 spam, 2 not spam) ⇒ Entropy = 1

Weighted entropy after split:
$H(Y|X2) = (2/6) * 1 + (4/6) * 1 = 1$
Information Gain:
$IG(X2) = 1 - 1 = 0$

Feature X3: "Sender in Contacts"

- For X3=Yes: Emails 4, 5, 6 (All not spam) ⇒ Entropy = 0
- For X3=No: Emails 1, 2, 3 (All spam) ⇒ Entropy = 0

Weighted entropy after split:
$H(Y|X3) = (3/6) * 0 + (3/6) * 0 = 0$
Information Gain:
$IG(X3) = 1 - 0 = 1$

Step 3: Which feature is best?

Both X1 and X3 have the highest information gain (1), meaning they perfectly split spam and not spam.
X2 has no gain and is not useful initially.

Step 4: Build the Decision Tree Using the Best Feature

We choose X1 ("Contains 'Free'") as the root node.

```
| X1 = Yes            | X1 = No                |
|---------------------|------------------------|
| Emails 1, 2, 3 (Spam)| Emails 4, 5, 6 (Not Spam)|
```

- For X1 = Yes branch: All emails are spam ⇒ Leaf: Spam
- For X1 = No branch: All emails are not spam ⇒ Leaf: Not Spam

Final Decision Tree:

```
        [Contains "Free"?]
         /           \
       Yes           No
     (Spam)        (Not Spam)
```

Alternative: Using X3 ("Sender in Contacts")

```
| X3 = Yes            | X3 = No                |
|---------------------|------------------------|
| Emails 4, 5, 6 (Not Spam) | Emails 1, 2, 3 (Spam) |
```

- For X3 = Yes branch: Not Spam
- For X3 = No branch: Spam

```
        [Sender in Contacts?]
         /           \
       Yes           No
   (Not Spam)       (Spam)
```

Summary:

- Splitting on X1 or X3 fully separates the dataset into pure classes.
- Splitting on X2 does not reduce uncertainty.
- Choose either X1 or X3 as the first split in your decision tree.