JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

# EN.601.464 Artificial Intelligence

## Final Project: Decision Trees

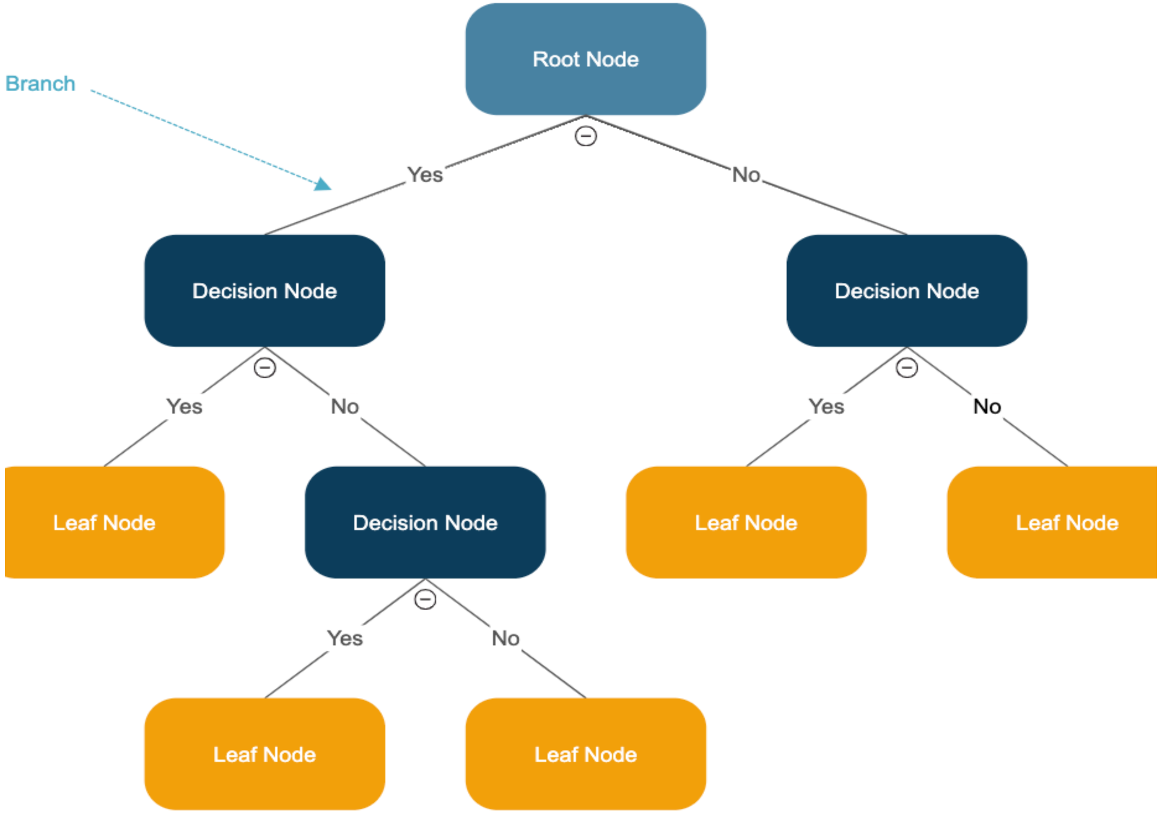
**TEAM MEMBER**

Danny (Iou-Sheng) Chang
Austin (Ching-Yang) Huang
Yutai Wang
Zijun Ding
Xinzhuo Yan
Bingchen Lu

# Decision Trees

# Introduction

- A non-parametric supervised learning algorithm for both classification and regression.
- Has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.
- Learning by recursive partitioning




Source: https://www.smartdraw.com/decision-tree/

# Pros and Cons

## Pros

- Less effort for data preparation during pre-processing.
- Does not require normalization or scale of data
- Missing values in dataset are allowed
- Intuitive and easy to explain

## Cons

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- Greedy search approach during construction leads to high cost in training
- Not good for huge dataset

# Algorithms for Splitting

# Algorithms for Splitting - Gini Impurity

## What is it? [1]

- Gini Impurity is a measure used in decision tree algorithms to determine the best feature to split on at each step of the tree. It measures the impurity or randomness of a set of data points, with higher values indicating greater impurity.
- The Gini Impurity ranges from 0 (perfect purity, all data points in a group belong to the same class) to 0.5 (maximal impurity, data points are evenly distributed across all classes).

$$Gini = 1 - \sum_{i=1}^{j} P(i)^2$$

Where j represents the number of classes in the target variable. $P(i)$ represents the ratio of Results/Total number of observations in node.

# Algorithms for Splitting - Gini Impurity

## The goal? [2]

- Gini Impurity tells us what is the probability of misclassifying an observation. The goal is to minimize the Gini Impurity when choosing the best split for a node.

## Limitations? [3]

- Gini impurity **only operates on the categorical target variables** in terms of "success" or "failure" and **performs only binary split.**

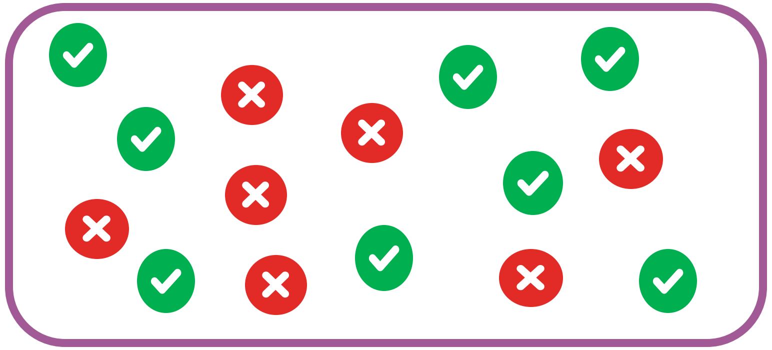
## Now let's look at an example!

# Algorithms for Splitting - Gini Impurity

## Example [4]

Dataset: 15 data points of student data on pass or fail an online ML exam

| No. | Exam Result | Other online courses | Student background | Working status |
|---|---|---|---|---|
| 1 | Pass | Y | Math | NW |
| 2 | Fail | N | Math | W |
| 3 | Fail | Y | Math | W |
| 4 | Pass | Y | CS | NW |
| 5 | Fail | N | Other | W |
| 6 | Fail | Y | Other | W |
| 7 | Pass | Y | Math | NW |
| 8 | Pass | Y | CS | NW |
| 9 | Pass | N | Math | W |
| 10 | Pass | N | CS | W |
| 11 | Pass | Y | CS | W |
| 12 | Pass | N | Math | NW |
| 13 | Fail | Y | Other | W |
| 14 | Fail | N | Other | NW |
| 15 | Fail | N | Math | W |



How to split it?

# Algorithms for Splitting - Gini Impurity

## Example 4

Let us calculate the Gini index for the root node for Student Background attribute.

- Math sub node: 4Pass, 3Fail

$$Gini_{Math} = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.4897$$

- CS sub node: 4Pass, 0 Fail

$$Gini_{CS} = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$$

- Others sub node: 0Pass, 4 Fail

$$Gini_{others} = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

- The overall Gini Impurity for this split:

$$Gini_{Background} = \frac{7}{15} \times 0.4897 + \frac{4}{15} \times 0 + \frac{4}{15} \times 0 = 0.2286$$

# Algorithms for Splitting - Gini Impurity

## Example [4]

Similarly, we can also compute the Gini Impurity for Working Status and Online Courses:

- Working/ Not working:

$$Gini_{working} = 1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2 = 0.44$$

$$Gini_{notworking} = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 0.278$$

$$Gini_{workstatus} = \frac{9}{15} \times 0.44 + \frac{6}{15} \times 0.2780 = 0.378$$

- Online courses:

$$Gini_{online} = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 = 0.4688$$

$$Gini_{notonline} = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.4898$$

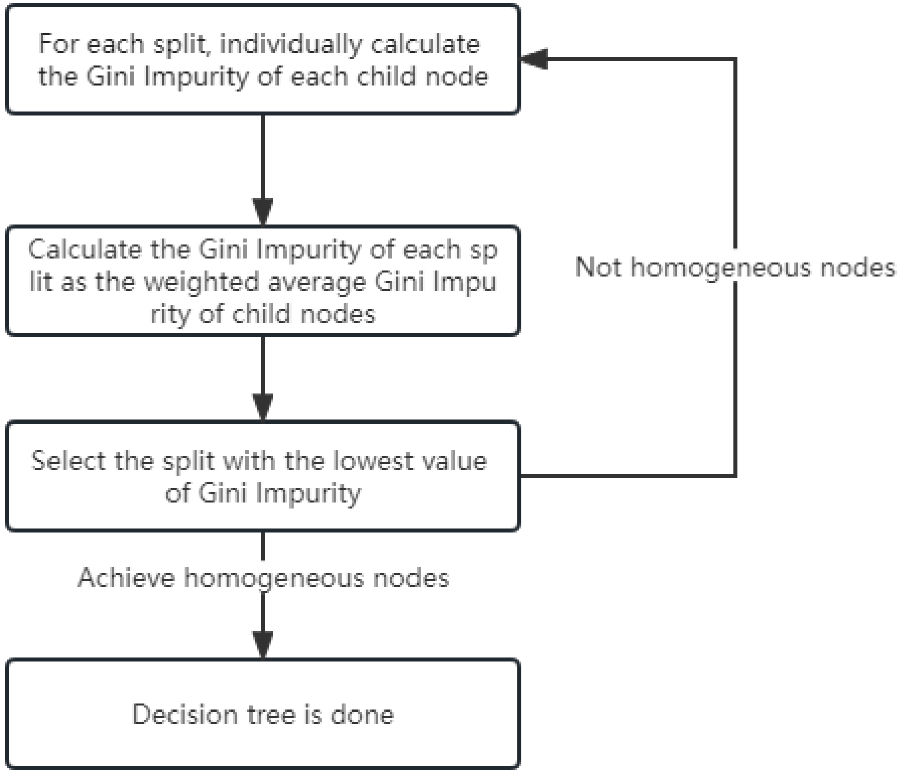
$$Gini_{online} = \frac{8}{15} \times 0.4688 + \frac{7}{15} \times 0.4898 = 0.479$$

- **The Gini Impurity is lowest for the Student Background variable. Hence, we pick this variable for the root node.** In a similar fashion we would again proceed to move down the tree, carrying out splits where node purity is less.

# Algorithms for Splitting - Gini Impurity

## Conclusion

The flow chart to split a decision tree using Gini Impurity:



# Algorithms for Splitting - Gini Impurity

## Conclusion[4]

### Advantages

1. Gini impurity is faster to compute, especially when the number of classes is small.
2. Gini impurity tends to perform well in practice, particularly when the tree is not deep (i.e., it has few levels).

### Disadvantages

1. Gini impurity assumes that every split is binary, which may not be the case for all decision trees.

# Algorithms for Splitting – Chi-Square

## What is it? [5]

1. Chi-Square determines if there is a significant relationship between the predictor and the target, allowing the algorithm to choose the most relevant features to split on at each node of the tree.

$$Chi - Square = \sqrt{\frac{(Actual - Expected)^2}{Expected}}$$

- Expected: expected value for a class in a child node based on the distribution of classes in the parent node.
- Actual: the actual value for a class in a child node.

# Algorithms for Splitting - Chi-Square

## The goal?

- When the Chi-Square value is high, it indicates that there is a strong association between the predictor and the target class, making it a good candidate for splitting. The goal is to maximize the Chi-Square value when choosing the best split for a node.

## Limitations?

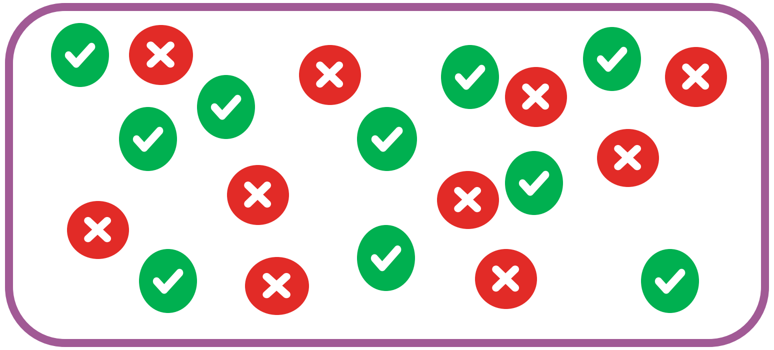
- Chi-Square **only operates on the categorical target variables** in terms of “success” or “failure”**.** However, Chi-Square is capable of making **two or more splits**.

# Now let’s look at an example!

# Algorithms for Splitting - Chi-Square

## Example [6]

Here, we have 20 students’ midterm results: Pass/ Fail



Root Node

Students = 20

Pass = 10

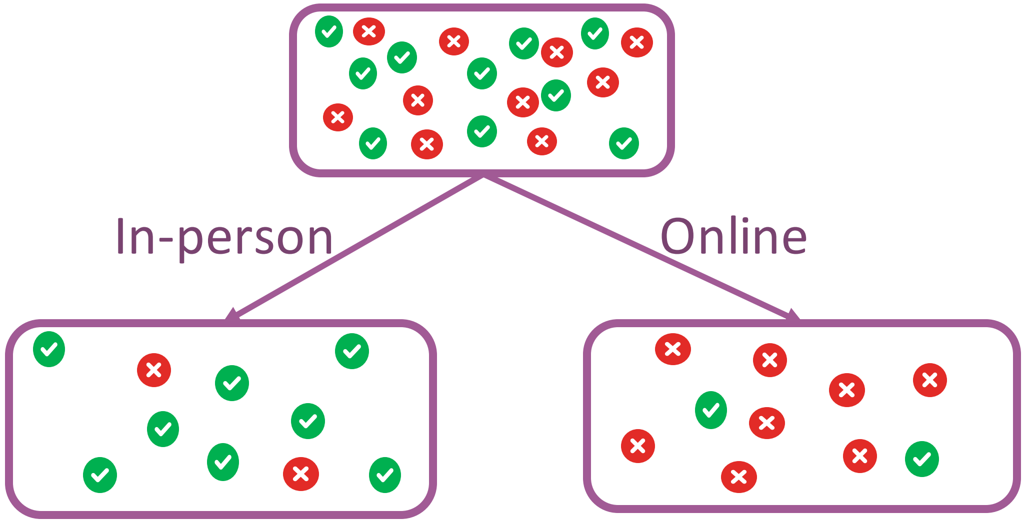
Fail = 10

# Algorithms for Splitting - Chi-Square

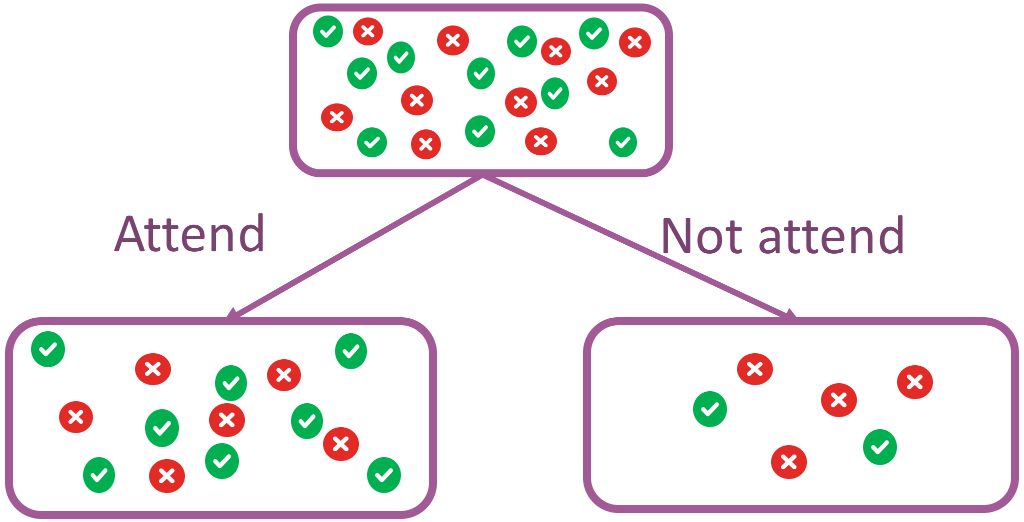
## Example [6]

We can split in two ways:

- Take the online section or in-person section of this course:



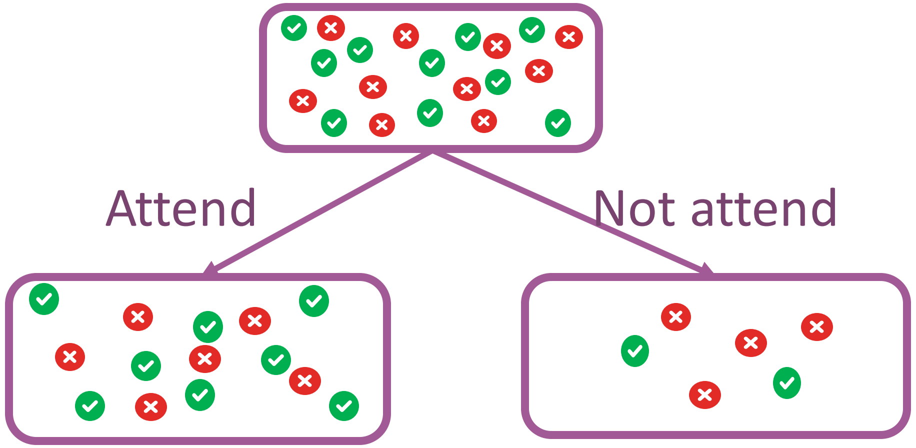

- Attend the office hour or not:



# Algorithms for Splitting - Chi-Square

## Example [6]

Now, let's do the calculation for **Split on attendance in office hour**:




Students = 20

Pass = 10

Fail = 10

Percentage Pass = 50%

Percentage Fail = 50%

$$Expected = Node\,Total \times Percentage$$

| Node | Actual Pass | Actual Fail | Expected Pass | Expected Fail | Deviation Pass | Deviation Fail | Chi-Square(Pass) | Chi-Square(Fail) |
|---|---|---|---|---|---|---|---|---|
| Attend | 8 | 6 | 7 | 7 | 1 | -1 | | |
| Not attend | 2 | 4 | 3 | 3 | -1 | 1 | | |

# Algorithms for Splitting - Chi-Square

## Example [6]

| Node | Actual Pass | Actual Fail | Expected Pass | Expected Fail | Deviation Pass | Deviation Fail | Chi-Square(Pass) | Chi-Square(Fail) |
|---|---|---|---|---|---|---|---|---|
| Attend | 8 | 6 | 7 | 7 | 1 | -1 | 0.38 | 0.38 |
| Not attend | 2 | 4 | 3 | 3 | -1 | 1 | 0.58 | 0.58 |

$$Attend\ Chi - Square(Pass) = \sqrt{\frac{1^2}{7}} = 0.38$$

$$Attend\ Chi - Square(Fail) = \sqrt{\frac{(-1)^2}{7}} = 0.38$$

$$Not\ Attend\ Chi - Square(Pass) = \sqrt{\frac{(-1)^2}{3}} = 0.58$$

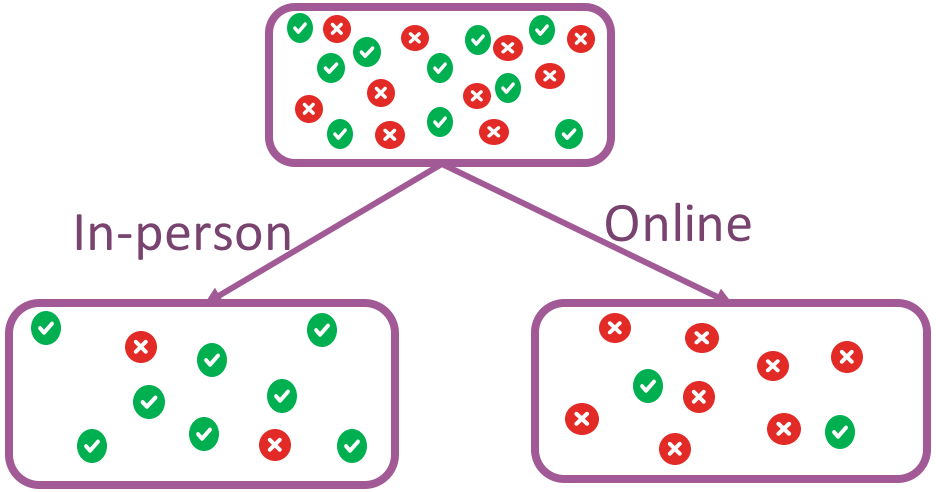
$$Not\ Attend\ Chi - Square(Fail) = \sqrt{\frac{1^2}{3}} = 0.58$$

$$\boldsymbol{Total = 0.38 + 0.38 + 0.58 + 0.58 = 1.92}$$

# Algorithms for Splitting - Chi-Square

## Example [6]

Similarly, we can do the calculation for **Split on different course section**:




Students = 20

Pass = 10

Fail = 10

Percentage Pass = 50%

Percentage Fail = 50%

$$Expected = Node\,Total \times Percentage$$

| Node | Actual Pass | Actual Fail | Expected Pass | Expected Fail | Deviation Pass | Deviation Fail | Chi-Square(Pass) | Chi-Square(Fail) |
|---|---|---|---|---|---|---|---|---|
| In-person | 8 | 2 | 5 | 5 | 3 | -3 | | |
| Online | 2 | 8 | 5 | 5 | -3 | 3 | | |

# Algorithms for Splitting - Chi-Square

## Example [6]

| Node | Actual Pass | Actual Fail | Expected Pass | Expected Fail | Deviation Pass | Deviation Fail | Chi-Square(Pass) | Chi-Square(Fail) |
|---|---|---|---|---|---|---|---|---|
| In-person | 8 | 2 | 5 | 5 | 3 | -3 | 1.34 | 1.34 |
| Online | 2 | 8 | 5 | 5 | -3 | 3 | 1.34 | 1.34 |

$$In-person\ Chi-Square(Pass) = \sqrt{\frac{(8-5)^2}{5}} = 1.34$$

$$In-person\ Chi-Square(Fail) = \sqrt{\frac{(2-5)^2}{5}} = 1.34$$

$$Online\ Chi-Square(Pass) = \sqrt{\frac{(2-5)^2}{5}} = 1.34$$

$$Online\ Chi-Square(Fail) = \sqrt{\frac{(8-5)^2}{5}} = 1.34$$

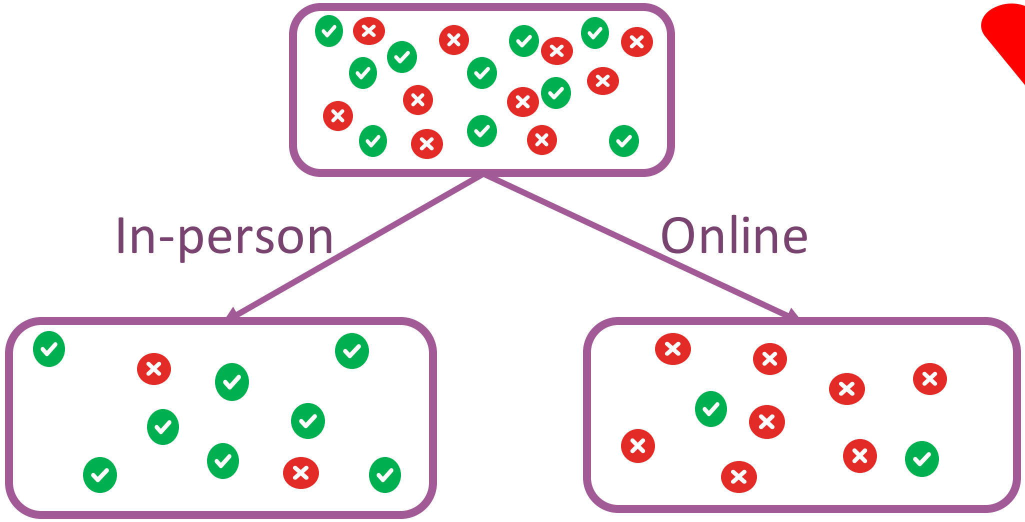
$$\boldsymbol{Total = 1.34 + 1.34 + 1.34 + 1.34 = 5.36}$$

# Algorithms for Splitting - Chi-Square

## Example [6]

Choose the splitting method with the **highest** Chi-Square value!

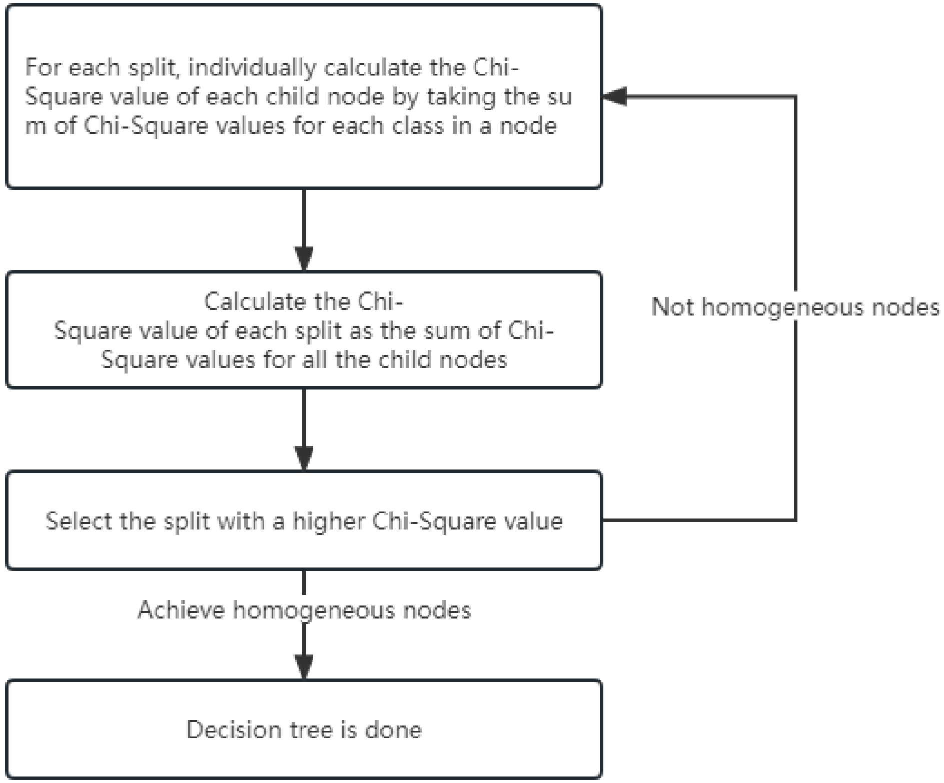
| Split | Chi-Square |
|---|---|
| Office hour attendance (Attend/ Not) | 1.92 |
| Course section (In-person/ Online) | 5.36 |



# Algorithms for Splitting - Chi-Square

## Conclusion

The flow chart to split a decision tree using Chi-Square:



# Algorithms for Splitting - Chi-Square

## Conclusion[7]

### Advantages

1. It is fast!
2. It builds "wider" decision trees, because it is not constrained to make binary splits.
3. It may yield many terminal nodes connected to a single branch, which can be conveniently summarized in a simple two-way contingency table, with multiple categories for each variable.

### Disadvantages

1. Since multiple splits fragment the variable's range into smaller subranges, the algorithm requires larger quantities of data to get dependable results.
2. Variables of the real data-type variables (continuous numbers with decimals) are forced into categorical bins before analysis, which may not be helpful.

# Algorithms for Splitting - Information Gain

## What is it? [5]

- The Information Gain method is used for splitting the nodes when the target variable is categorical, and it works on the concept of entropy.
- Entropy is used for calculating the purity of a node. The lower the value of entropy, the higher the purity of the node. The entropy of a homogeneous node is zero. Since we subtract entropy from 1, the Information Gain is higher for the purer nodes with a maximum value of 1.

$$Information\ Gain = 1 - Entropy$$

$$Entropy = -\sum_{i=1}^{n} p_i \, log_2 p_i$$

where n is the number of samples and pi is the probability that it is a function of entropy.

# Algorithms for Splitting - Information Gain

## The goal? [3]

- Information gain is used to quantify which feature provides the most information about the classification based on the concept of entropy. In general, the aim is to reduce the amount of entropy from the top (root node) to the bottom (leaf nodes).

## Limitations? [10]

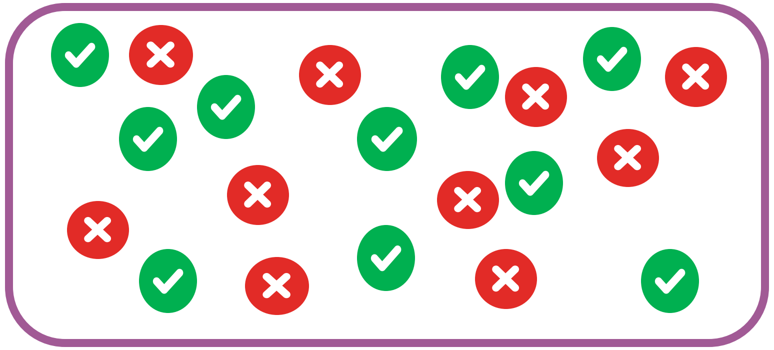
- One of the main limitations of the algorithms for splitting using information gain is their tendency to **overfit** the training data, which can result in poor generalization performance on new, unseen data.

# Now let’s look at an example!

# Algorithms for Splitting - Information Gain

## Example [5]

Here, we have 20 students' midterm results: Pass/ Fail



Root Node

Students = 20

Pass = 10

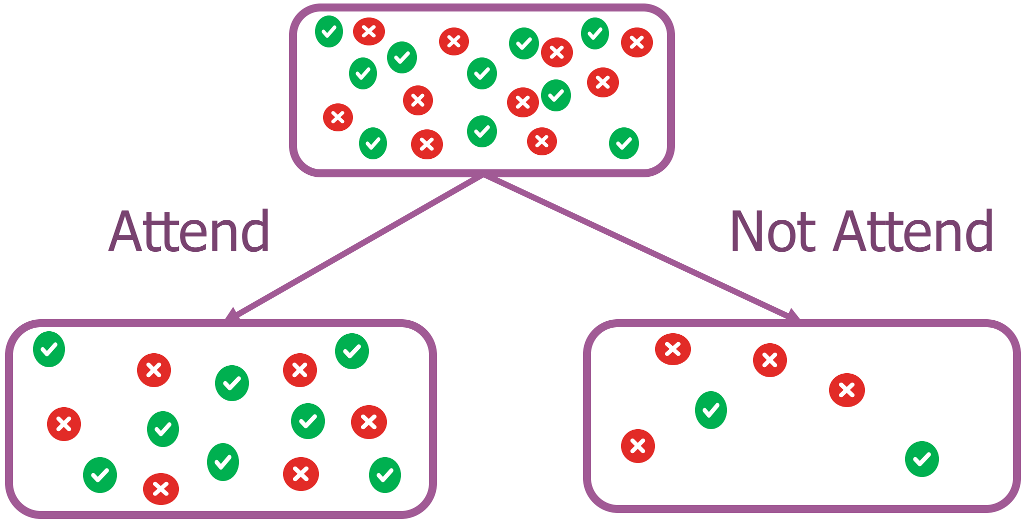
Fail = 10

# Algorithms for Splitting - Information Gain

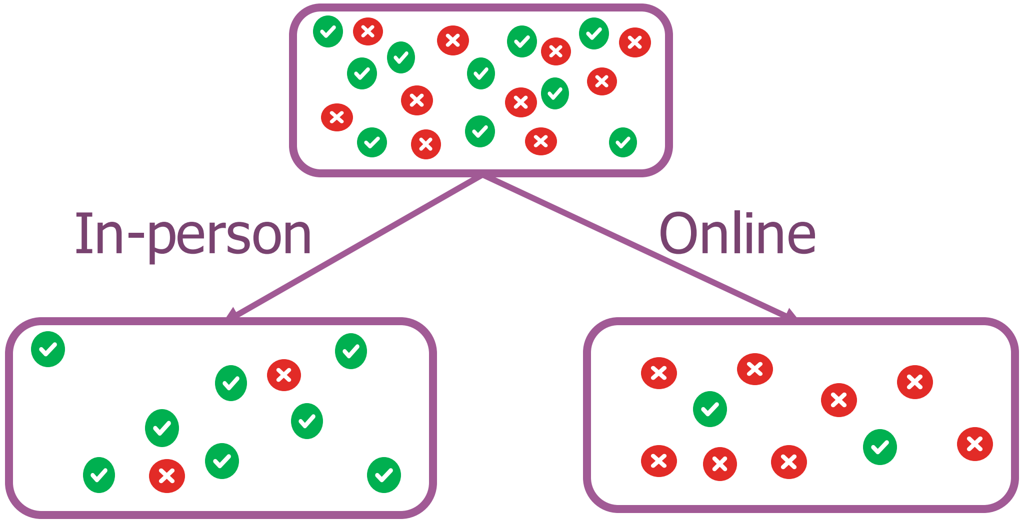
## Example 5

We can split in two ways:

- Attend the office hour or not:




- Take the online section or in-person section of this course:



# Algorithms for Splitting - Information Gain

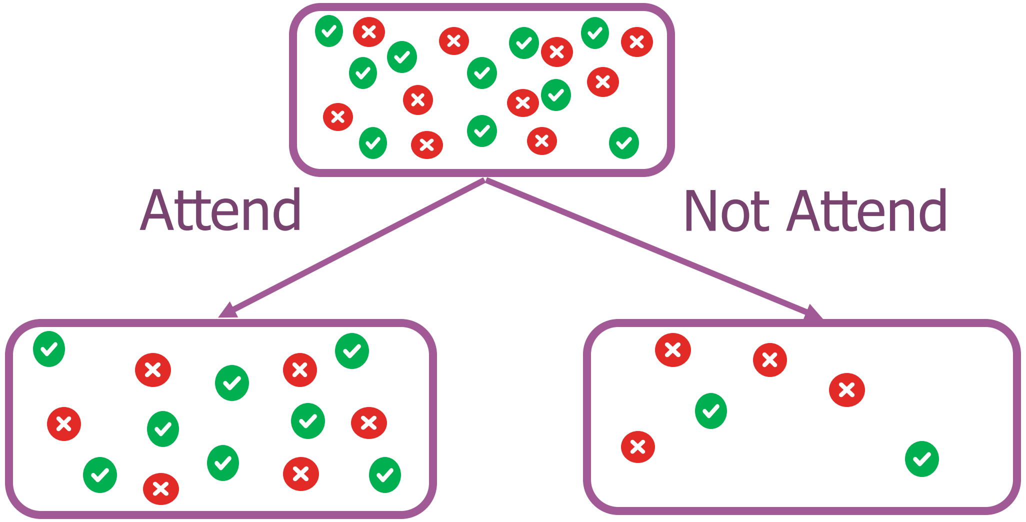
## Example [5]

Now, let’s do the calculation for **Split on attendance in office hour**:

Students = 20

Pass = 10 % Pass = 0.5

Fail = 10 % Fail = 0.5




Students = 14

Pass = 8

Fail = 6

% Pass = 0.57

% Fail = 0.43

Students = 6

Pass = 2

Fail = 4

% Pass = 0.33

% Fail = 0.67

# Algorithms for Splitting - Information Gain

## Example [5]

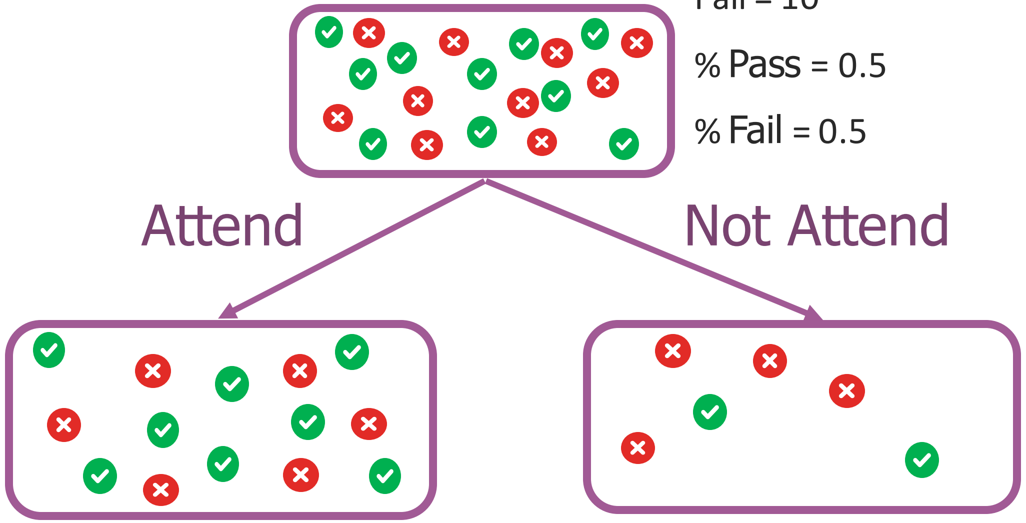
Entropy for parent node:
$-0.5 \times log_2(0.5) - 0.5 \times log_2(0.5) = 1$

Entropy for sub-node Attend:
$-0.57 \times log_2(0.57) - 0.43 \times log_2(0.43) = 0.98$

Entropy for sub-node Not Attend:
$-0.33 \times log_2(0.33) - 0.67 \times log_2(0.67) = 0.91$

Weighted Entropy: attendance in office hour:
$\left(\frac{14}{20}\right) \times 0.98 + \left(\frac{6}{20}\right) \times 0.91 = 0.959$

Students = 20
Pass = 10
Fail = 10
% Pass = 0.5
% Fail = 0.5




Students = 14
Pass = 8
Fail = 6
% Pass = 0.57
% Fail = 0.43

Students = 6
Pass = 2
Fail = 4
% Pass = 0.33
% Fail = 0.67

# Algorithms for Splitting - Information Gain

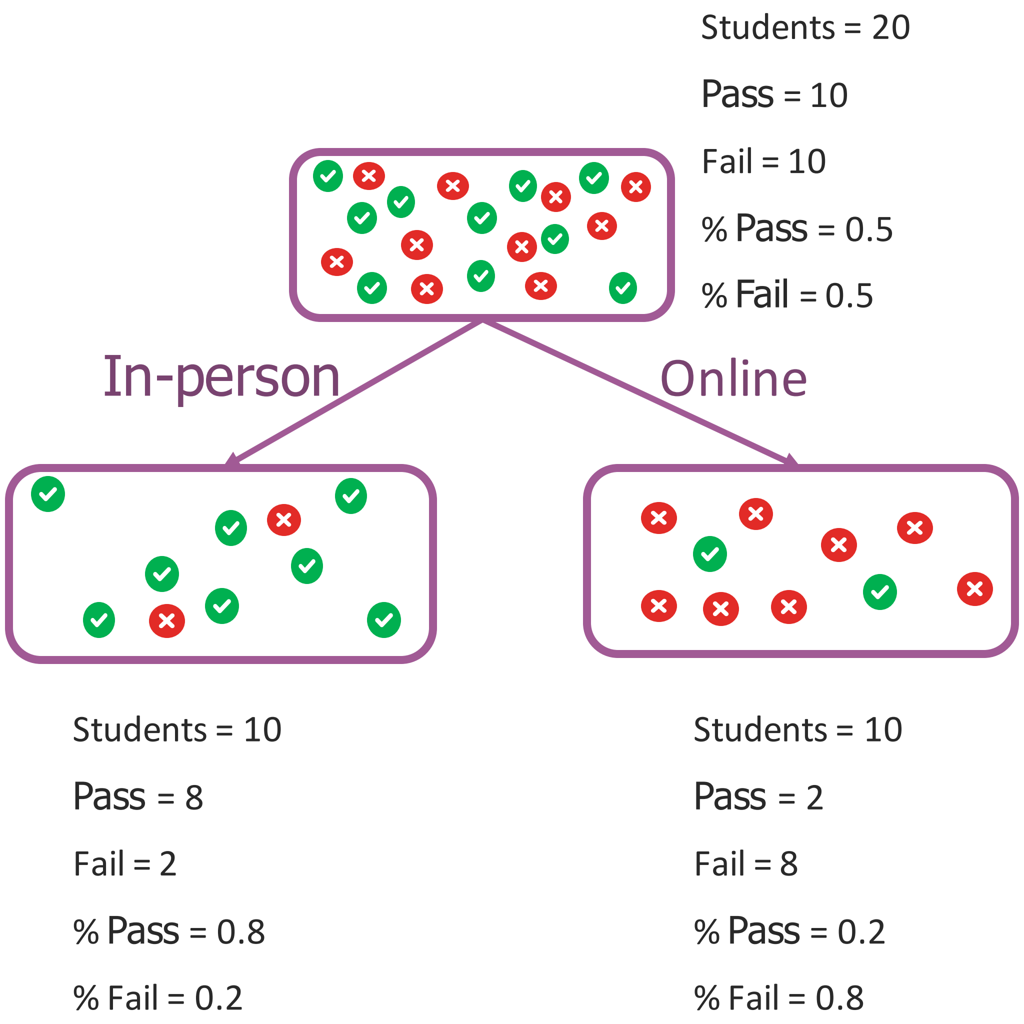
## Example [5]

Entropy for parent node:
$-0.5 \times log_2(0.5) - 0.5 \times log_2(0.5) = 1$

Entropy for sub-node In-person:
$-0.8 \times log_2(0.8) - 0.2 \times log_2(0.2) = 0.722$

Entropy for sub-node Online:
$-0.2 \times log_2(0.2) - 0.8 \times log_2(0.8) = 0.722$

Weighted Entropy: Take the online section or in-person section of this course:
$\left(\frac{10}{20}\right) \times 0.722 + \left(\frac{10}{20}\right) \times 0.722 = 0.722$

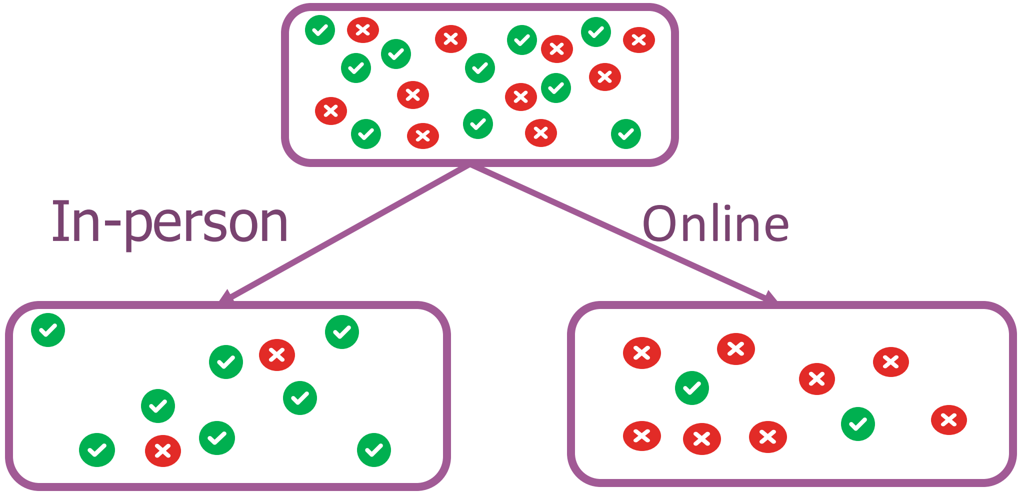


# Algorithms for Splitting - Information Gain

## Example [5]

Choose the splitting method with the **highest** Information Gain value!

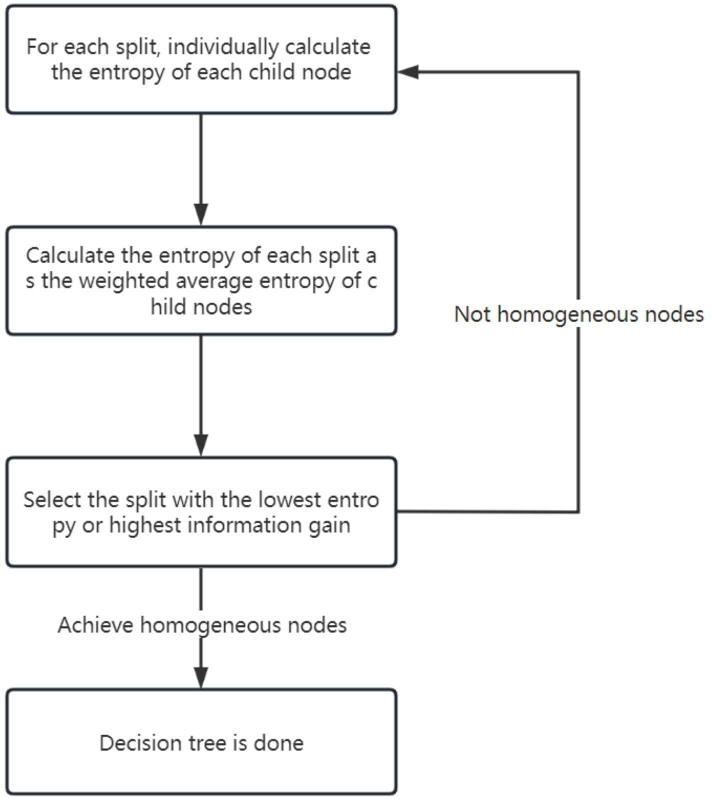
| Split | Entropy | Information Gain |
|---|---|---|
| Office hour attendance (Attend/ Not) | 0.959 | 0.041 |
| Course section (In-person/ Online) | 0.722 | 0.278 |



# Algorithms for Splitting - Information Gain

## Conclusion

The flow chart to split a decision tree using Information Gain:



# Algorithms for Splitting - Information Gain

## Conclusion

**Advantages**[3]

1. Information Gain favors lesser distributions having small count with multiple specific values.
2. Information Gain computes the difference between entropy before and after the split and indicates the impurity in classes of elements.

**Disadvantages**

1. Overfitting: Very good at training, this tree does not perform well at predicting unknown instances[8].
2. If the split information of the attribute is too low, Gain ratio will try to split on the attribute[9].

# Algorithms for Splitting - Reduction in Variance

## What is it? [5]

- Reduction in Variance is a method for splitting the node used when the target variable is continuous, i.e., regression problems. It is called so because it uses variance as a measure for deciding the feature on which a node is split into child nodes.
- Variance is used for calculating the homogeneity of a node. If a node is entirely homogeneous, then the variance is zero.

$$Variance = \frac{\sum(X - \mu)^2}{N}$$

where μ is the mean, N is the sample size, and X is each specific value in the sample.

# Algorithms for Splitting - Reduction in Variance

## The goal? [11]

- The goal of the algorithms for splitting using reduction in variance is to partition the data into subsets that minimize the variance of the target variable within each subset.

## Limitations? [12]

- When the target variable has a **non-linear** relationship with the features, the reduction in variance tends to produce **biased or suboptimal splits**.

# Now let's look at an example!

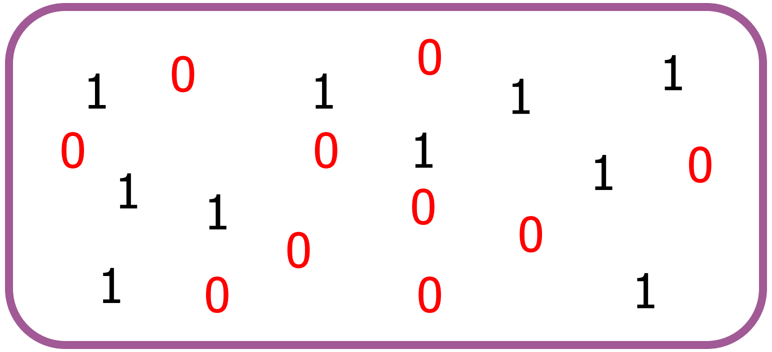
# Algorithms for Splitting - Reduction in Variance

## Example [5]

Here, we have 20 students' midterm results: Pass/ Fail

Pass = 1

Fail= 0




Root Node

Students = 20

Pass = 10

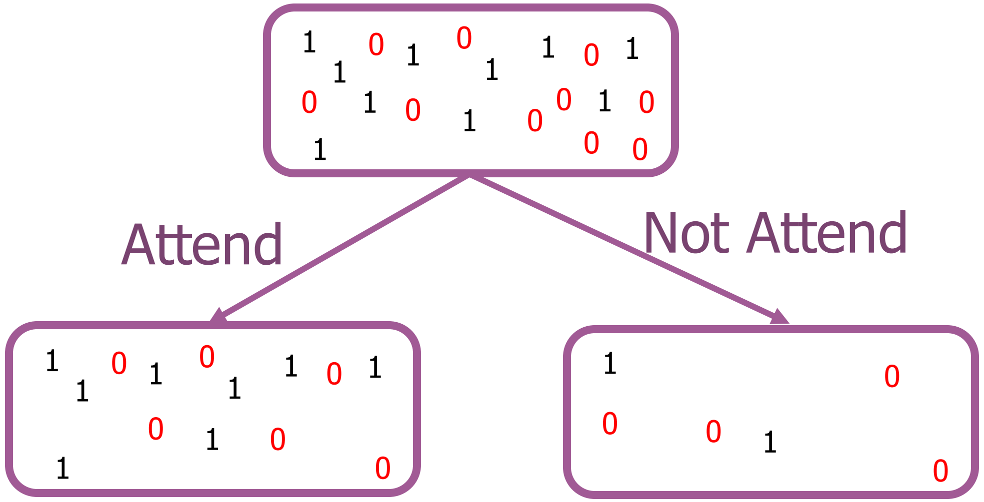
Fail = 10

# Algorithms for Splitting - Reduction in Variance

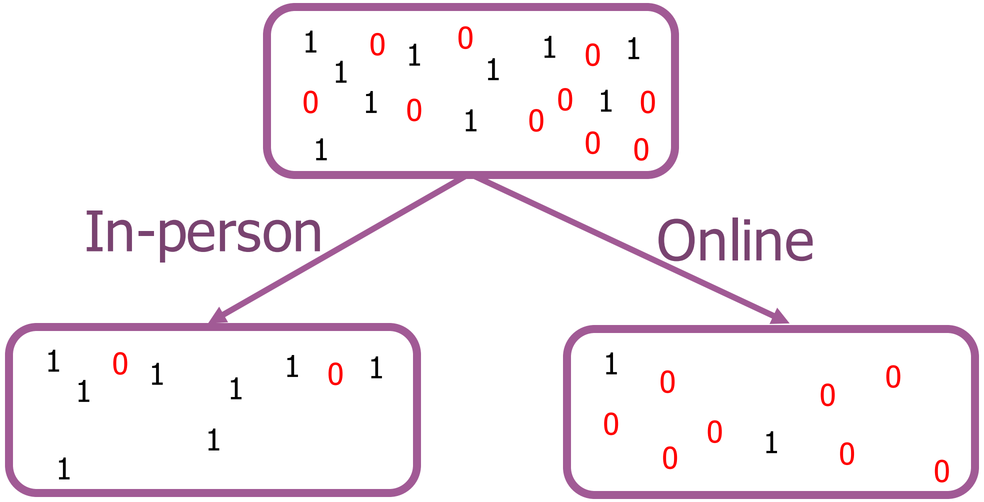
## Example [5]

We can split in two ways:

- Attend the office hour or not:




- Take the online section or in-person section of this course:



# Algorithms for Splitting - Reduction in Variance

## Example [5]

Now, let’s do the calculation for **Split on attendance in office hour**:

Students = 20

Pass = 10 % Pass = 0.5

Fail = 10 % Fail = 0.5

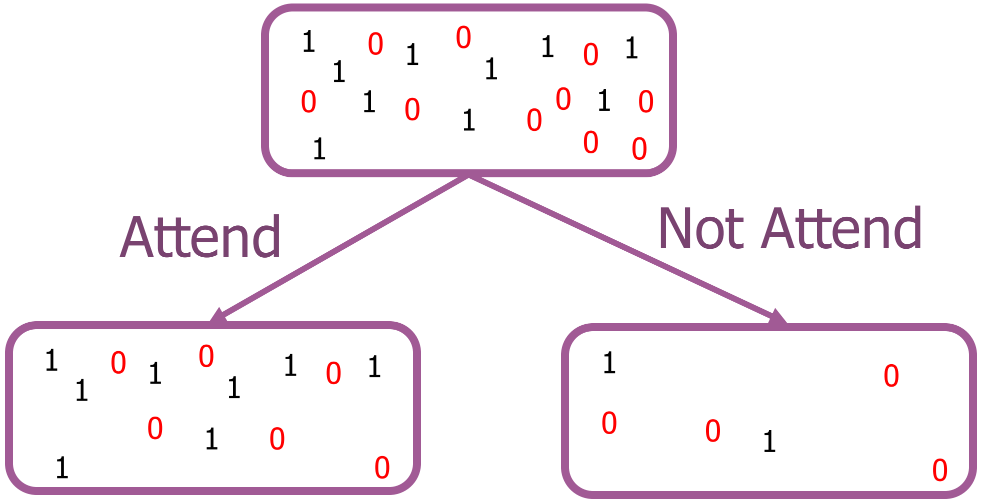
Students = 14

Pass = 8

Fail = 6

% Pass = 0.57

% Fail = 0.43




Students = 6

Pass = 2

Fail = 4

% Pass = 0.33

% Fail = 0.67

# Algorithms for Splitting - Reduction in Variance

## Example [5]

**Attend node:**

Mean $= \frac{(8\times1+6\times0)}{14} = 0.57$

Variance $= \frac{\left(8\times(1-0.57)^2+6\times(0-0.57)^2\right)}{14} = 0.245$

**Not Attend node:**

Mean $= \frac{(2\times1+4\times0)}{6} = 0.33$

Variance $= \frac{\left(2\times(1-0.33)^2+4\times(0-0.33)^2\right)}{6} = 0.222$

**Variance: Split on attendance in office hour:**

$\left(\frac{14}{20}\right)\times 0.245 + \left(\frac{6}{20}\right)\times 0.222 = 0.238$

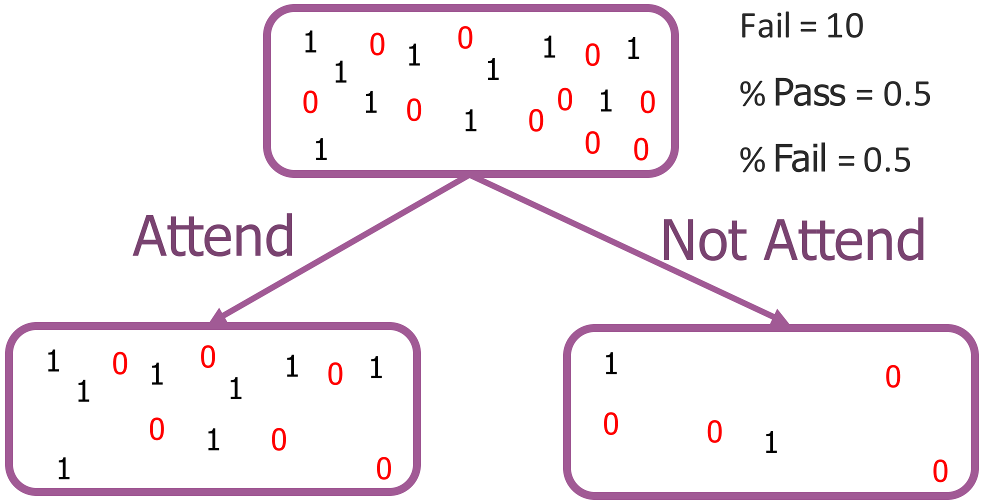
Students = 20

Pass = 10

Fail = 10

% Pass = 0.5

% Fail = 0.5




Students = 14

Pass = 8

Fail = 6

% Pass = 0.57

% Fail = 0.43

Students = 6

Pass = 2

Fail = 4

% Pass = 0.33

% Fail = 0.67

# Algorithms for Splitting - Reduction in Variance

## Example [5]

**In-person node:**

Mean $= \frac{(8\times1+2\times0)}{10} = 0.8$

Variance $= \frac{\left(8\times(1-0.8)^2+2\times(0-0.8)^2\right)}{10} = 0.16$

**Online node:**

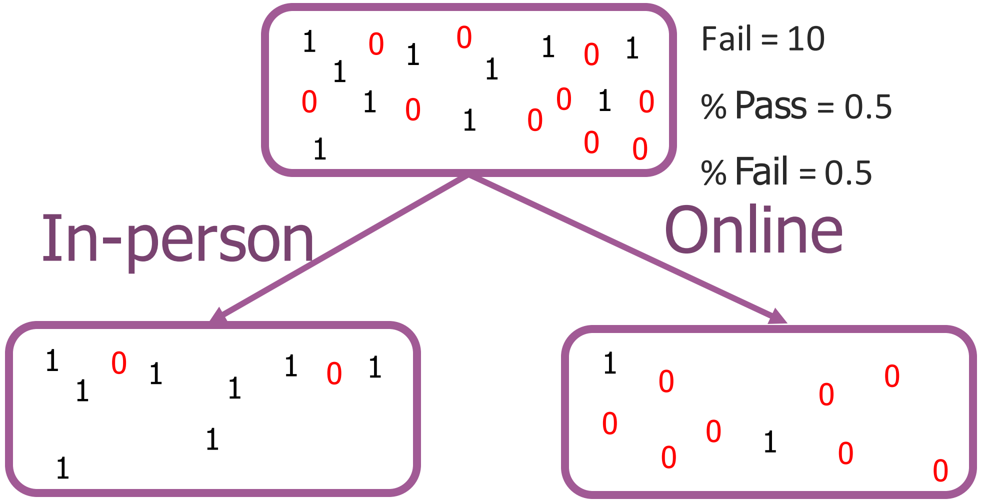
Mean $= \frac{(2\times1+8\times0)}{10} = 0.2$

Variance $= \frac{\left(2\times(1-0.2)^2+8\times(0-0.2)^2\right)}{10} = 0.16$

**Variance: Take the online section or in-person section of this course:**

$$\left(\frac{10}{20}\right)\times 0.16+\left(\frac{10}{20}\right)\times 0.16 = 0.16$$

Students = 20
Pass = 10
Fail = 10
% Pass = 0.5
% Fail = 0.5




Students = 10
Pass = 8
Fail = 2
% Pass = 0.8
% Fail = 0.2

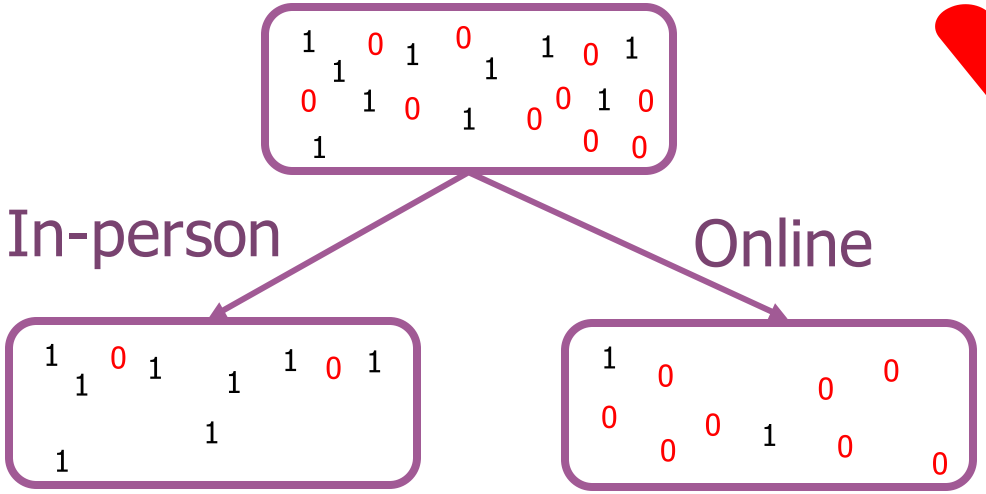
Students = 10
Pass = 2
Fail = 8
% Pass = 0.2
% Fail = 0.8

# Algorithms for Splitting - Reduction in Variance

## Example [5]

Choose the splitting method with the **lowest** Variance value!

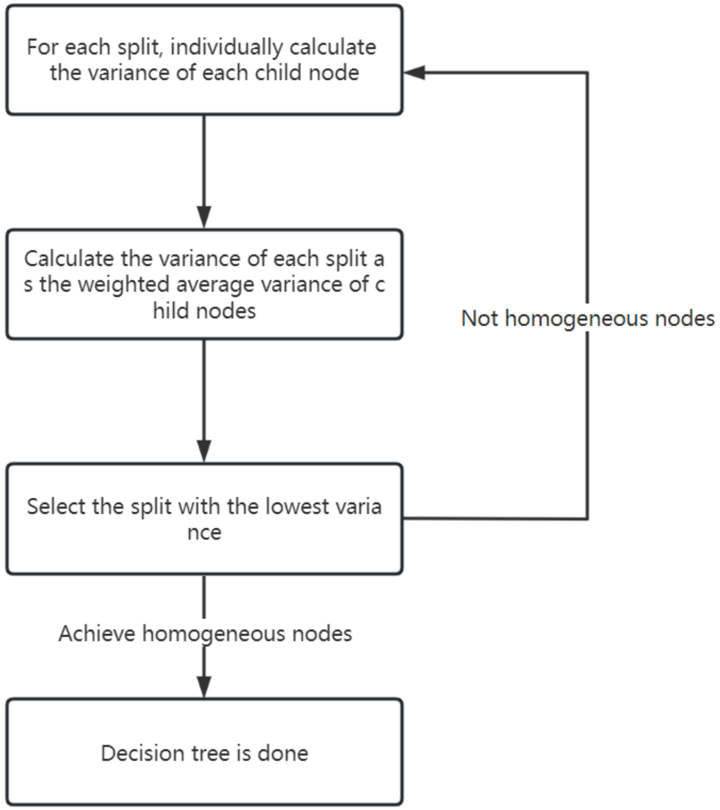
| Split | Variance |
|---|---|
| Office hour attendance (Attend/ Not) | 0.238 |
| Course section (In-person/ Online) | 0.16 |



# Algorithms for Splitting - Reduction in Variance

## Conclusion

The flow chart to split a decision tree using Reduction in Variance:



# Algorithms for Splitting - Reduction in Variance

## Conclusion

**Advantages**[11]

1. Reduction in variance can handle continuous and numerical data.
2. Reduction in variance can result in faster training times and reduced computational complexity.

**Disadvantages**[11]

1. Reduction in variance is too susceptibility to outliers in the data.

# Jupyter Notebook Demo

# Decision Trees Implementation

- Decision Tree Classifier
- Splitting Criterion
  - Gini Impurity
  - Chi-Square
  - Information Gain
  - Reduction in Variance
- Decision Tree Node
- Datasets
- Demo of the Decision Tree Classifier on the Iris Dataset

# [EN.601.664] Artificial Intelligence

- JHU Spring 2023
- Instuctor: Musad Haque, Ph.D. `email` `personal webpage`

## Final Project

### Project Category #2: Decision Trees

| Group Member | JHU email |
|---|---|
| Danny (Iou-Sheng) Chang | `email` ichang9@jhu.edu |
| Austin (Ching-Yang) Huang | `email` chuan120@jhu.edu |
| Yutai Wang | `email` ywang790@jhu.edu |
| Zijun Ding | `email` zding26@jhu.edu |
| Xinzhuo Yan | `email` xyan28@jhu.edu |
| Bingchen Lu | `email` blu16@jhu.edu |

### Deliverables

+ 4 cells hidden

### Import Required Packages

+ 16 cells hidden

### The Decision Tree Classifier

The Decision Tree Classifier, which recursively splits data based on the specified splitting criterion to create a tree structure for predicting class labels. Contains methods to fit the model, predict class labels, and print ASCII visualization of trees.

+ 6 cells hidden

### The Splitting Criterion

The Splitting Criterion class contains all supported node splitting algorithms, computation of the node's purity score, and logging information for debugging purpose.

+ 10 cells hidden

### The Decision Tree Node

The Node class contains the properties for a node in decision tree, and the printing methods for visualizing trees by ASCII.

+ 6 cells hidden

### Datasets

+ 8 cells hidden

### Demonstration of the Decision Tree Classifier

+ 28 cells hidden

# Applications

# Applications [13]

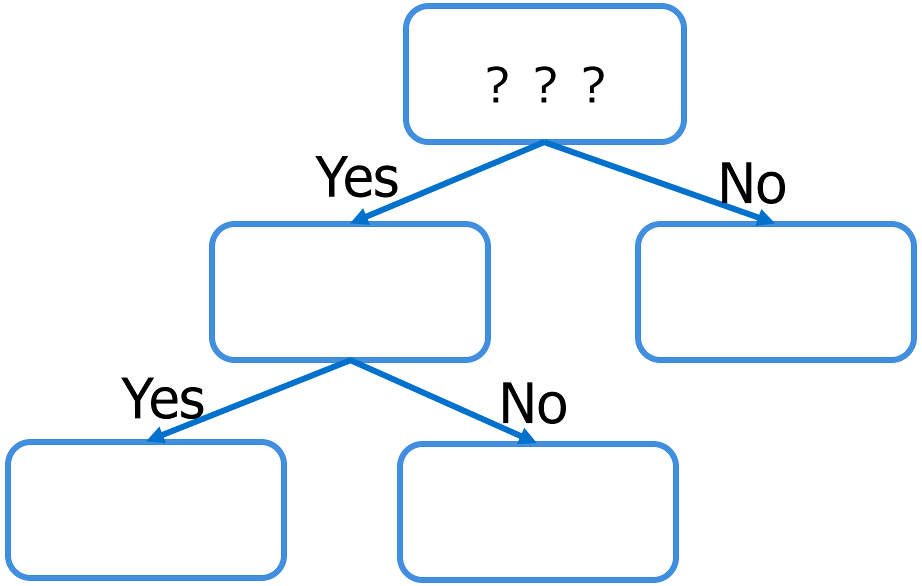
Decision trees are useful for categorizing results where attributes can be sorted against known criteria to determine the final category. Decision trees map possible outcomes of a series of related choices. They are widely used in:

1. **Marketing:** Businesses can use decision trees to enhance the accuracy of their promotional campaigns by observing the performance of their competitors' products and services.
2. **Retention of Customers:** Companies can use decision trees for customer retention through analyzing their behaviors and releasing new offers or products to suit those behaviors.
3. **Diagnosis of Diseases and Ailments:** Decision trees can help physicians and medical professionals in identifying patients that are at a higher risk of developing serious ( or preventable) conditions such as heart disease or dementia.
4. **Detection of Frauds:** Companies can prevent fraud by using decision trees to identify fraudulent behavior beforehand.

# Applications–Diagnosis of Heart Disease [1]

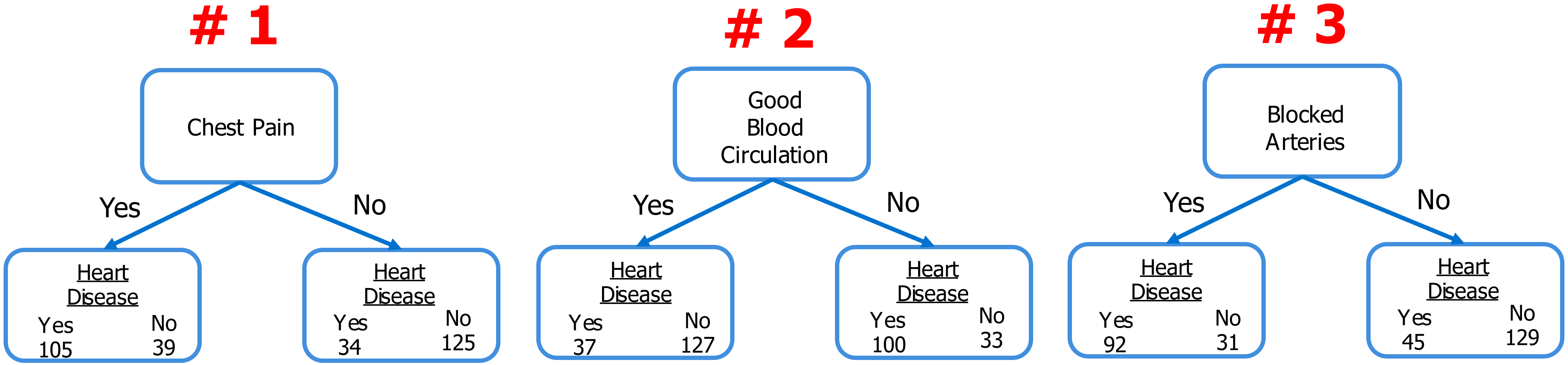
JHU loves to solve medical problems! Let's take a look at this application: **How to build a decision tree to assist the diagnosis of heart disease?**

| Chest Pain | Good Blood Circulation | Blocked Arteries | Heart Disease |
|---|---|---|---|
| Yes | No | Yes | No |
| Yes | Yes | Yes | Yes |
| No | No | Yes | No |
| Yes | Yes | No | Yes |
| No | Yes | No | No |
| Yes | Yes | Yes | No |
| No | No | Yes | Yes |
| Yes | No | Yes | No |
| …… | …… | …… | …… |



# Applications–Diagnosis of Heart Disease [1]

Due to space limitations, we can't put up a complete table here, but the root node can be divided in the following ways:

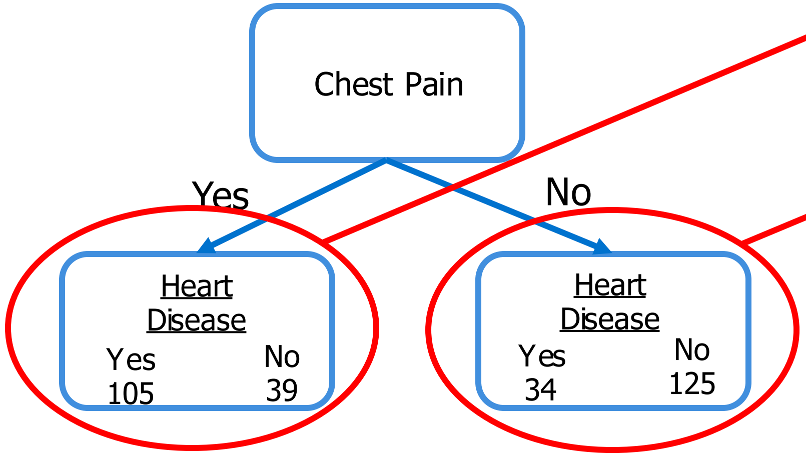



**Which one should we choose? Roll the dice to choose? or...**

# Applications–Diagnosis of Heart Disease [1]

**A trick here:** Most of the time **Gini impurity** is used as it gives good results for splitting and its computation is inexpensive.

**# 1**




**Using equations on slide 5, we get:**

$$Gini\ impurity = 1 - \left[\frac{105}{105+39}\right]^2 - \left[\frac{39}{105+39}\right]^2 = 0.395$$

$$Gini\ impurity = 1 - \left[\frac{34}{125+34}\right]^2 - \left[\frac{125}{125+34}\right]^2 = 0.336$$

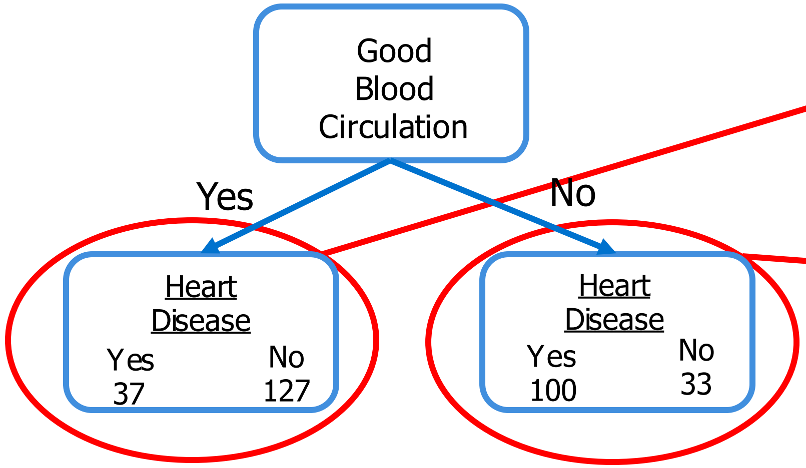
$$Gini\ for\ chest\ pain = \left[\frac{144}{144+159}\right] \times 0.395 + \left[\frac{159}{144+159}\right] \times 0.336 = 0.364$$

# Applications–Diagnosis of Heart Disease [1]

Do not stop here. Keep going for rest two splitting methods:

**Using equations on slide 5, we get:**

**# 2**



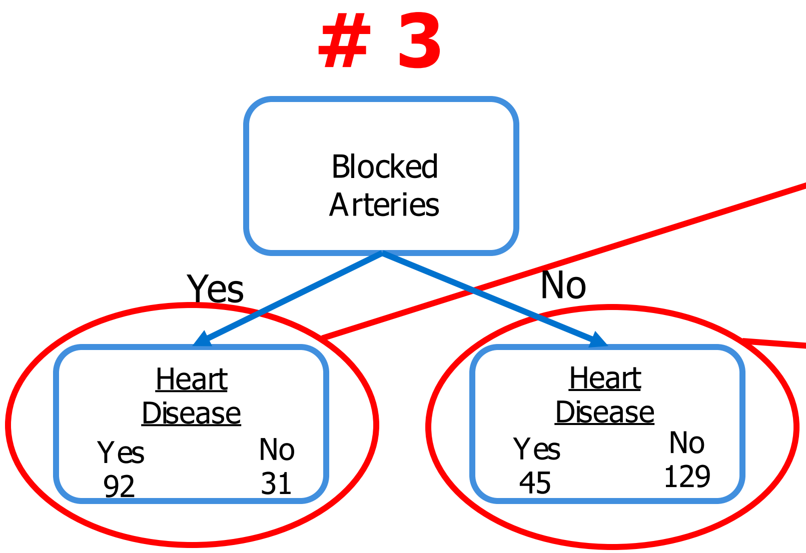

$$Gini\ impurity = 1 - \left[\frac{37}{127+37}\right]^2 - \left[\frac{127}{127+37}\right]^2 = 0.349$$

$$Gini\ impurity = 1 - \left[\frac{100}{100+33}\right]^2 - \left[\frac{100}{100+33}\right]^2 = 0.373$$

$$Gini\ for\ good\ blood\ circulation = \left[\frac{164}{164+133}\right] \times 0.349 + \left[\frac{133}{164+133}\right] \times 0.373 = 0.360$$

# Applications–Diagnosis of Heart Disease [1]

Do not stop here. Keep going for rest one splitting method:




**Using equations on slide 5, we get:**

$$Gini\ impurity = 1 - \left[\frac{92}{92+31}\right]^2 - \left[\frac{31}{92+31}\right]^2 = 0.377$$

$$Gini\ impurity = 1 - \left[\frac{45}{45+129}\right]^2 - \left[\frac{129}{45+129}\right]^2 = 0.384$$

$$Gini\ for\ blocked\ arteries = \left[\frac{123}{123+174}\right] \times 0.377 + \left[\frac{174}{123+174}\right] \times 0.384 = 0.381$$

# Applications–Diagnosis of Heart Disease [1]

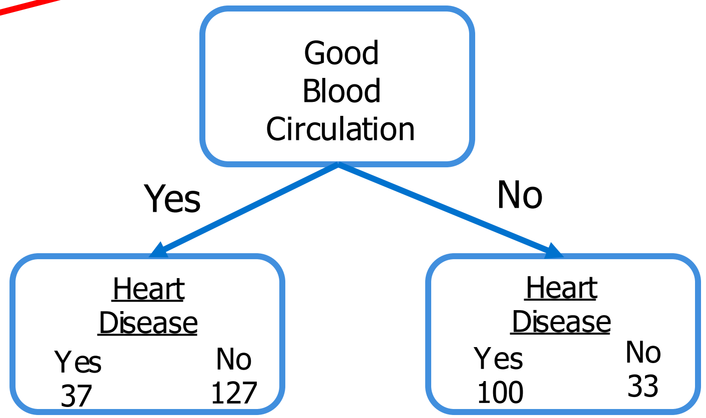
Great! Now we have "valid" evidence for splitting. Time to show decision tree what we got:

| | Gini Impurity |
|---|---|
| Chest pain | 0.364 |
| Good Blood Circulation | 0.360 |
| Blocked Arteries | 0.381 |

We got a winner!

**ROOT NODE:**

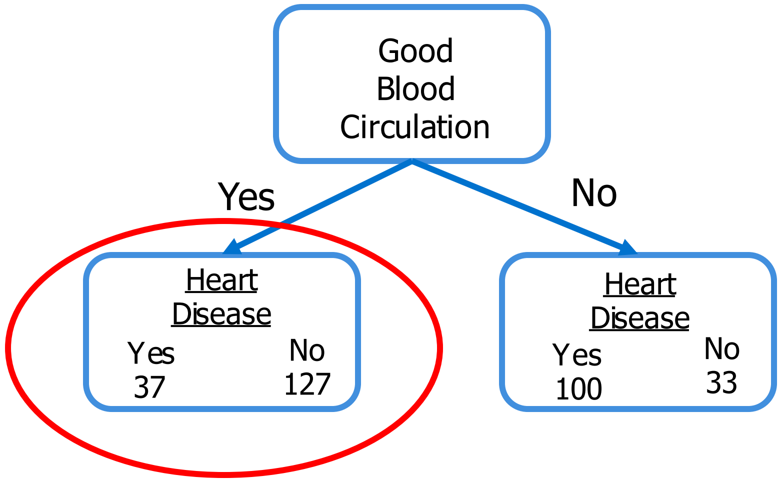
**# 2**



# Applications–Diagnosis of Heart Disease [1]

## Is that it?

**Of course not! We do the same for a child node of Good blood circulation now.** In the below image we will split **the left child** with a total of 164 sample.



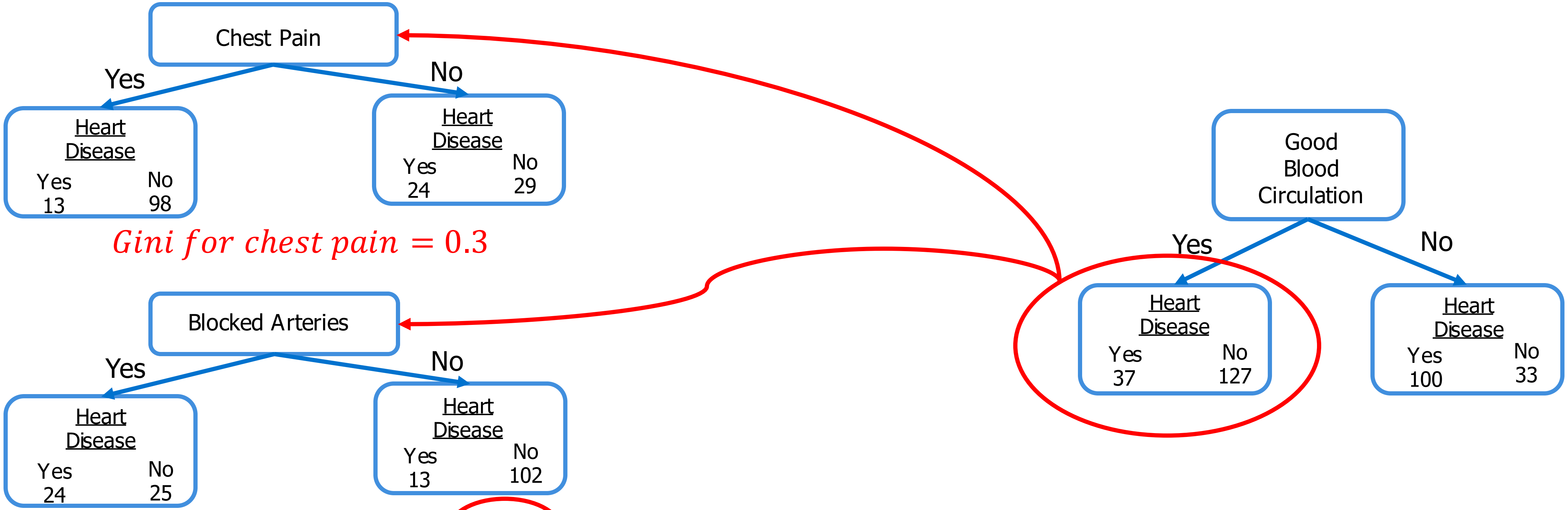

**Let's choose this one.**

**Split this node further.**

# Applications–Diagnosis of Heart Disease [1]

## Math is easy for JHUer!

We skip the tedious mathematical calculations and show the results directly:

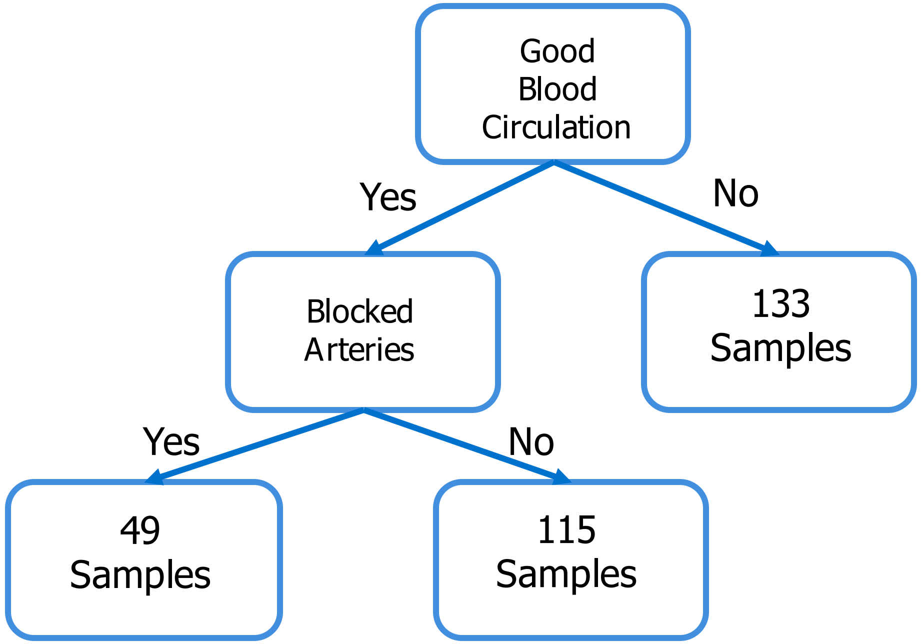



$Gini\ for\ chest\ pain = 0.3$

$Gini\ for\ blocked\ arteries = 0.29$ We got a winner! Split on blocked arteries

# Applications–Diagnosis of Heart Disease [1]

## Update the decision tree

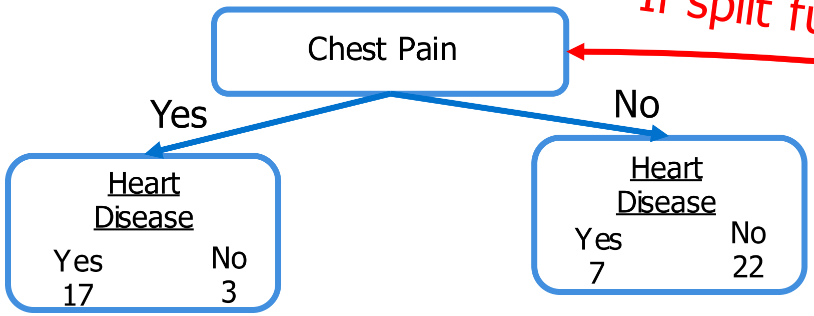


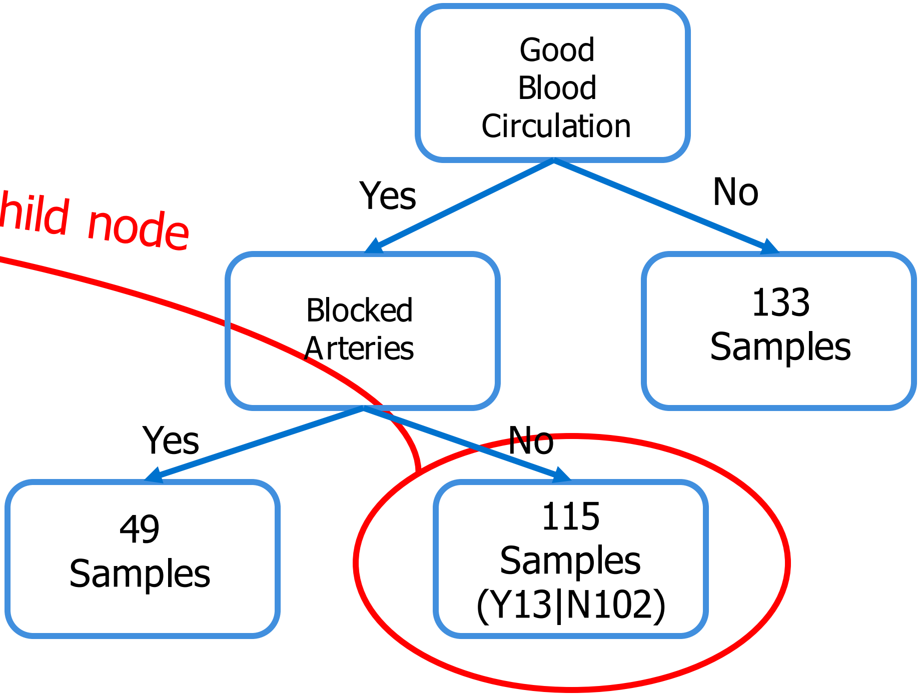
# Applications–Diagnosis of Heart Disease [1]

## How do we know if a node is worth splitting further?

**There should be a decrease in Gini Impurity.** Otherwise, do not split the child node further!

*If split further on the right child node*



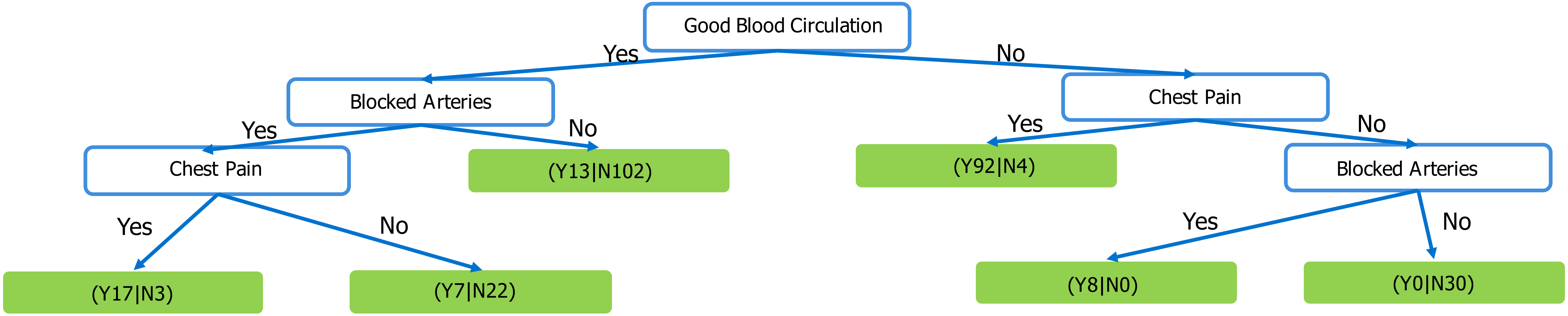

$$Gini\ Impurity\ for\ blocked\ arteries = 0.29$$

**There is no decrease in Gini Impurity for splitting the right child node. Just leave it as it is!**

$$Original\ Gini\ impurity = 1 - \left[\frac{13}{13+102}\right]^2 - \left[\frac{102}{13+102}\right]^2 = 0.2$$

# Applications–Diagnosis of Heart Disease [1]

## Keep going for other "worthy" nodes. Decision tree:



# References

# Reference


1. Analytics Vidhya. (2020, October 27). All About Decision Tree From Scratch with Python Implementation. Retrieved April 15, 2023, from https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/.
2. Towards Data Science. (n.d.). Gini Impurity Measure. Retrieved April 15, 2023, from https://towardsdatascience.com/gini-impurity-measure-dbd3878ead33.
3. Mandal, S. (2019, March 28). Understanding the Gini Index and Information Gain in Decision Trees. Analytics Steps. Retrieved April 15, 2023, from https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8.
4. Towards Data Science. (2019, February 28). Decision Trees Explained: Entropy, Information Gain, Gini Index, and CCP Pruning. Retrieved April 15, 2023, from https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c.

# Reference


6. Analytics Vidhya. (2020, June 3). 4 Ways to Split a Decision Tree. Retrieved April 15, 2023, from https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree/.
7. Analytics Vidhya. (2021, March 22). How to Select Best Split in Decision Trees using Chi-Square? Retrieved April 15, 2023, from https://www.analyticsvidhya.com/blog/2021/03/how-to-select-best-split-in-decision-trees-using-chi-square/.
8. Singh, M. K. (2020, October 29). CHAID Decision Tree. Retrieved April 15, 2023, from https://mksingh0892.medium.com/chaid-decision-tree-30a4c7ba6efc.
9. Goddumarri, Suryanarayana. (2021). Re: What is the disadvantage of using Information Gain for feature selection? Retrieved from https://www.researchgate.net/post/What-is-the-disadvantage-of-using-Information-Gain-for-feature-selection.
10. Sadeghi, Reza. (2019). What are the disadvantages of using Information Gain Ratio as a metric for building decision trees? Retrieved from https://www.quora.com/What-are-the-disadvantages-of-using-Information-Gain-Ratio-as-a-metric-for-building-decision-trees.

# Reference


11. Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.
12. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.
13. Subha Ganpati(Mar 16, 2021). Decision Trees and Splitting Functions (Gini, Information Gain and Variance Reduction). Retrieved from: https://medium.com/nerd-for-tech/decision-trees-and-splitting-functions-gini-information-gain-and-variance-reduction-23192f639048.
14. Srivastava, M. (2019, August 7). Guide to Decision Tree Algorithm. upGrad blog. Retrieved from https://www.upgrad.com/blog/guide-to-decision-tree-algorithm/.

# Thank You for Listening