Data Mining

INFS 4203/7203

Miao Xu

miao.xu@uq.edu.au

The University of Queensland, 2020 Semester 2

Last lecture

Classification

1. Input variables (or called features/attributes):
   - Missing values
   - Normalization
   - Curse of dimensionality
2. Output variable (or labels/supervised information)
   - Number of classes
3. Classification model/classifier
   - Generalization and overfitting
   - Cross-validation
   - Evaluation

Normalization and centralization

- Two ways to do normalization:

Max-Min normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Standardization:

$$x' = \frac{x - \mu}{\sigma}$$

- Data centralization: a centralized database is one where the information is collected, stored, and maintained in one location, but is accessible from many points.

One vs all

- **One vs. all**: learn one binary classifier for each class;
  - Horse/not horse (classifier 1), dog/not dog (classifier 2), cat/not cat (classifier 3),
  - Decide by confidence (a real value from the classifier)

Classifier 1: **not** horse with confidence 0.9

Classifier 2: dog with confidence 0.8

Classifier 3: cat with confidence 0.6

Decision: dog

Classifier 1: horse with confidence 0.6

Classifier 2: dog with confidence 0.6

Classifier 3: not cat with confidence 0.9

Decision: ???

One vs one

- **One vs. one**: learn one binary classifier for each pair of classes
  - Horse/dog (classifier 1), dog/cat (classifier 2), cat/horse (classifier 3)
  - Decide by vote

Classifier 1: dog

Classifier 2: dog

Classifier 3: cat

Decision: dog

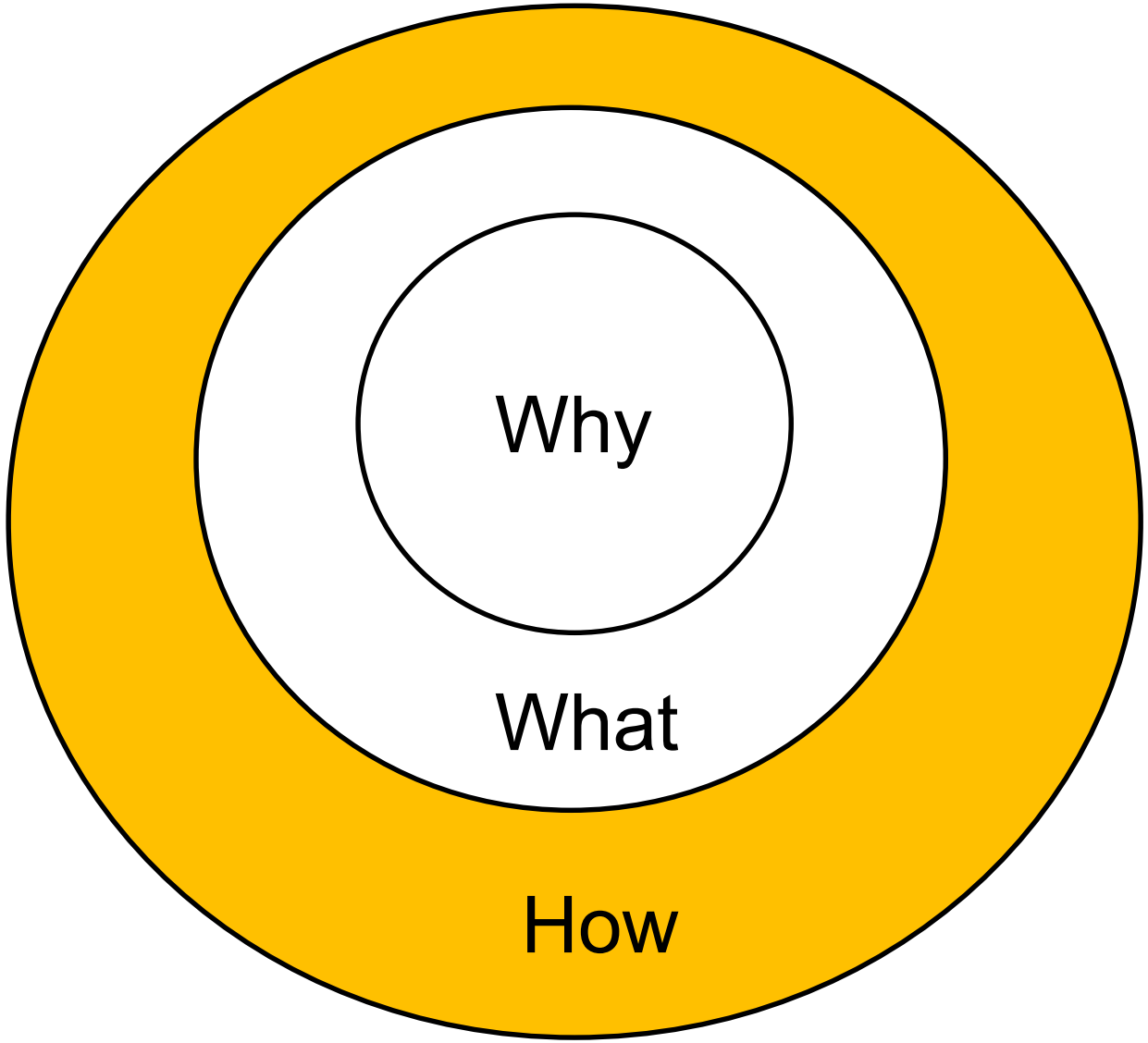
Classifier 1: horse

Classifier 2: dog

Classifier 3: cat

Decision: ???

Lecture 8: Classification 2-Decision Tree


Why
What
How

Outline

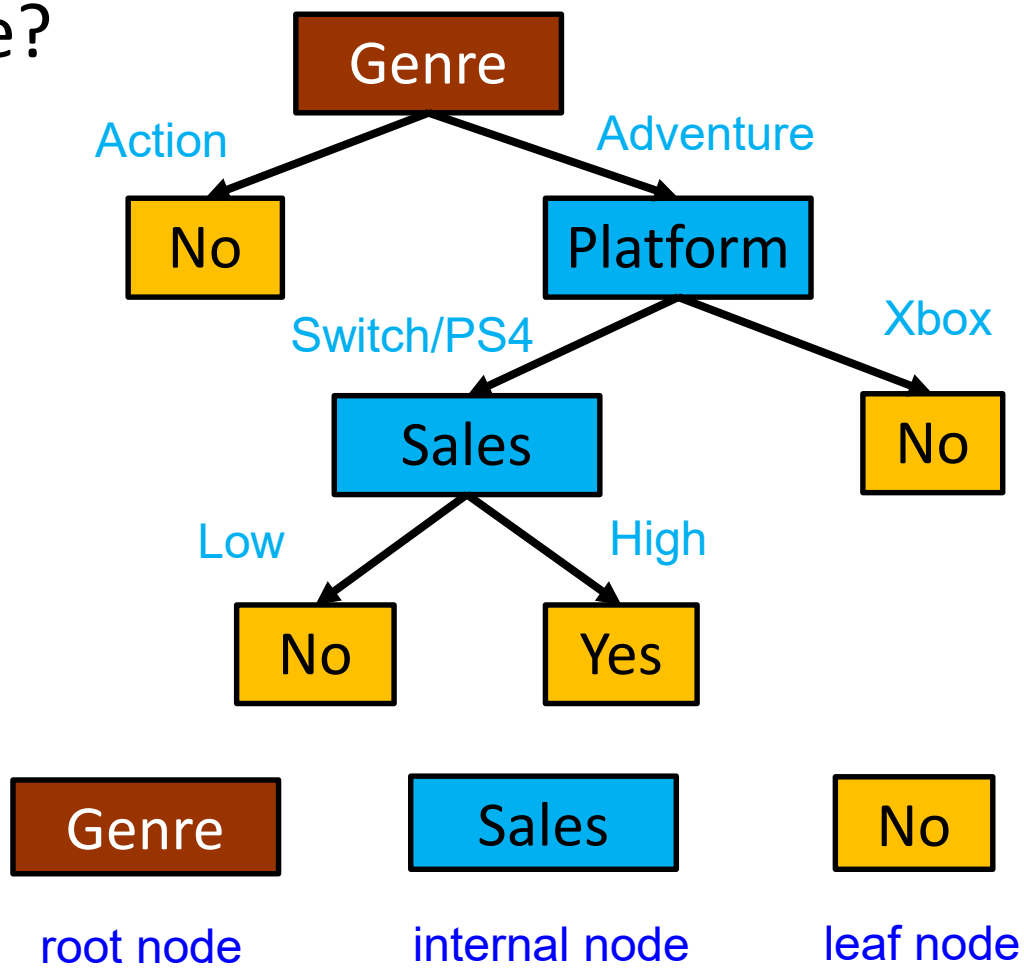
- General framework
- Splitting criteria
- Pruning
- Continuous values
- Random forest

Outline

- General framework
- Splitting criteria
- Pruning
- Continuous values
- Random forest

Decision Tree

- Make a decision based on a tree structure:
  - Q1: What is the genre of the game?
    - Action 
    - Adventure – go to Q2
  - Q2: What is the game platform
    - Xbox 
    - Switch/PS4– go to Q3
  - Q3: Is the sales number high?
    - No 
    - Yes 



Construction of a decision tree 1

For a current node:

- Input:
  - Training set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
  - Feature set $A = \{a_1, \dots, a_d\}$
- Process: Function **TreeGenerate**(D, A)

1. If **stop criteria** have reached: make the current node a **leaf** node and mark its class, return
2. **Select the optimal** splitting feature a_* from A
3. For all possible values of a_*, generate a branch (child node) and a subset $D_v \subset D$, such that $a_* = v$ for all samples in D_v
4. For all D_v
5. **TreeGenerate**$(D_v, A \backslash \{a_*\})$

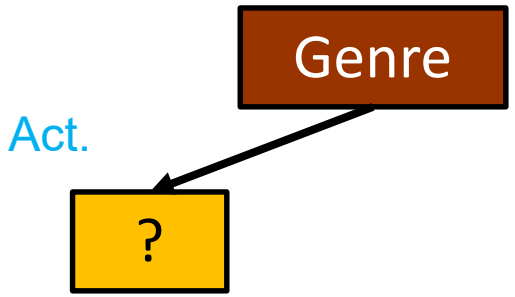
Construction of a decision tree 2

- A recursive procedure
- It stops when **stop criteria** have reached:
  1. All instances in the current node belong to **the same class** C
     - Mark the leaf node to the class C, return
  2. The **feature set A is empty**, **or** all instances in D have **the same values on A**
     - Mark the leaf node to the **majority** class in D, return
  3. D is **empty**
     - Mark the leaf node to the **majority** class of its **parent** node, return
- Construct a decision tree from the **root node**

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	**Action**	**Xbox**	**High**	**No**
2	**Action**	**Switch**	**Low**	**No**
3	**Action**	**PS4**	**High**	**No**
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

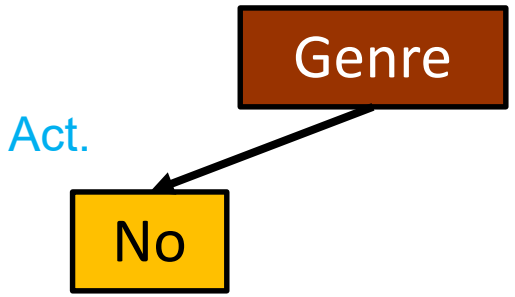


- D for **Genre=Act.**: {1, 2, 3}
- Criterion 1: All instances in the current node belong to the same class “No”
- Mark the leaf node to the class “No”, return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	**Action**	**Xbox**	**High**	**No**
2	**Action**	**Switch**	**Low**	**No**
3	**Action**	**PS4**	**High**	**No**
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

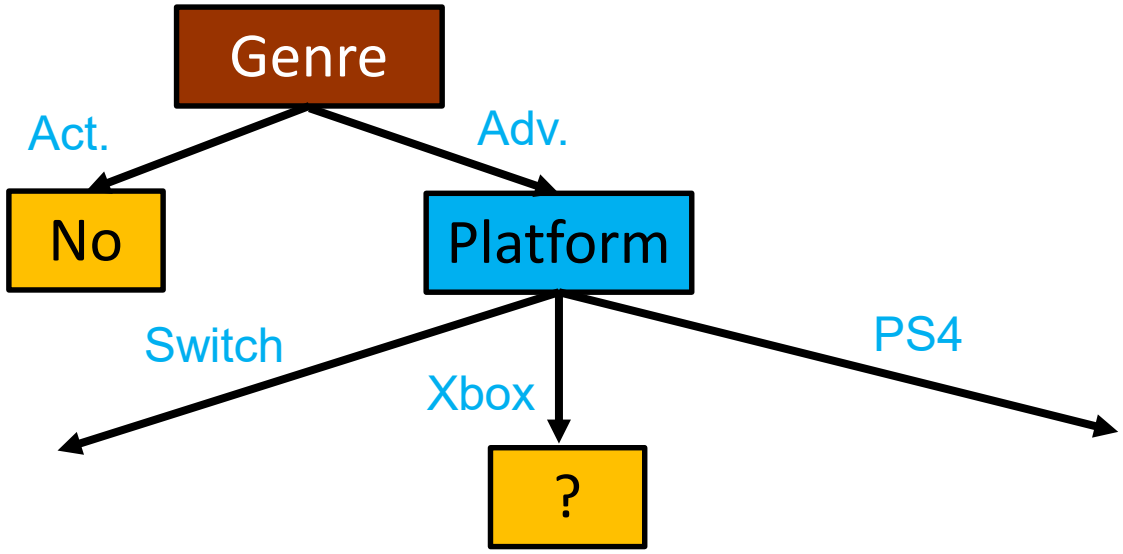


- D for **Genre=Act.**: {1, 2, 3}
- Criterion 1: All instances in the current node belong to the same class “No”
- Mark the leaf node to the class “No”, return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

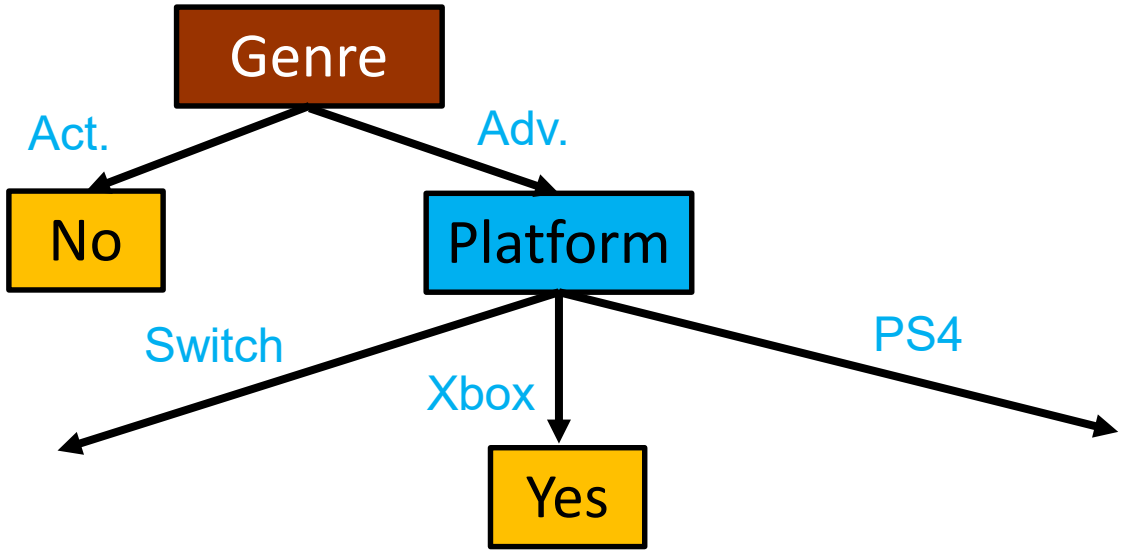


- Generate branches for all possible values of “Platform”
- Criterion 3: D for **Genre= Adv., Platform = Xbox**: empty
- D is empty: mark the leaf node to the majority class of its parent node (**Platform**), return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

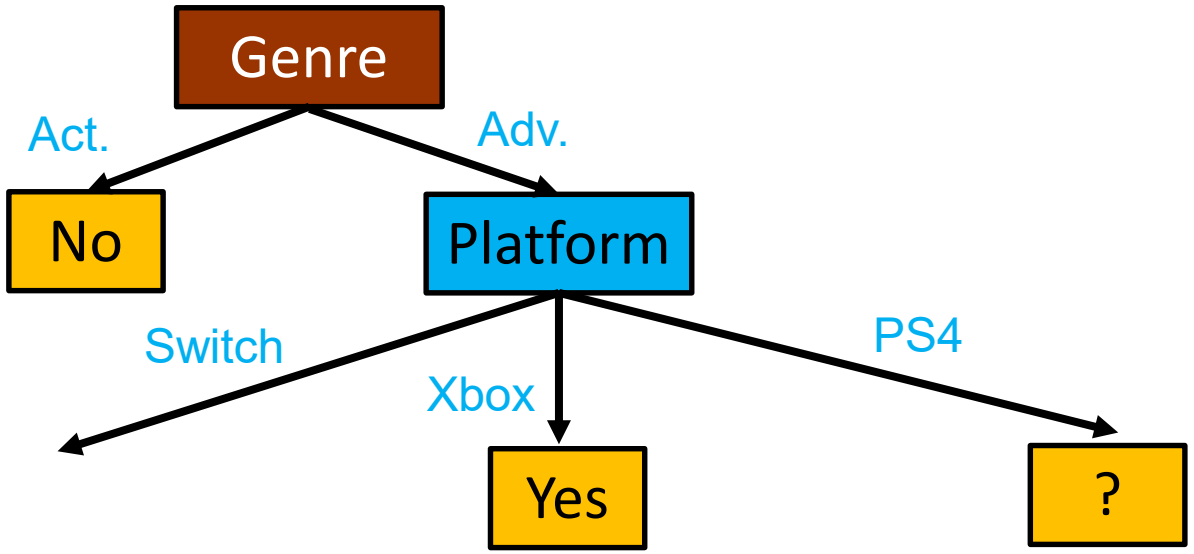


- Generate branches for all possible values of "Platform"
- Criterion 3: D for **Genre= Adv., Platform = Xbox**: empty
- D is empty: mark the leaf node to the majority class of its parent node (**Platform**), return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	**High**	**Yes**
9	Adventure	PS4	**High**	**Yes**
10	Adventure	PS4	**High**	**No**

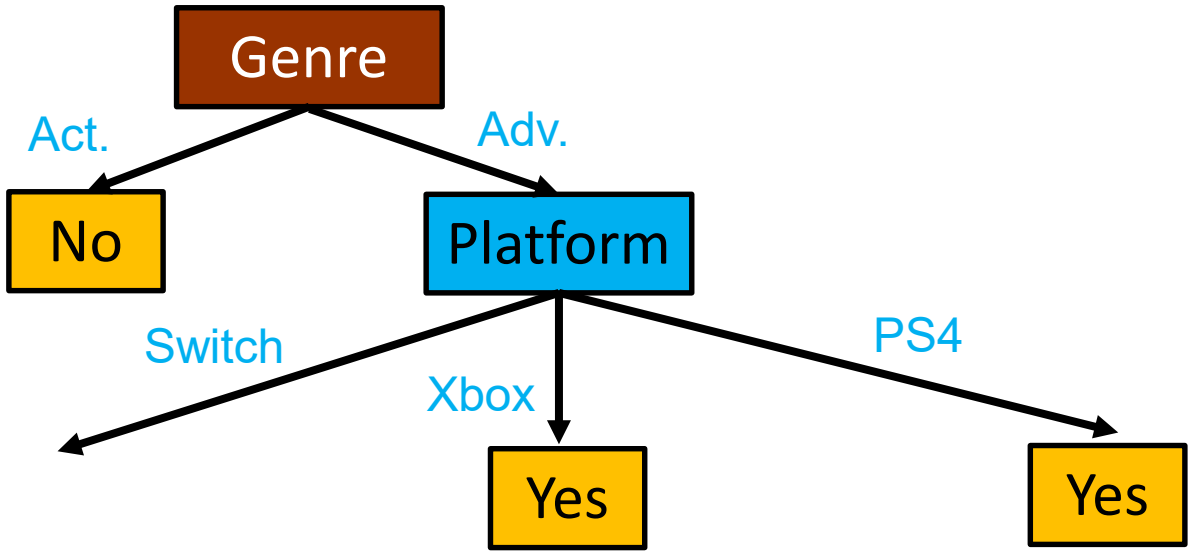


- D for **Genre= Adv., Platform = PS4**: {8, 9,10};
- A for **Genre= Adv., Platform = PS4** : {Sales}
- Criterion 2: All instances in D have the same values on A
- Mark the leaf node to the majority class in D, return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	**High**	**Yes**
9	Adventure	PS4	**High**	**Yes**
10	Adventure	PS4	**High**	**No**

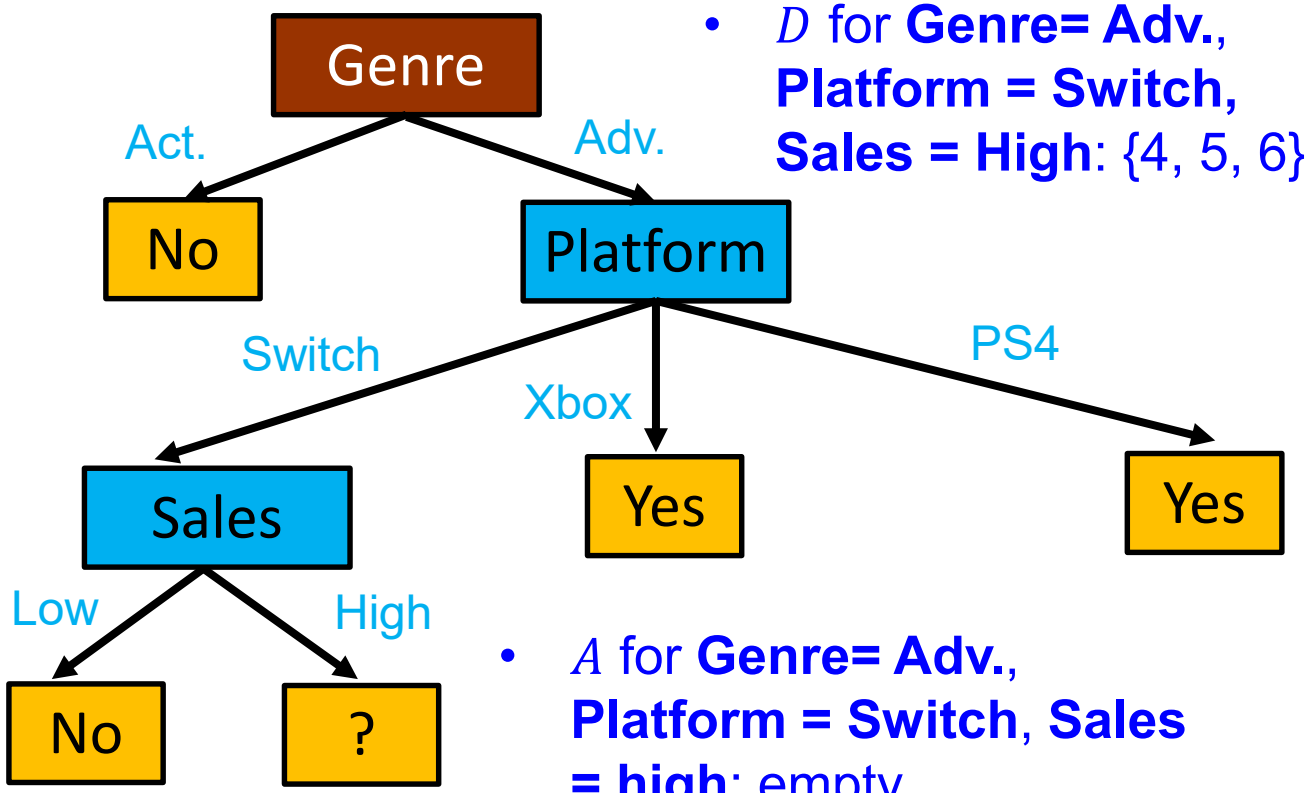


- D for **Genre= Adv., Platform = PS4**: {8, 9,10};
- A for **Genre= Adv., Platform = PS4** : {Sales}
- Criterion 2: All instances in D have the same values on A
- Mark the leaf node to the majority class in D, return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	**High**	**Yes**
5	Adventure	Switch	**High**	**Yes**
6	Adventure	Switch	**High**	**No**
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No



- D for **Genre= Adv., Platform = Switch, Sales = High**: {4, 5, 6}

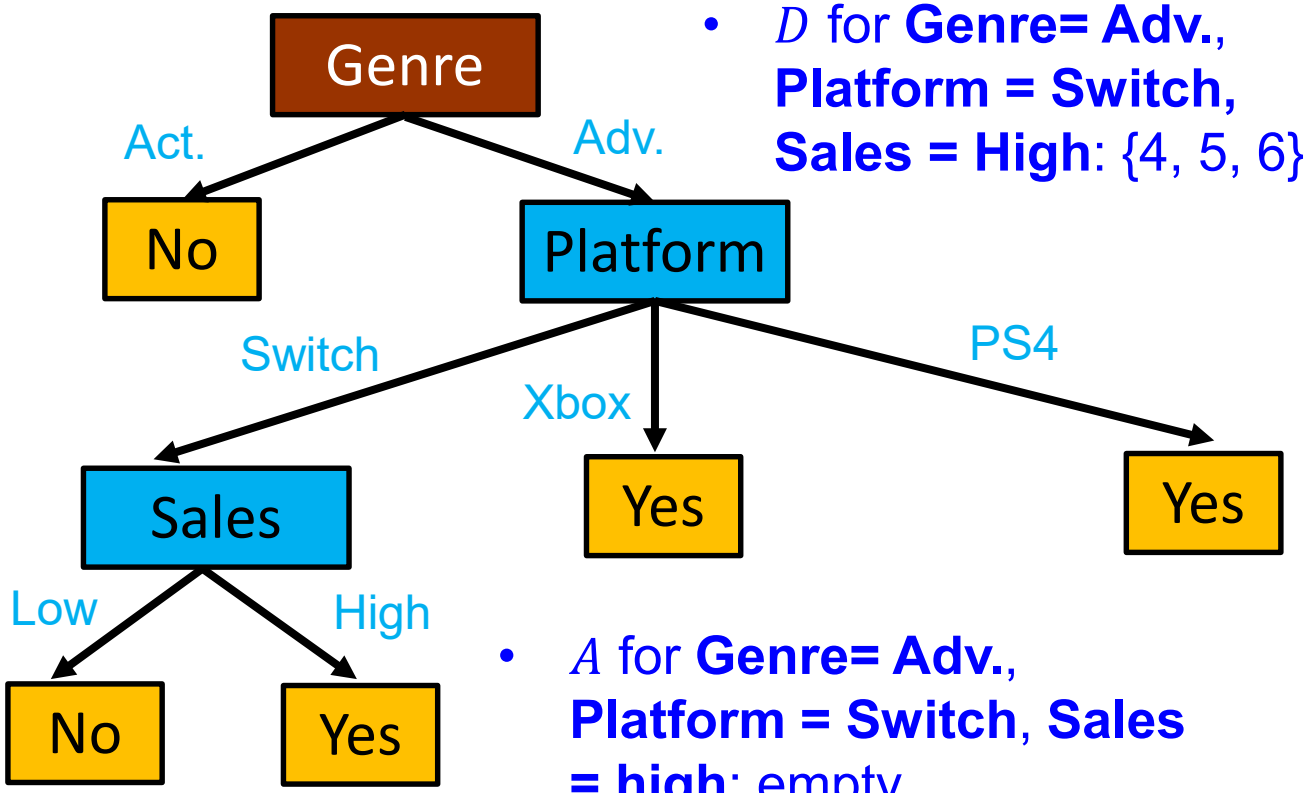
- A for **Genre= Adv., Platform = Switch**, **Sales = high**: empty

- Criterion 2: The feature set A is empty
- Mark the leaf node to the majority class in D, return

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	**High**	**Yes**
5	Adventure	Switch	**High**	**Yes**
6	Adventure	Switch	**High**	**No**
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No



- D for **Genre= Adv., Platform = Switch, Sales = High**: {4, 5, 6}
- A for **Genre= Adv., Platform = Switch**, **Sales = high**: empty

- Criterion 2: The feature set A is empty
- Mark the leaf node to the majority class in D, return

How to select the optimal splitting feature a_* from A?

Principle: as the splitting process proceeds, we want samples within each node to have increasing tendency to belong to the same class, i.e., increasing ***purity***.

Outline

- General framework
- Splitting Criteria (Purity)
- Pruning
- Continuous values
- Random forest

Three measurements of purity

- Information gain
- Gain ratio
- Gini index

Three measurements of purity

- Information gain
- Gain ratio
- Gini index

Information entropy: Ent(D)

- Assume there are m classes in total: $C_1, C_2, \ldots, C_m$
- p_i: the ratio of examples belonging to class C_i in dataset D
- Information entropy (if $p = 0$, $p\log_2 p = 0$)

$$\mathrm{Ent}(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

- The less Ent(D), the purer the node is
  - Minimum value: 0
  - Maximum value: $\log_2 m$

Information Gain: Gain(D, a)

- Assume for feature a, there are V possible values:
  - $\{v_1, v_2, \ldots, v_V\}$
- Make branches according to different values of a
- D_{v_j}: all samples in D with the a feature value equal to v_j ($|D|$ measures the number of elements in D)

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \text{Ent}(D_{v_j})$$

- Select the optimal splitting feature by the one having the **largest** information gain

ID3 algorithm [Quinlan, 1986]

Information gain: an example on Genre

- Information Entropy of the whole data set with *TID* 1-10:

$$\text{Ent}(\{1-10\}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9710$$

- For "Genre"
  - Information Entropy of Act.
  - (3 samples)

$$\text{Ent}(\{1-3\}) = -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3} = 0$$

  - Information Entropy of Adv.
  - (7 samples)

$$\text{Ent}(\{4-10\}) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.9852$$

  - Information Gain

$$\text{Gain}(\{1-10\}, \text{Genre}) = 0.9710 - \frac{3}{10}\times 0 - \frac{7}{10}\times 0.9852 = 0.2814$$

TID	Genre	Platform	Sales	Buy?
1	**Action**	Xbox	High	**No**
2	**Action**	Switch	Low	**No**
3	**Action**	PS4	High	**No**
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	**No**
7	Adventure	Switch	Low	**No**
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	**No**

Activity: information gain: an example on Platform

- Information Entropy of the whole data set with *TID* 1-10:

$$\text{Ent}(\{1-10\}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9710$$

- For "Platform"
  - Information Entropy of Xbox

$$\text{Ent}(\{1\}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

  - Information Entropy of Switch

$$\text{Ent}(\{2,4-7\}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.9710$$

  - Information Entropy of PS4

$$\text{Ent}(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

  - Information Gain

$$\text{Gain}(D,a) = \text{Ent}(D) - \sum_{j=1}^{V}\frac{|D_{v_j}|}{|D|}\text{Ent}(D_{v_j})$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Information gain: an example on Platform

- Information Entropy of the whole data set with *TID* 1-10:

$$\text{Ent}(\{1-10\}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9710$$

- For "Platform"
  - Information Entropy of Xbox

$$\text{Ent}(\{1\}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

  - Information Entropy of Switch

$$\text{Ent}(\{2,4-7\}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.9710$$

  - Information Entropy of PS4

$$\text{Ent}(\{3,8-10\}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$$

  - Information Gain

$$\text{Gain}(\{1-10\}, \text{Platform}) = 0.9710 - \frac{1}{10}\times 0 - \frac{5}{10}\times 0.9710 - \frac{4}{10}\times 1 = 0.0855$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Information gain: an example on Sales

- Information Entropy of the whole data set with *TID* 1-10:

$$\text{Ent}(\{1-10\}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9710$$

- For "Sales"
  - Information Entropy of High
  - (8 samples)

$$\text{Ent}(\{1, 3-6, 8-10\}) = -\frac{4}{8}\log_2\frac{4}{8} - \frac{4}{8}\log_2\frac{4}{8} = 1$$

  - Information Entropy of Low
  - (2 samples)

$$\text{Ent}(\{2, 7\}) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$$

  - Information Gain

$$\text{Gain}(\{1-10\}, \text{Sales}) = 0.9710 - \frac{2}{10}\times 0 - \frac{8}{10}\times 1 = 0.1710$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Information gain: selection

- Select the optimal splitting feature by the one having the **largest** information gain

$$\text{Gain}(\{1-10\}, \text{Genre}) = 0.9710 - \frac{3}{10} \times 0 - \frac{7}{10} \times 0.9852 = 0.2814$$

$$\text{Gain}(\{1-10\}, \text{Platform}) = 0.9710 - \frac{1}{10} \times 0 - \frac{5}{10} \times 0.9710 - \frac{4}{10} \times 1 = 0.0855$$

$$\text{Gain}(\{1-10\}, \text{Sales}) = 0.9710 - \frac{2}{10} \times 0 - \frac{8}{10} \times 1 = 0.1710$$

- Select "Genre" as the **first** splitting feature!
- How about next?

Information gain: second splitting on Platform

- Information entropy of 4-10

$$\text{Ent}(\{4-10\}) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.9852$$

- For "Platform"
  - Information Entropy of Switch

$$\text{Ent}(\{4-7\}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$$

  - Information Entropy of PS4

$$\text{Ent}(\{8-10\}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.9183$$

  - Information Gain

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

$$\text{Gain}(\{4-10\}, \text{Platform}) = 0.9852 - \frac{4}{7}\times 1 - \frac{3}{7}\times 0.9183 = 0.0202$$

Information gain: second splitting on Sales

- Information entropy of 4-10

$$\text{Ent}(\{4-10\}) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.9852$$

- For "Sales"
  - Information Entropy of Low

$$\text{Ent}(\{7\}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

  - Information Entropy of High

$$\text{Ent}(\{4-6, 8-10\}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.9183$$

  - Information Gain

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

$$\text{Gain}(\{4-10\}, \text{Sales}) = 0.9852 - \frac{1}{7}\times 0 - \frac{6}{7}\times 0.9183 = 0.1980$$

Select "Sales" as the second splitting feature!

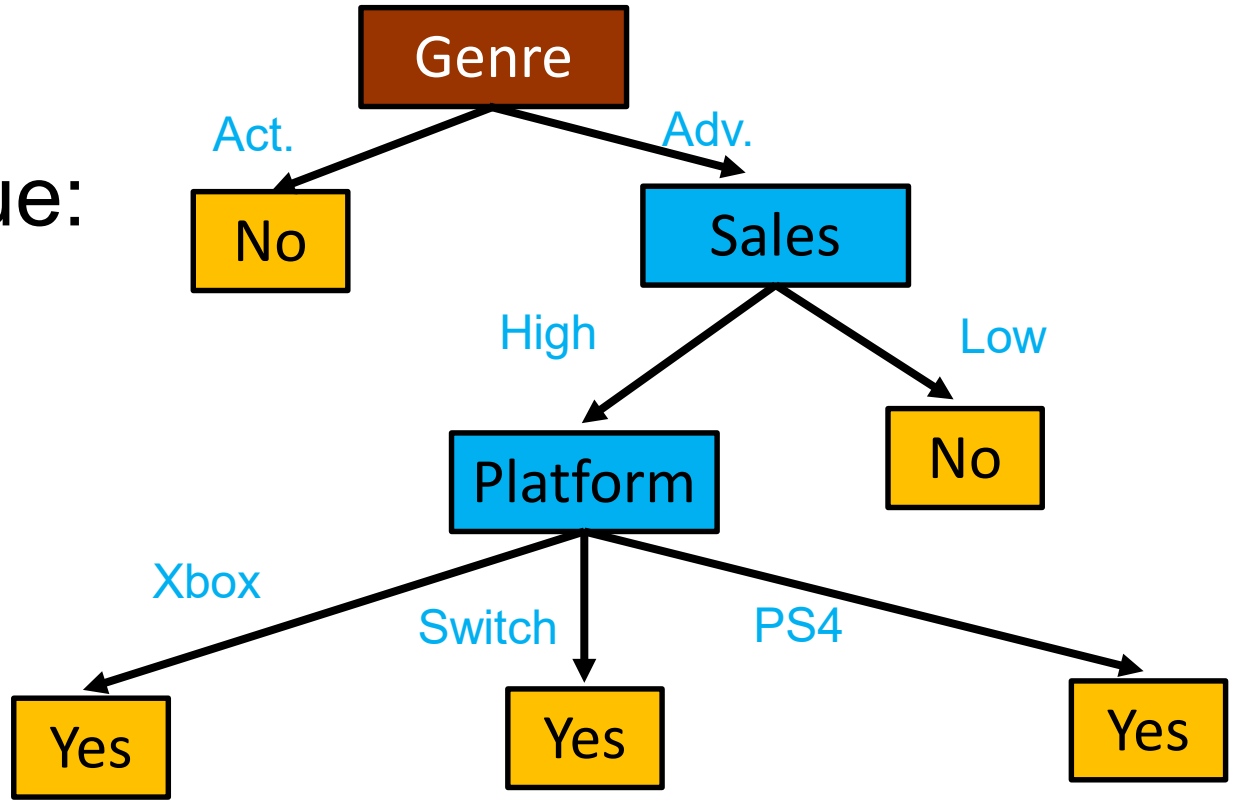
The constructed tree

- If Genre = Act., not buy
- If Genre = Adv., continue:
  - If Sales = Low, not buy
  - If Sales = High, continue:
    - If Platform = Xbox: buy
    - If Platform = Switch: buy
    - If Platform = PS4: buy

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

The constructed tree

- If Genre = Act., not buy
- If Genre = Adv., continue:
  - If Sales = Low, not buy
  - If Sales = High, continue:
    - If Platform = Xbox: buy
    - If Platform = Switch: buy
    - If Platform = PS4: buy



Three measurements of purity

- Information gain
- Gain ratio
- Gini index

- How about we use "*TID*" to calculate the information gain?

$$\text{Ent}(\{1\}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

$$\text{Gain}(\{1-10\}, \text{TID}) = 0.9710 - 10 \times \frac{1}{10} \times 0 = 0.9710$$

$$\text{Gain}(\{1-10\}, \text{Genre}) = 0.2814$$

$$\text{Gain}(\{1-10\}, \text{Platform}) = 0.0855$$

$$\text{Gain}(\{1-10\}, \text{Sales}) = 0.1710$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Should we choose *TID* as the splitting feature? 

No. It does not contribute to the generalization of the constructed decision tree.

Intrinsic value: IV(a)

- Assume for feature a, there are V possible values:
  - $\{v_1, v_2, \ldots, v_V\}$
- Make branches according to different values of a
- D_{v_j}: all samples in D with the a feature value equal to v_j ($|D|$ measures the number of elements in D)

$$\mathrm{IV}(a) = -\sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \log_2 \frac{|D_{v_j}|}{|D|}$$

Intrinsic value: example

- TID

$$\mathrm{IV}(TID) = -\sum_{1}^{10} \frac{1}{10} \log_2 \frac{1}{10} = 3.3219$$

- Genre

$$\mathrm{IV}(\mathrm{Genre}) = -\frac{3}{10} \log_2 \frac{3}{10} - \frac{7}{10} \log_2 \frac{7}{10} = 0.8813$$

- Sales

$$\mathrm{IV}(\mathrm{Sales}) = -\frac{2}{10} \log_2 \frac{2}{10} - \frac{8}{10} \log_2 \frac{8}{10} = 0.7219$$

- Platform

$$\mathrm{IV}(\mathrm{Platform}) = -\frac{1}{10} \log_2 \frac{1}{10} - \frac{4}{10} \log_2 \frac{4}{10} - \frac{5}{10} \log_2 \frac{5}{10} = 1.3610$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Gain ratio: $\text{Gain\_ratio}(D, a)$

$$\text{Gain\_ratio}(D, a) = \frac{\text{Gain}(D,a)}{\text{IV}(D,a)}$$

- Select the optimal splitting feature by the one having the **largest** gain ratio

- C4.5 algorithm [Quinlan, 1993]

  Does not use gain ratio directly.

  1. Select those features with information gain higher than the average
  2. Select the feature with the highest gain ratio among those higher than the average features

Gain ratio: Gain_ratio(D, a)

$$\text{Gain\_ratio}(D, a) = \frac{\text{Gain}(D,a)}{\text{IV}(D,a)}$$

$$\text{Gain}(\{1-10\}, \text{TID}) = 0.9710 - 10 \times \frac{1}{10} \times 0 = 0.9710$$

$$\text{Gain}(\{1-10\}, \text{Genre}) = 0.2814$$

$$\text{Gain}(\{1-10\}, \text{Platform}) = 0.0855$$

$$\text{Gain}(\{1-10\}, \text{Sales}) = 0.1710$$

$$\text{IV}(TID) = 3.3219$$

$$\text{IV}(\text{Genre}) = 0.8813$$

$$\text{IV}(\text{Platform}) = 1.3610$$

$$\text{IV}(\text{Sales}) = 0.7219$$

	TID	Genre	Platform	Sales
Gain_ratio	0.2923	**0.3193**	0.0628	0.2369

More about ID3 and C4.5*

- Developed by Australian computer scientist John Ross Quinlan (1943-)
- ID3 was developed in 1978 when Quinlan was doing a course assignment at Stanford University
- In 1993, Quinlan developed successor of ID3. However, since the names ID4, ID5 were already occupied, it is named as C4.0
- C4.5: “a slightly improved version of C4.0”

Three measurements of purity

- Information gain
- Gain ratio
- Gini index

Gini value: Gini(D)

- Assume there are m classes in total: $C_1, C_2, \ldots, C_m$
- p_i: the ratio of examples belonging to class C_i in dataset D
- Gini value for dataset D

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2$$

- Intuitive meaning: the likelihood of two samples randomly selected from D belonging to different classes
- The lower the Gini value is, the purer the dataset

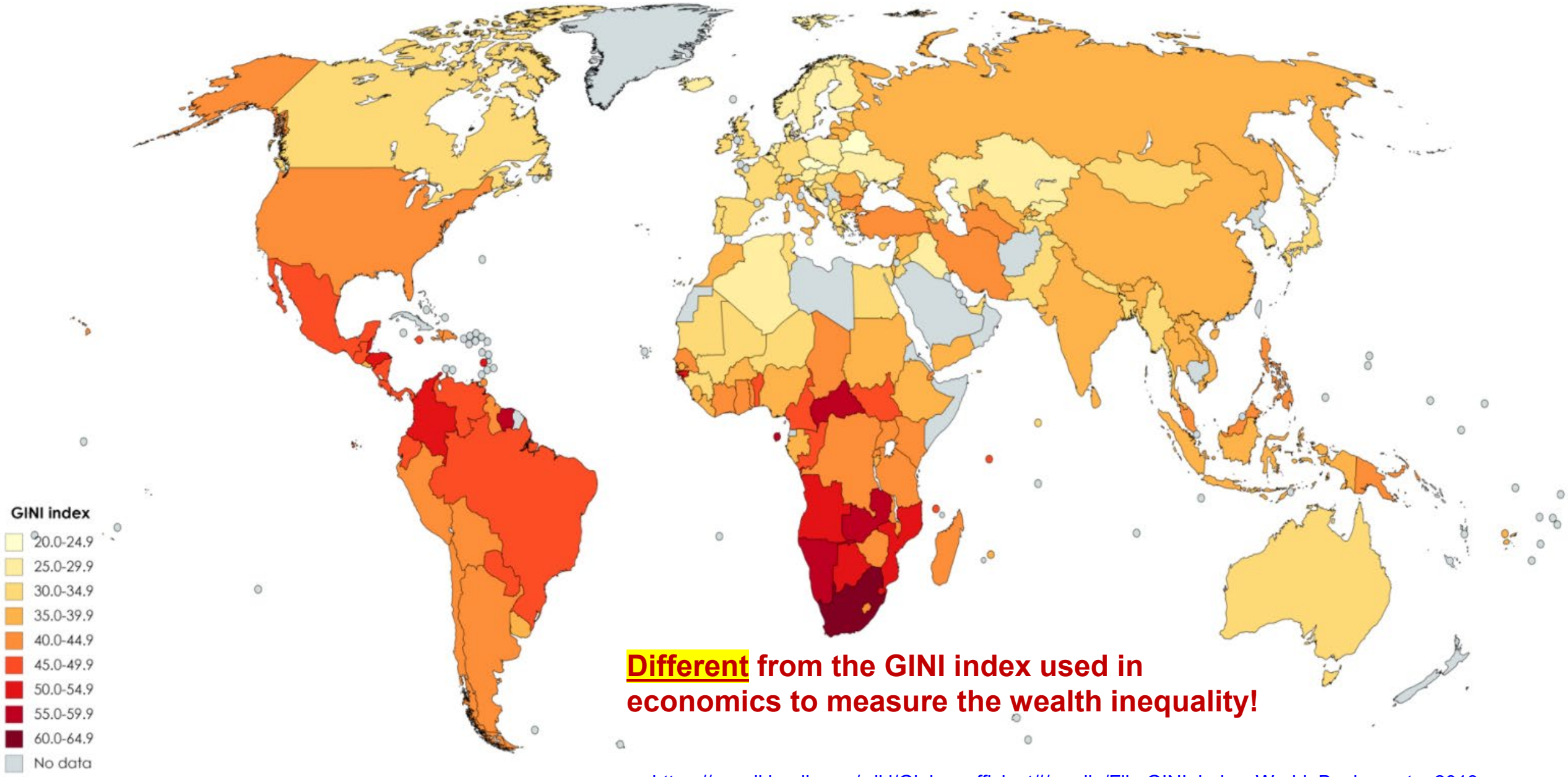
Gini index (or GINI index): Gini_index(D, a)

- Assume for feature a, there are V possible values:
  – $\{v_1, v_2, \ldots, v_V\}$
- Make branches according to different values of a
- D_{v_j}: all samples in D with the a feature value equal to v_j ($|D|$ measures the number of elements in D)

$$\text{Gini\_index}(D, a) = \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \text{Gini}(D_{v_j})$$

- Select the optimal splitting feature by the one having the **lowest** Gini index

CART algorithm [Breiman et al., 1984]



Different from the GINI index used in economics to measure the wealth inequality!

https://en.wikipedia.org/wiki/Gini_coefficient#/media/File:GINI_index_World_Bank_up_to_2018.png

Gini index: example

- Gini index for Genre
  - Gini Value of Act.

$$\text{Gini}(\{1,2,3\}) = 1 - 1^2 - 0^2 = 0$$

  - Gini Value of Adv.

$$\text{Gini}(\{4-10\}) = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.4898$$

  - Gini Index

$$\text{Gini\_index}(\{1-10\}, \text{Genre}) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.4898 = 0.3429$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Activity: Gini index: example

- Gini index for Platform
  - Gini Value of Xbox

$$\text{Gini}(\{1\}) = 1 - 1^2 - 0^2 = 0$$

  - Gini Value of Switch

$$\text{Gini}(\{2, 4-7\}) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$$

  - Gini Value of PS4

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2$$

  - Gini Index

$$\text{Gini\_index}(D, a) = \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \text{Gini}(D_{v_j})$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Gini index: example

- Gini index for Platform
  - Gini Value of Xbox

$$\text{Gini}(\{1\}) = 1 - 1^2 - 0^2 = 0$$

  - Gini Value of Switch

$$\text{Gini}(\{2, 4-7\}) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$$

  - Gini Value of PS4

$$\text{Gini}(\{3, 8-10\}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

  - Gini Index

$$\text{Gini\_index}(\{1-10\}, \text{Platform}) = \frac{1}{10} \times 0 + \frac{5}{10} \times 0.48 + \frac{4}{10} \times 0.5 = 0.44$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Gini index: example

- Gini index for Sales
  - Gini Value of High

$$\text{Gini}(\{1, 3-6, 8-10\}) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = 0.5$$

  - Gini Value of Low

$$\text{Gini}(\{2, 7\}) = 1 - 1^2 - 0^2 = 0$$

  - Gini Index

$$\text{Gini\_index}(\{1-10\}, \text{Sales}) = \frac{2}{10} \times 0 + \frac{8}{10} \times 0.5 = 0.4$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Gini index: selection

- Select the optimal splitting feature by the one having the **lowest** Gini index

$$\text{Gini\_index}(\{1-10\}, \text{Genre}) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.4898 = 0.3429$$

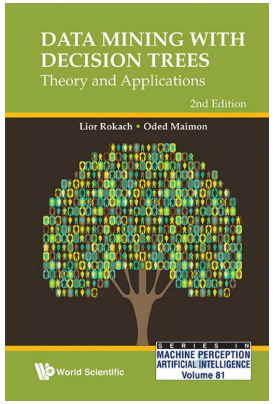
$$\text{Gini\_index}(\{1-10\}, \text{Platform}) = \frac{1}{10} \times 0 + \frac{5}{10} \times 0.48 + \frac{4}{10} \times 0.5 = 0.44$$

$$\text{Gini\_index}(\{1-10\}, \text{Sales}) = \frac{2}{10} \times 0 + \frac{8}{10} \times 0.5 = 0.4$$

- Select "Genre" as the **first** splitting feature!

There are many other selection measures!

See Chapter 5 of *Data Mining with Decision Trees: Theory and Applications 2nd Edition*



Three measurements of purity

- Information gain (select the largest, ID3)

$$\mathrm{Ent}(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad \mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \mathrm{Ent}(D_{v_j})$$

- Gain ratio (select the largest, or as the heuristics in C4.5)

$$\mathrm{IV}(a) = -\sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \log_2 \frac{|D_{v_j}|}{|D|} \qquad \mathrm{Gain\_ratio}(D, a) = \frac{\mathrm{Gain}(D,a)}{\mathrm{IV}(D,a)}$$

- Gini index (select the lowest, CART)

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2 \qquad \mathrm{Gini\_index}(D, a) = \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \mathrm{Gini}(D_{v_j})$$

Outline

- General framework
- Splitting Criteria
- Pruning
- Continuous values
- Random forest

- Overfitting: the phenomenon that a **model**, which has learned some **special characteristics** of the training data, does well on the training samples but not well on unseen new samples, resulting in reduced generalization.
- In decision tree, if we have too many branches, the learner may be misled by “special characteristics” of the training data
- Pruning: prune some of the branches in decision tree to reduce the risk of overfitting

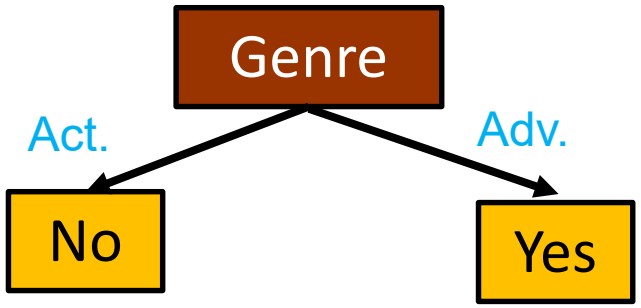
Two pruning approaches

- Pre-pruning (in the construction)
  - Halt tree construction early: do not split a node if this would result in poorer generalization
- Post-pruning (after the construction)
  - Remove branches from a “fully grown” tree, if this would result in better generalization

We use **test data** or **cross validation** to measure the generalization performance.

Pre-pruning: example 1

- No branch for Genre
  - All instances classified as “No” (Not Buy)
  - Test accuracy: 67%
- Branch for Genre



  - Test accuracy: 100%

Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	No
6	Adventure	Switch	High	No
7	Adventure	Xbox	Low	Yes
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

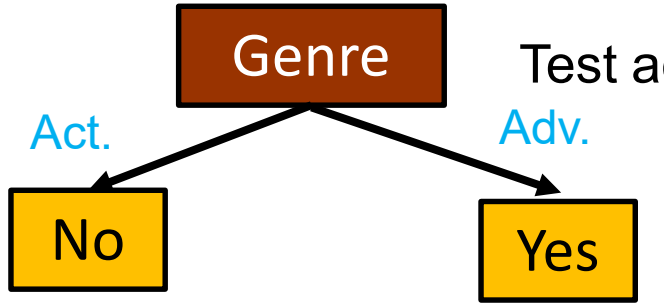
Test data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Adventure	Switch	Low	Yes
3	Action	PS4	High	No

Splitting! (No pre-pruning)

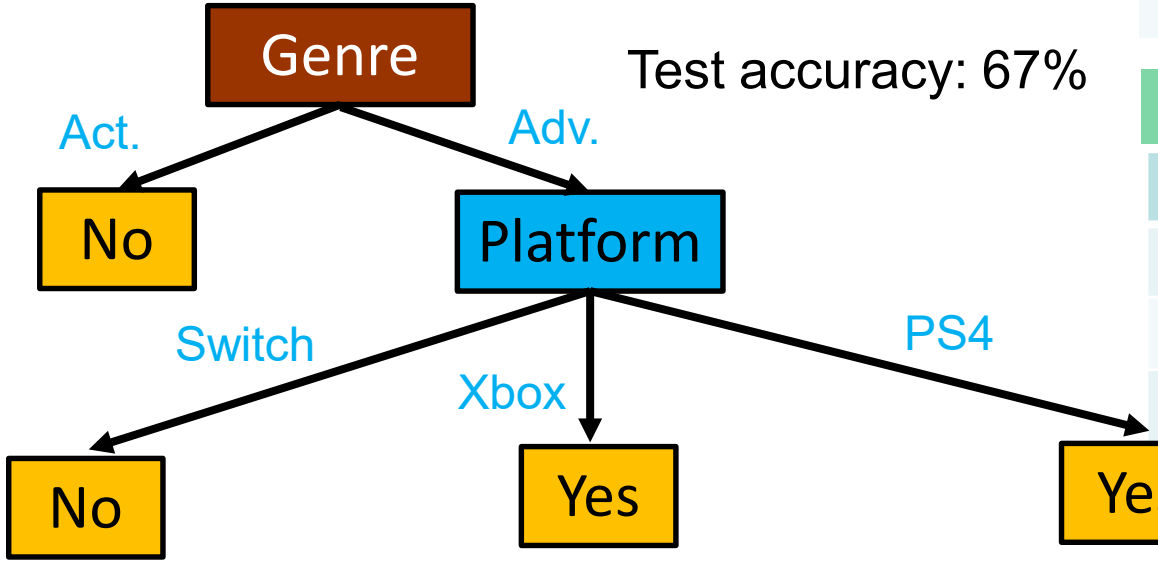
Pre-pruning: example 2

- No branch for Platform



Test accuracy: 100%

- Branch for Platform



Test accuracy: 67%

Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	No
6	Adventure	Switch	High	No
7	Adventure	Xbox	Low	Yes
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Test data

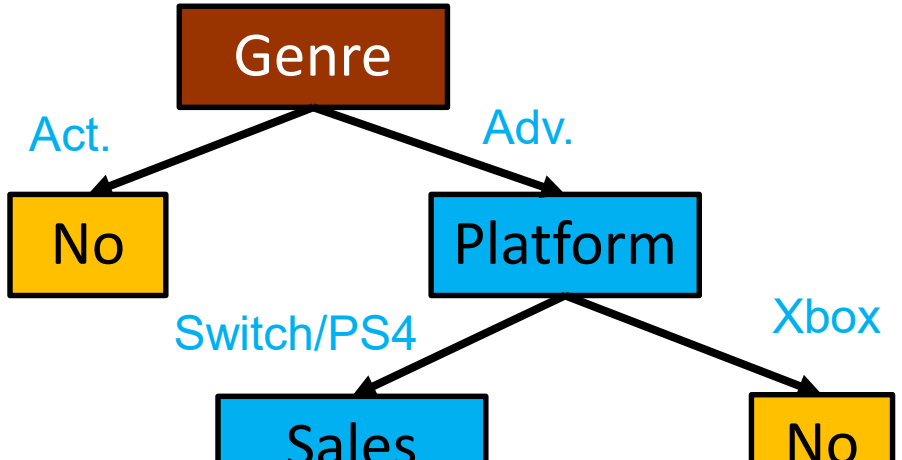
TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Adventure	Switch	Low	Yes
3	Action	PS4	High	No

Not splitting! (Pre-Pruning)

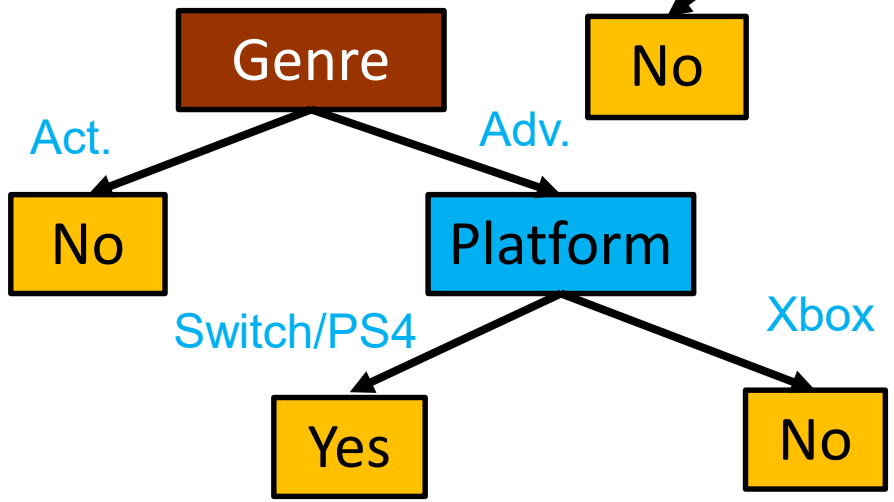
Post-pruning: example

- Fully grown

Test accuracy: 67%



- If pruning Sales



Test accuracy: 100%

Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No

Test data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Adventure	Switch	Low	Yes
3	Action	PS4	High	No

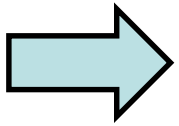
Post-pruning "Sales"

Outline

- General framework
- Splitting Criteria
- Pruning
- Continuous values
- Random forest

Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No



Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No

Bi-partition

- Assume for continuous valued feature a in data set D, there are V values observed:
  - $\{v_1, v_2, \ldots, v_V\}$
- With a slit value v_t, $D = D_t^- \cup D_t^+$
  - D_t^-: samples in D with a's value not greater than v_t
  - D_t^+: samples in D with a's value greater than v_t
- Choose the optimal v_t among **candidate split values**

$$\left\{ \frac{v_j + v_{j+1}}{2} \middle| 1 \le j \le V - 1 \right\}$$

- The continuous feature can be used as a splitting feature **more than once** in a decision tree

Bi-partition: example

- Sort the values and determine the candidate split values:

	0.3	0.4	0.6	0.8	0.9	1.0	1.1	1.2	1.3	1.4
Candidate split values	0.35	0.5	0.7	0.85	0.95	1.05	1.15	1.25	1.35	

Sorting

- If partition data by 0.5M:
  - D_t^-:{2,7}

$$\text{Gini}(\{2,7\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

  - D_t^+:{1, 3-6, 8-10}

$$\text{Gini}(\{1, 3-6, 8-10\}) = 1 - \left(\frac{4}{8}\right)^2 - \left(\frac{4}{8}\right)^2 = 0.5$$

  - Gini index

$$\text{Gini\_index}(\{1-10\}, \text{Sales}, 0.5M) = \frac{2}{10} \times 0 + \frac{8}{10} \times 0.5 = 0.4$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No

	0.3	0.4	0.6	0.8	0.9	1.0	1.1	1.2	1.3	1.4
	0.35	**0.5**	**0.7**	**0.85**	**0.95**	**1.05**	**1.15**	**1.25**	**1.35**	
Gini index:	0.4444	0.4000	0.3429	0.2667	0.4000	0.4667	0.4190	0.4750	0.4444	

$$\text{Gini\_index}(\{1-10\}, \text{Genre}) = 0.3429$$

$$\text{Gini\_index}(\{1-10\}, \text{Platform}) = 0.44$$

- Select "Sales = 0.85" as the **first** splitting feature and value!

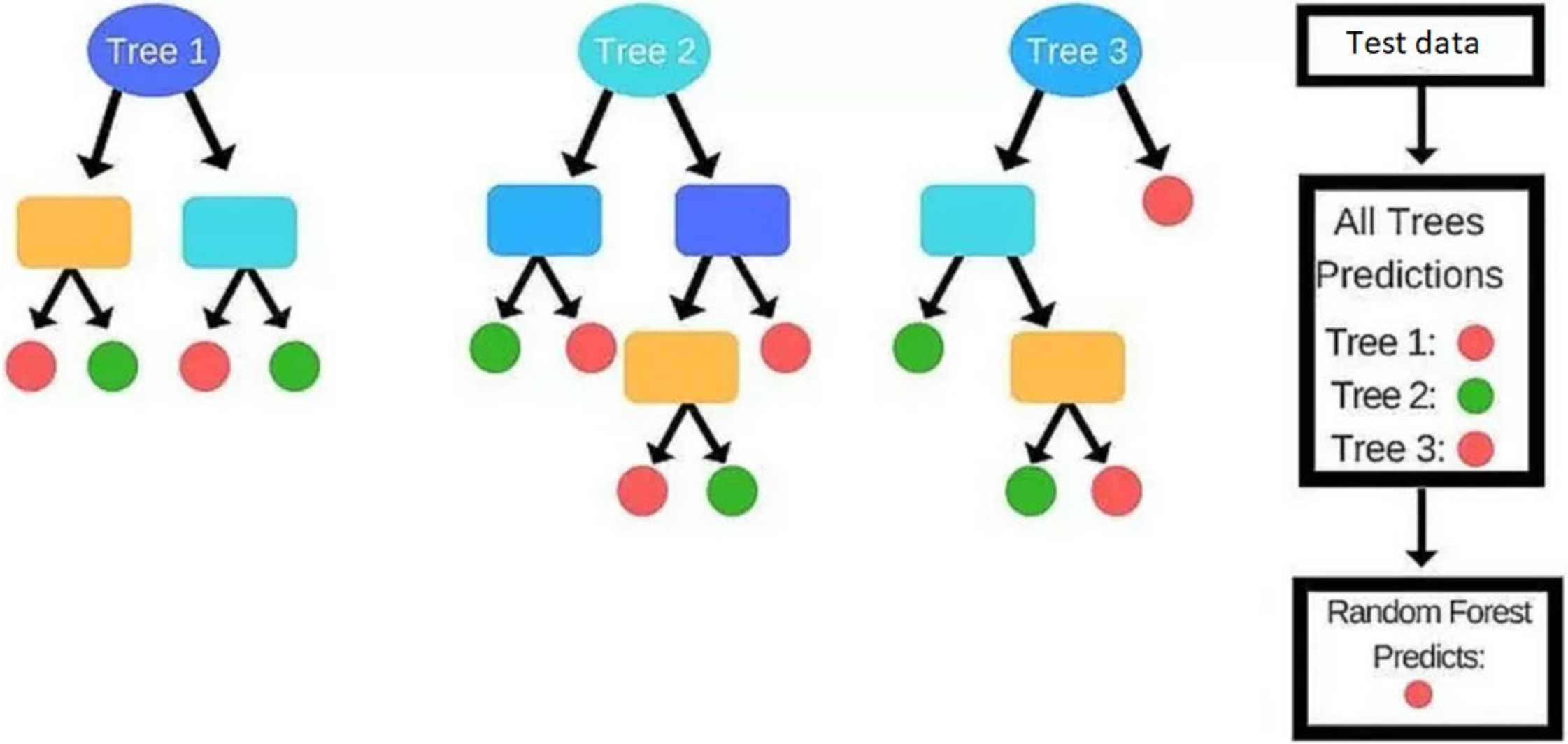
TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No

Outline

- General framework
- Splitting Criteria
- Pruning
- Continuous values
- Random forest

Random forest

- Construct several decision trees instead of only one decision tree
- Decision trees are constructed by
  - Determine *k*, which is a positive integer
  - Randomly sample *k* features among all *d* features
  - Construct a decision tree based on the sampled *k* features of all data
- Combine the output of all the trees together
  - E.g. majority voting
- Ensemble learning method: predict with multiple classifiers



https://dataaspirant.com/random-forest-algorithm-machine-learing/

Other methods based on decision tree

- Bagging: an ensemble method based on decision trees
  (the same proposer for CART and Random forest)

- Adaboost: an ensemble method sequentially learning decision stumps [The 2003 Gödel Prize]

- Isolation forest: using decision trees to do (unsupervised) anomaly detection [Best Paper Runner-Up Award in ICDM 2008]

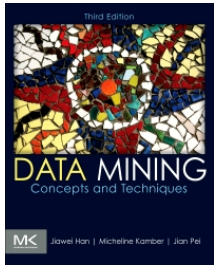
Summary

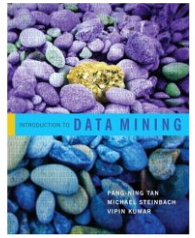
- Decision tree algorithm
  - Stopping criteria
  - Splitting criteria: Information Gain, Gain ratio, GINI index
  - Pruning: pre- and post-
  - Continuous values

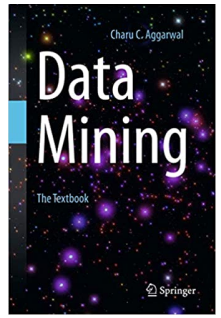
- Random forest algorithm
  - An ensemble method based on decision trees

Recommended reading

- [Han et al., 2012]
  - Chapter 6.3
- [Tan et al., 2005]
  - Chapter 4.3
- [Aggarwal, 2015]
  - Chapter 10.3, 11.8







- All other references (links) within the slides

Next week

- k-Nearest Neighbour
- Naïve Bayesian
- …