Chapter 8. Classification: Concepts and Decision Tree Model

Meng Jiang

Data Science

Supervised vs. Unsupervised Learning

- Supervised learning (**classification**)
  - Supervision: The training data instances and their attributes/features are accompanied by labels indicating the class of the instances.
  - **Predict labels** for testing data instances.
- Unsupervised learning (**clustering**)
  - The class **labels** of training data is **unknown**
  - Given a set of attributes, with the aim of establishing the existence of classes or clusters.

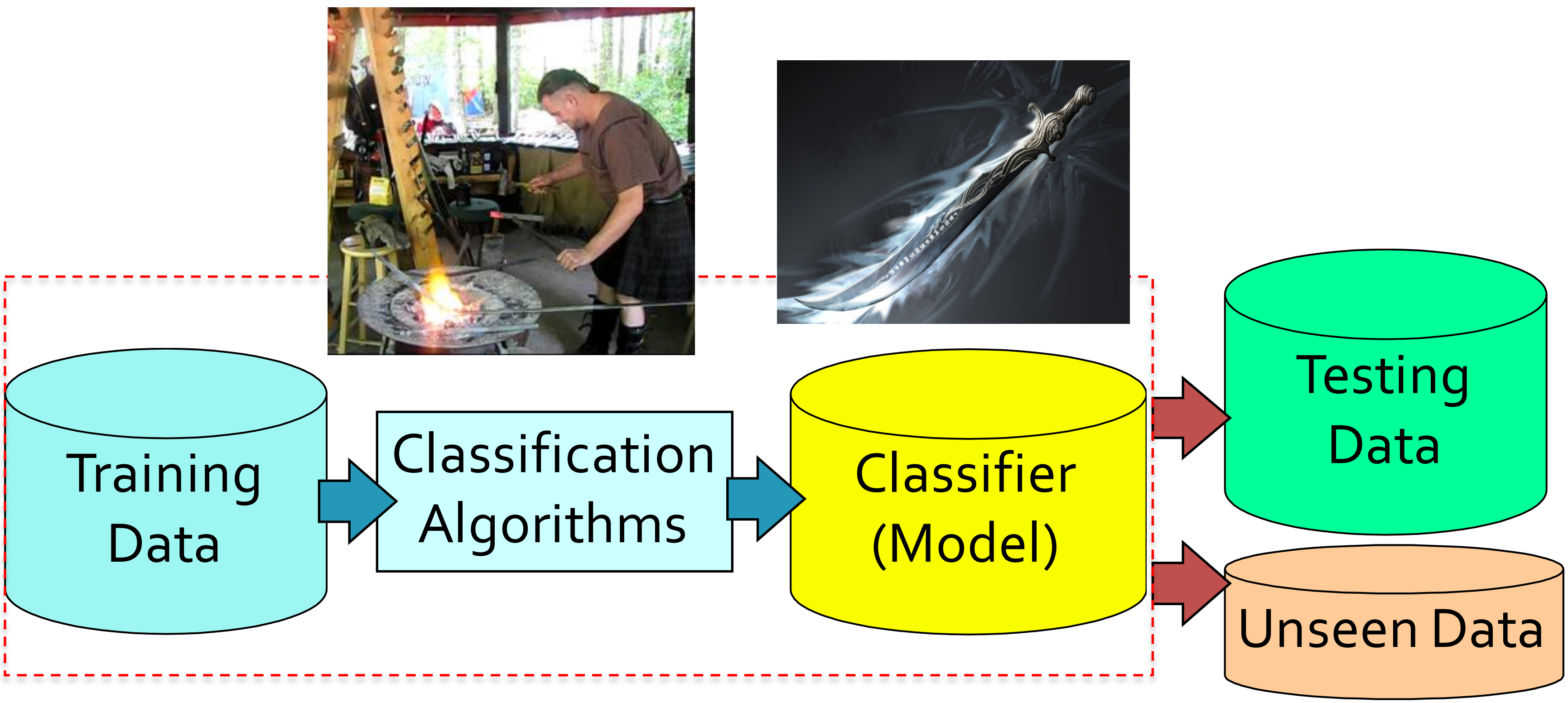
Machine learning types	Data mining tasks
Supervised learning	**Classification** Regression …
Unsupervised learning	Clustering Pattern/Association mining …

Classification: Applications

- Credit/loan approval: Yes or No
- Medical diagnosis: if a tumor is cancerous or benign
- Fraud detection: if a transaction is fraudulent
- Web page categorization: which category it is
- …

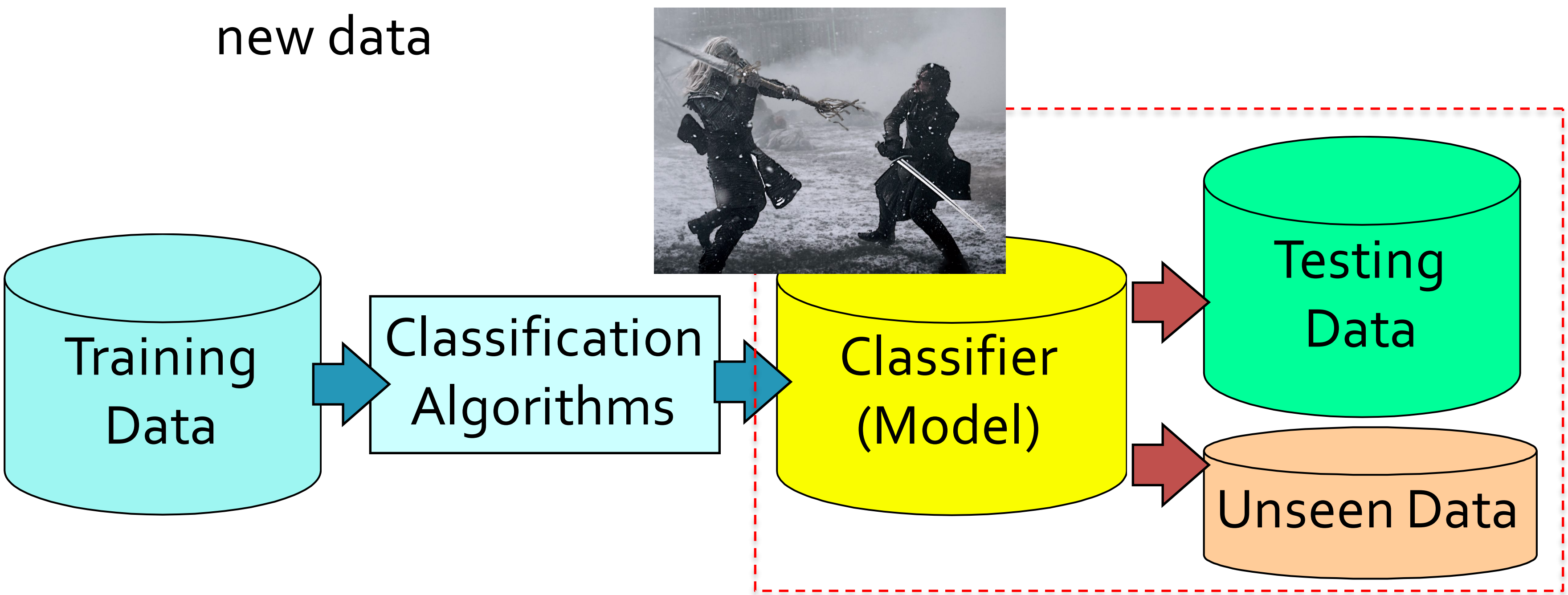
Classification: A Two-Step Process

- (1) Model construction
  - Models: Decision trees, Naïve Bayes, SVM, Neural Networks, etc.



Classification: A Two-Step Process

- (2) Model usage
  - Estimate accuracy of the model
    - Accuracy: % of test instances that are correctly classified
    - Test set is independent of training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to classify new data



Today: Decision Tree for Classification

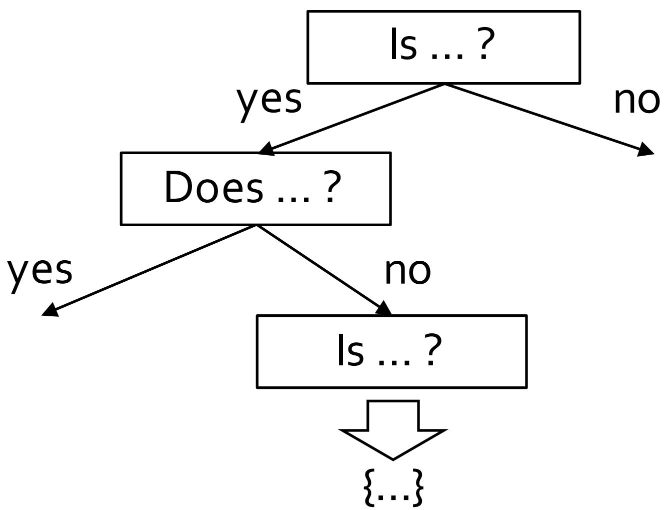
- Describe the difference between classification and clustering
- Describe two steps of the classification process

- Describe what is **entropy**; describe and compare the following "**feature selection measures**" or called "**splitting criteria**": **information gain**, **gain ratio**, and **gini index**.
- Given training instances and their attributes, construct by hand and implement using Python **Decision Tree models**:
  - ID3: information gain
  - C4.5: gain ratio
  - CART: gini index

Let’s Play a Game!

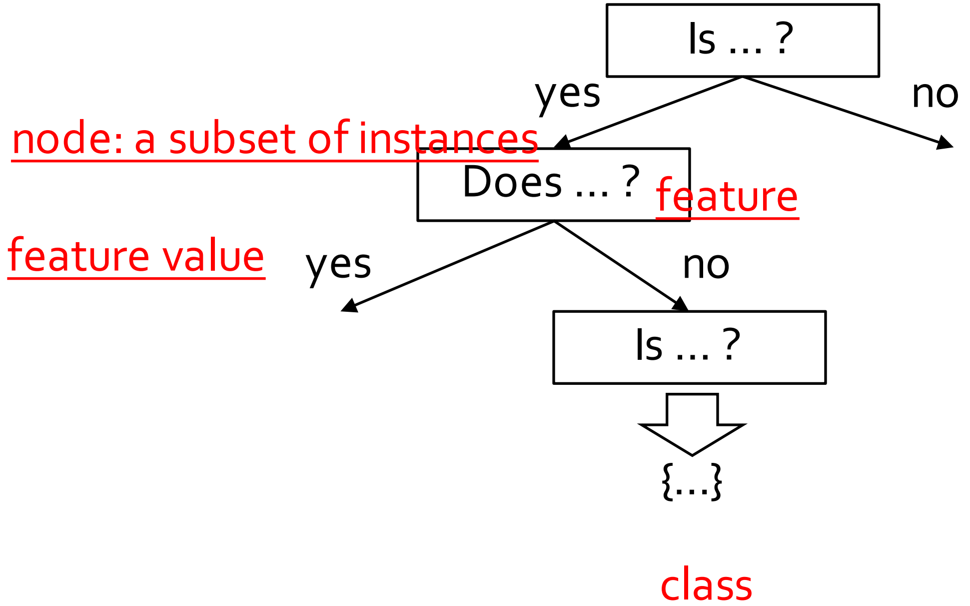
- How will you solve this multi-class classification problem?

{Barack Obama, Hillary Clinton, Ellen DeGeneres, Abraham Lincoln, Superman, …}



Decision Tree: Concepts

- A directed tree structure comprised of nodes
- Each **node** specified an evaluation on a feature
- Each **branch** corresponds to a feature value
- Each **leaf** signified a categorical decision or class



Decision Tree: Model Construction

- Top down, recursive divide-and-conquer
- Select best feature for root node
- Construct branch for each possible feature value
- Split data into mutually exclusive subsets along each branch
- **Repeat** procedure recursively for each branch
- Terminate into leaf node after adequate performance

- *Q1: Which feature to select?*
- *Q2: What is adequate performance?*

Call Back the Game

- How did you select your question/feature?

Call Back the Game

- How did you select your question/feature?

- Reduce **uncertainty** as much as possible

max ReducedUncertainty(instances|selected_attribute)

= Uncertainty({instances at parent node})

– Uncertainty({instances at child nodes*|selected_attribute})

*child nodes: values of the selected attributes

Q: How to measure uncertainty?

Entropy

- Entropy
  - A measure of **uncertainty** associated with a random number
  - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, y_2, \ldots, y_m\}$

$$H(Y) = -\sum_{i=1}^{m} p_i \log(p_i) \quad where\ p_i = P(Y = y_i)$$

  - Higher entropy → higher uncertainty
  - Lower entropy → lower uncertainty

Entropy

- High uncertainty

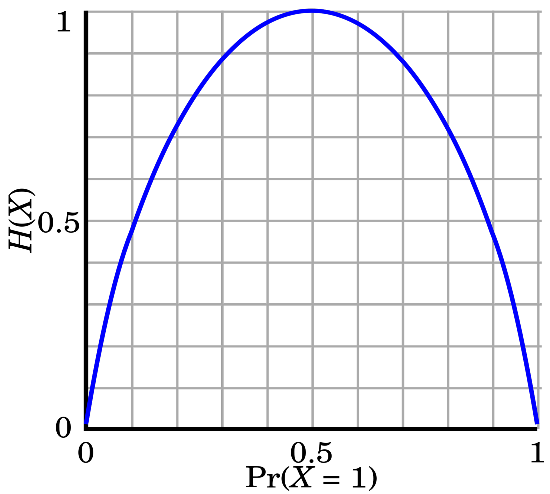
H(Pr(X=1)=0.5) = $-0.5\log_2(0.5) - 0.5\log_2(0.5) = 1$

- Low uncertainty

H(Pr(X=1)=0 or 1) = $-0\log_2(\boldsymbol{\varepsilon}) - 1\log_2(1-\boldsymbol{\varepsilon}) = 0$, if $\boldsymbol{\varepsilon} \to 0$

- Conditional entropy

$$H(Y|X) = \sum_x p(x) H(Y|X=x)$$



m = 2

Information Gain

- Information gain (IG) measures how much "information" an attribute gives us about the class
  - Attributes that perfectly partition should give maximal information
  - Unrelated attributes should give no information
- It measures the reduction in entropy: Defined as **expected reduction in entropy** by partitioning set of instances according to feature X:

$$max_X \quad IG(Y|X) = H(Y) - H(Y|X)$$

Information gain of class Y given feature X

Unconditional entropy of class Y

Conditional entropy of class Y given feature X

Exercise: Game Result Prediction

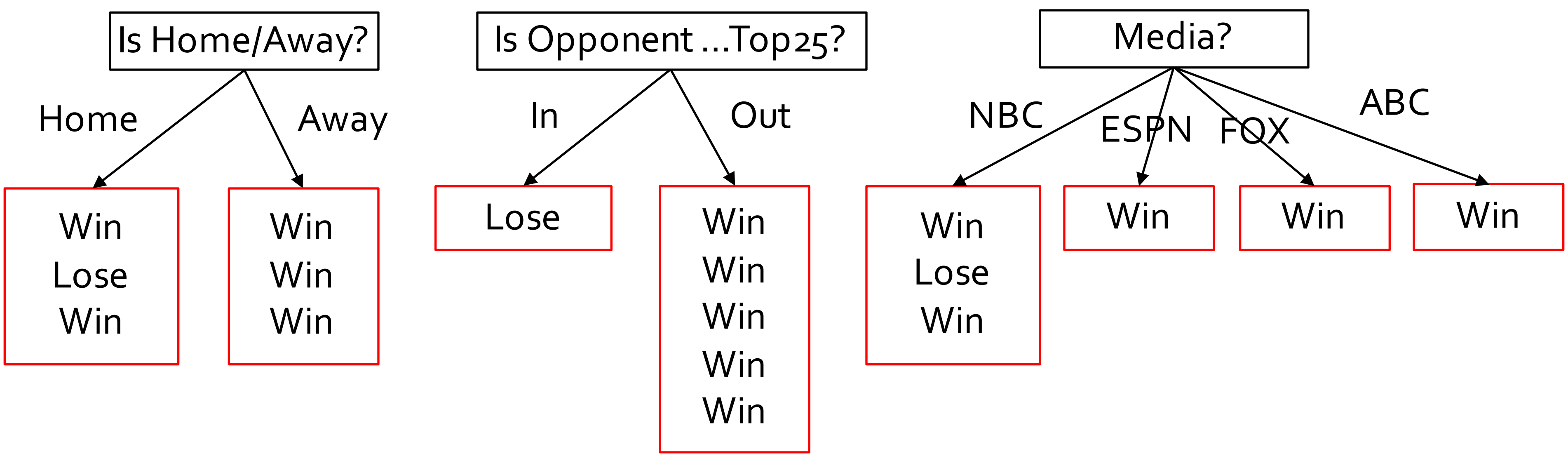
- It is a binary classification task: {Win, Lose}

			Is Home/Away?	Is Opponent in AP Top 25 at Preseason?	Media	Label: Win/Lose
1	9/2/17	Temple	Home	Out	1-NBC	Win
2	9/9/17	Georgia	Home	In	1-NBC	Lose
3	9/16/17	Boston College	Away	Out	2-ESPN	Win
4	9/23/17	Michigan State	Away	Out	3-FOX	Win
5	9/30/17	Miami Ohio	Home	Out	1-NBC	Win
6	10/7/17	North Carolina	Away	Out	4-ABC	Win
1	10/21/17	USC	Home	In	1-NBC	?
2	10/28/17	North Carolina State	Home	Out	1-NBC	?
3	11/4/17	Wake Forest	Home	Out	1-NBC	?
4	11/11/17	Miami Florida	Away	In	4-ABC	?
5	11/18/17	Navy	Home	Out	1-NBC	?
6	11/25/17	Stanford	Away	In	4-ABC	?

Splitting Instances to Nodes

			Is Home/Away?	Is Opponent in AP Top 25 at Preseason?	Media	Label: Win/Lose
1	9/2/17	Temple	Home	Out	1-NBC	Win
2	9/9/17	Georgia	Home	In	1-NBC	Lose
3	9/16/17	Boston College	Away	Out	2-ESPN	Win
4	9/23/17	Michigan State	Away	Out	3-FOX	Win
5	9/30/17	Miami Ohio	Home	Out	1-NBC	Win
6	10/7/17	North Carolina	Away	Out	4-ABC	Win

Partitioning set of instances according to feature



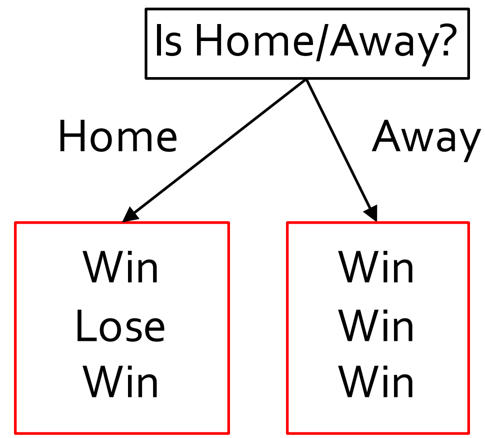
Calculating Information Gain (1)

		Label: Win/Lose
1	9/2/17	Win
2	9/9/17	Lose
3	9/16/17	Win
4	9/23/17	Win
5	9/30/17	Win
6	10/7/17	Win

Y = {Win*5, Lose*1}

H(Y) = $-\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} = 0.65$

$X_{HomeAway}$ = {Home*3, Away*3}



H(Y|$X_{HomeAway}$) = H(Y|Home)+H(Y|Away)

$= \frac{3}{6} \times \left(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right) + \frac{3}{6} \times (-\frac{3}{3}\log_2\frac{3}{3})$

= 0.5 $\times$ 0.92+0 = 0.46

IG(Y|$X_{HomeAway}$) = 0.65-0.46 = 0.19

Calculating Information Gain (2)

		Label: Win/Lose
1	9/2/17	Win
2	9/9/17	Lose
3	9/16/17	Win
4	9/23/17	Win
5	9/30/17	Win
6	10/7/17	Win

Y = {Win*5, Lose*1}

H(Y) = $-\frac{5}{6}\log_2\frac{5}{6}-\frac{1}{6}\log_2\frac{1}{6}=0.65$

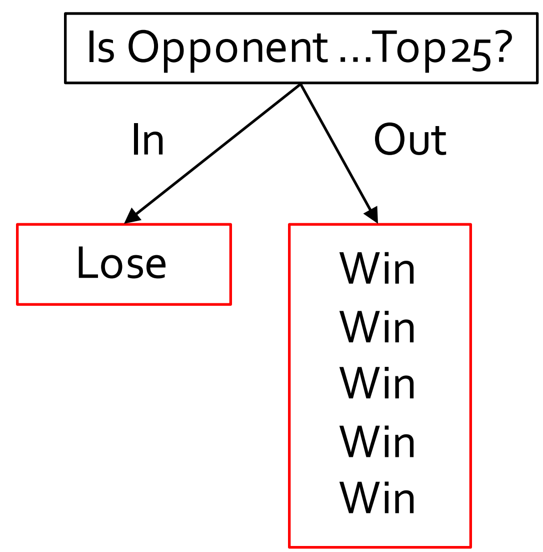
X_{Top25} = {In*1, Out*5}

H(Y|X_{Top25}) = H(Y|In)+H(Y|Out)

$=\frac{1}{6}\times\left(-\frac{1}{1}\log_2\frac{1}{1}\right)+\frac{5}{6}\times(-\frac{5}{5}\log_2\frac{5}{5})$

= 0

IG(Y|X_{Top25}) = 0.65-0 = 0.65



Calculating Information Gain (3)

		Label: Win/Lose
1	9/2/17	Win
2	9/9/17	Lose
3	9/16/17	Win
4	9/23/17	Win
5	9/30/17	Win
6	10/7/17	Win

Y = {Win*5, Lose*1}

H(Y) = $-\frac{5}{6}\log_2\frac{5}{6}-\frac{1}{6}\log_2\frac{1}{6}=0.65$

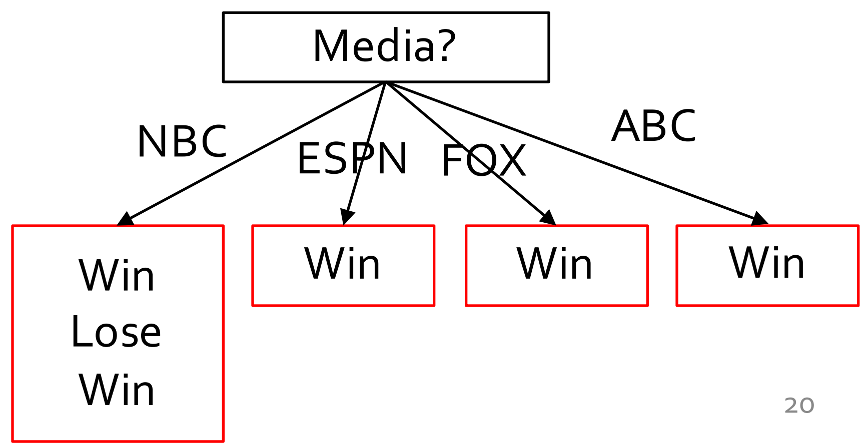
X_{Media} = {NBC*3, ESPN*1, FOX*1, ABC*1}

H(Y|X_{Media}) = H(Y|NBC)+H(Y|ESPN)+H(Y|FOX)+H(Y|ABC)

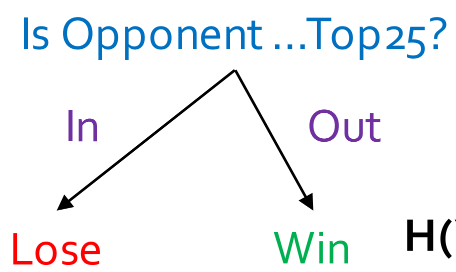
$=\frac{3}{6}\times\left(-\frac{2}{3}\log_2\frac{2}{3}-\frac{1}{3}\log_2\frac{1}{3}\right)+\frac{1}{6}\times(-\frac{1}{1}\log_2\frac{1}{1})+\frac{1}{6}\times(-\frac{1}{1}\log_2\frac{1}{1})+\frac{1}{6}\times(-\frac{1}{1}\log_2\frac{1}{1})$

= 0.5*0.92+0+0+0 = 0.46

IG(Y|X_{Media}) = 0.65-0.46 = 0.19



Final Decision Tree



$$\mathbf{H(Y)} = -\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} = 0.65$$

$$\mathbf{H(Y|X_{Top25})} = \mathrm{H(Y|In)}+\mathrm{H(Y|Out)}$$
$$= \frac{1}{6}\times\left(-\frac{1}{1}\log_2\frac{1}{1}\right) + \frac{5}{6}\times\left(-\frac{5}{5}\log_2\frac{5}{5}\right) = 0$$

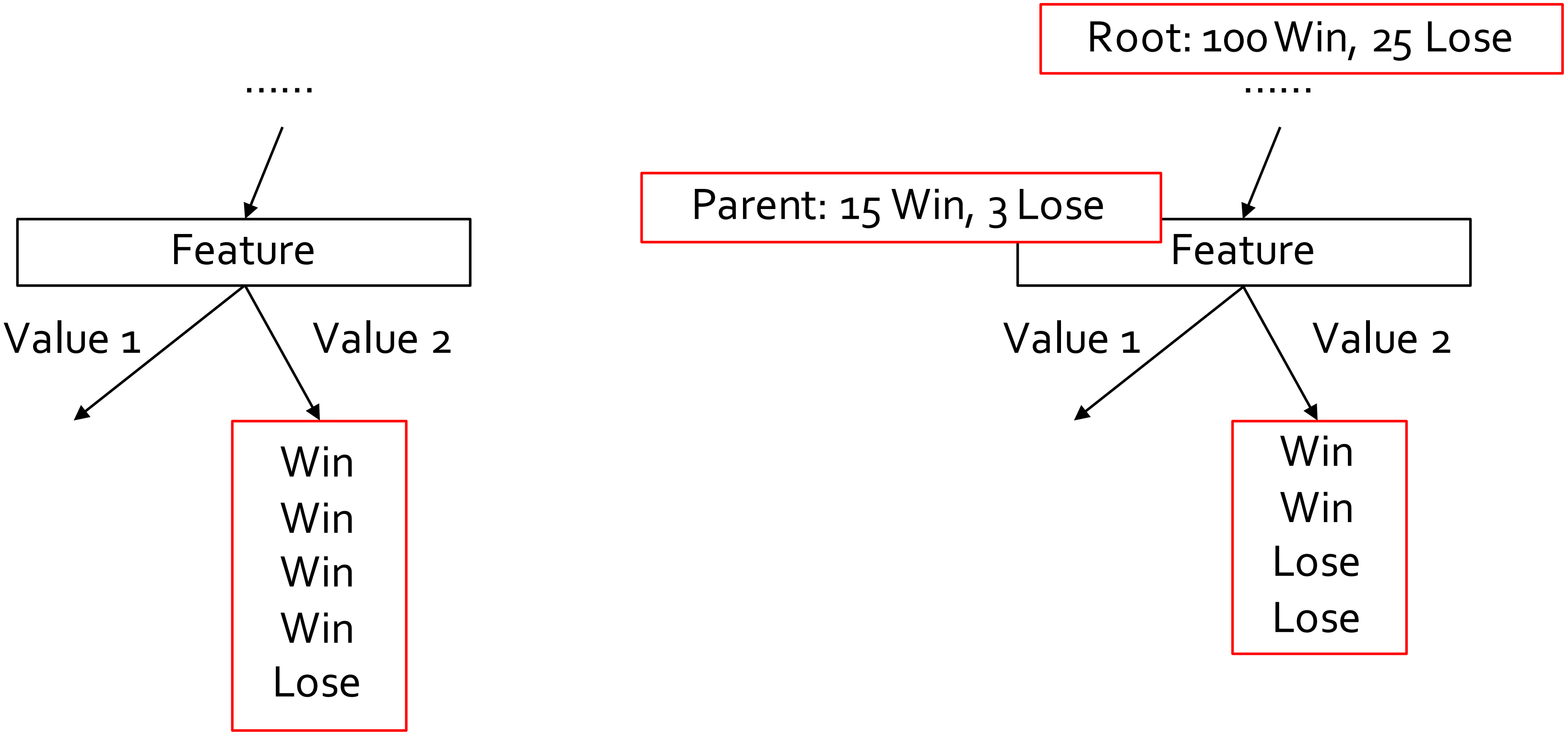
…

…

Terminate into leaf node after **adequate performance**

1. None of node splitting (with any other feature) can generate non-zero (or above-a-threshold) information gain.
2. All features have been used for splitting nodes.

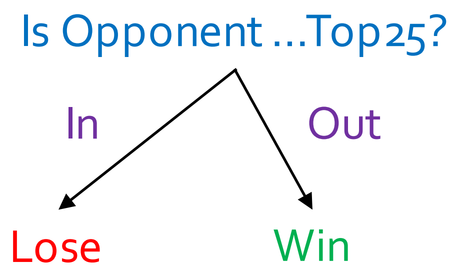
When We Terminate



Not pure, **imbalanced**:
(1) "Win": the majority at the leaf node
OR
(2) "Win=80%": a random variable

Not pure, **balanced**:
(1) "Win": the majority at the root node
(2) "Win": the majority at the parent node
OR
(3) "Win=50%": a random variable

Testing and Evaluation



			Is Home/Away?	Is Opponent in AP Top 25 at Preseason?	Media	Prediction	Ground truth
1	10/21/17	USC	Home	In	1-NBC	**Lose**	**Win**
2	10/28/17	North Carolina State	Home	Out	1-NBC	**Win**	**Win**
3	11/4/17	Wake Forest	Home	Out	1-NBC	**Win**	**Win**
4	11/11/17	Miami Florida	Away	In	4-ABC	**Lose**	**Lose**
5	11/18/17	Navy	Home	Out	1-NBC	**Win**	**Win**
6	11/25/17	Stanford	Away	In	4-ABC	**Lose**	**Lose**

Accuracy: 5/6 = 0.833

Q: How to improve it?

Improve this Game Prediction Model

- More relevant **features**: Correlation analysis?
- More training **instances**: Big data!
- More complicated **models**?

Quinlan's Example (1986): Playing Tennis

ID	Outlook	Temperature	Humidity	Windy	Label: Play?
1	Sunny	Hot	High	"False"	No
2	Sunny	Hot	High	"True"	No
3	Overcast	Hot	High	"False"	Yes
4	Rainy	Mild	High	"False"	Yes
5	Rainy	Cool	Normal	"False"	Yes
6	Rainy	Cool	Normal	"True"	No
7	Overcast	Cool	Normal	"True"	Yes
8	Sunny	Mild	High	"False"	No
9	Sunny	Cool	Normal	"False"	Yes
10	Rainy	Mild	Normal	"False"	Yes
11	Sunny	Mild	Normal	"True"	Yes
12	Overcast	Mild	High	"True"	Yes
13	Overcast	Hot	Normal	"False"	Yes
14	Rainy	Mild	High	"True"	No
15	Rainy	Hot	High	"False"	?

Information Gain Calculation

Y = {Yes*9, No*5}

$$\mathbf{H(Y)} = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

X_{Outlook} = {Sunny*5, Overcast*4, Rainy*5}

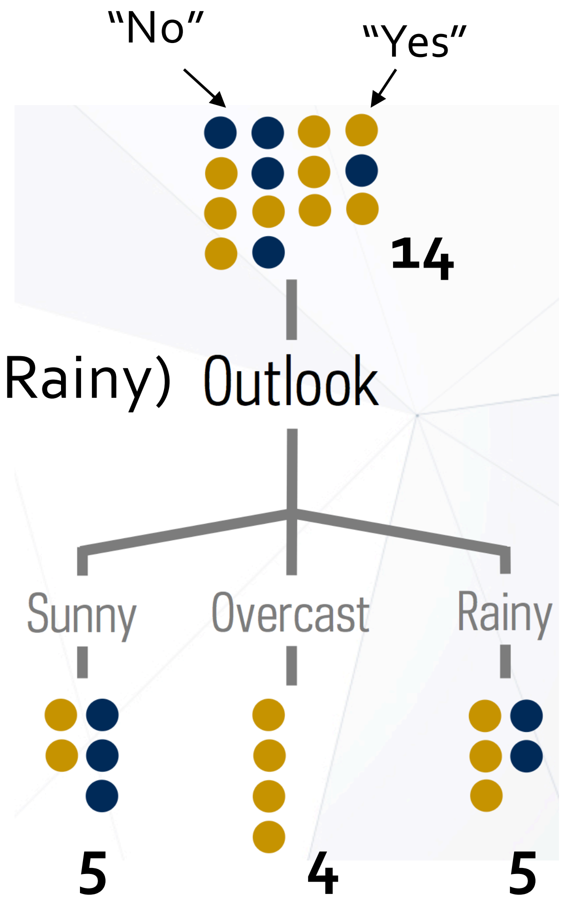
$\mathbf{H(Y|X_{Outlook})}$ = H(Y|Sunny)+H(Y|Overcast)+H(Y|Rainy)

$$= \frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)+\frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}\right)$$

$$+\frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)$$

= 0.345+0+0.345 = 0.69

$\mathbf{IG(Y|X_{Outlook})}$ = 0.94-0.69 = 0.25



Information Gain Calculation

$IG(Y|X_{Outlook}) = 0.25$

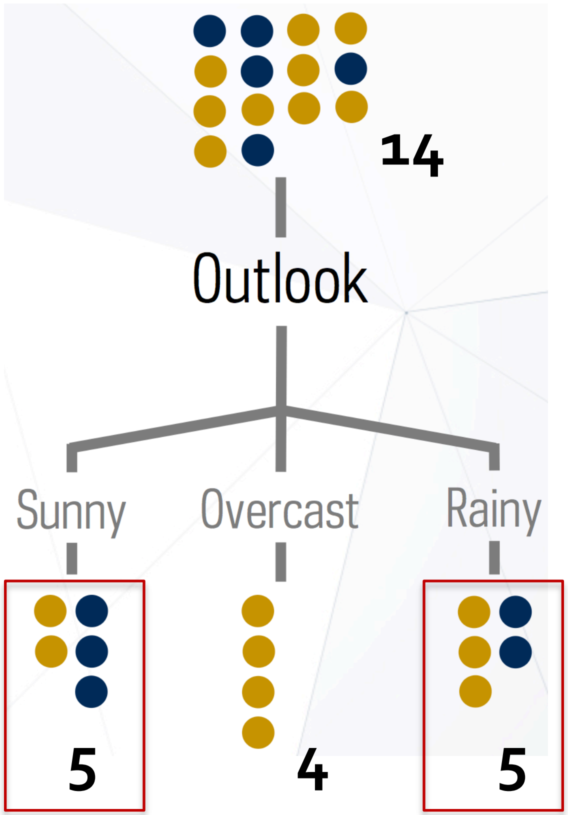
$IG(Y|X_{Temperature}) = 0.03$

$IG(Y|X_{Humidity}) = 0.15$

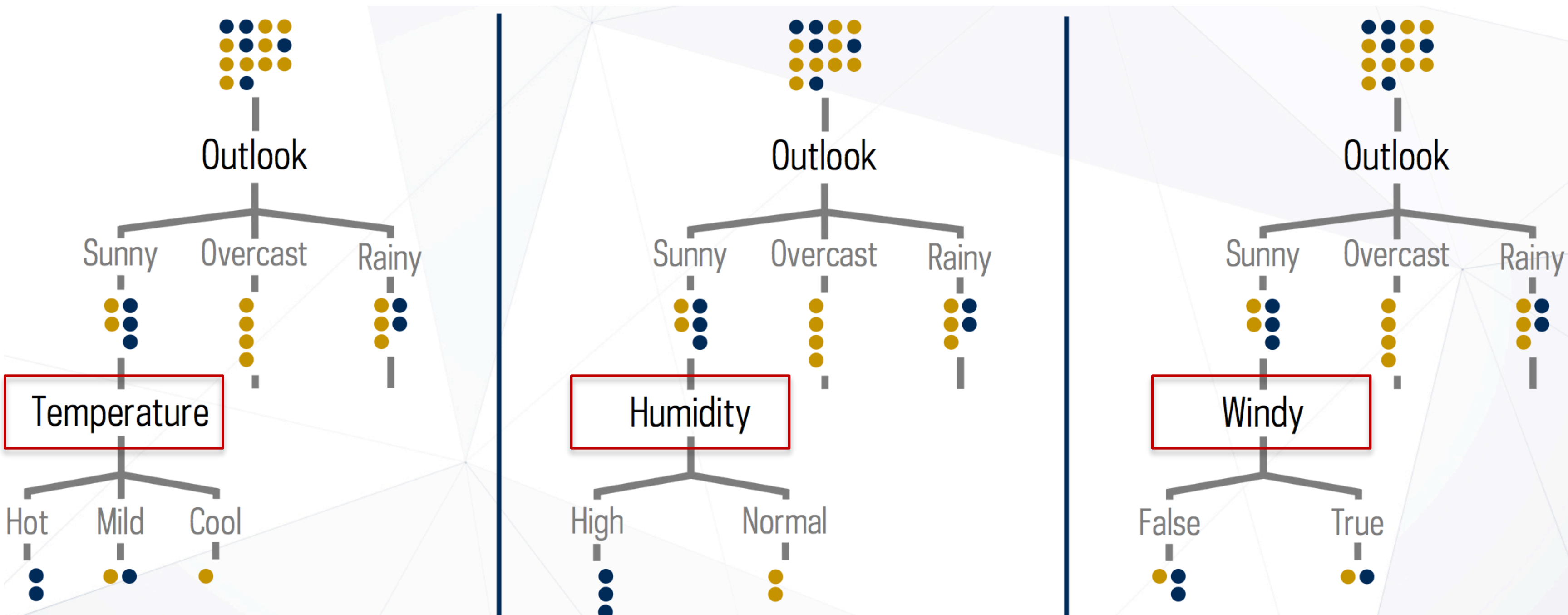
$IG(Y|X_{Windy}) = 0.05$

So the best feature is Outlook.

What's next step?



Next Step



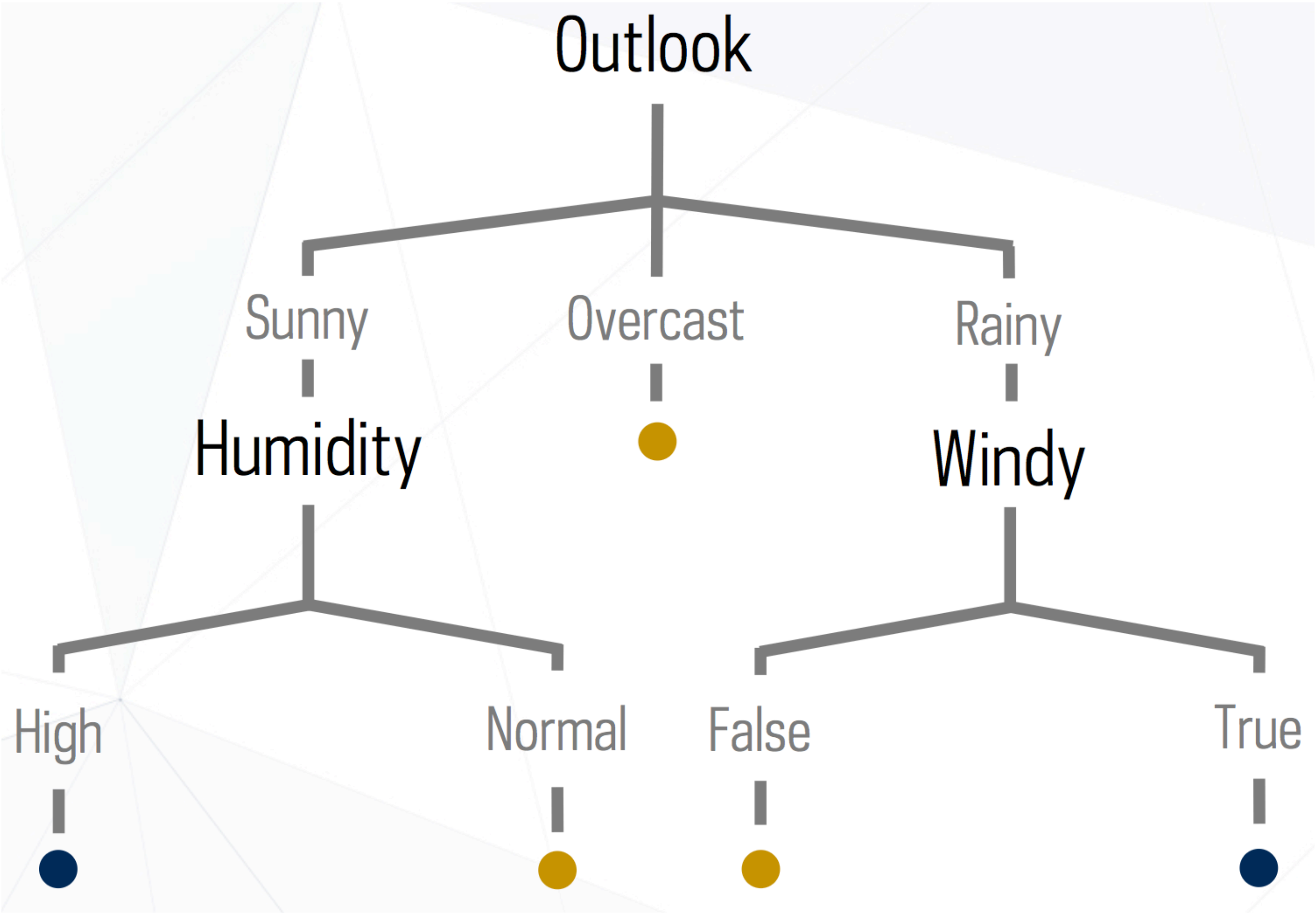
$IG(Y|\text{Temperature}) = 0.57$

$IG(Y|\text{Humidity}) = 0.97$

$IG(Y|\text{Windy}) = 0.02$

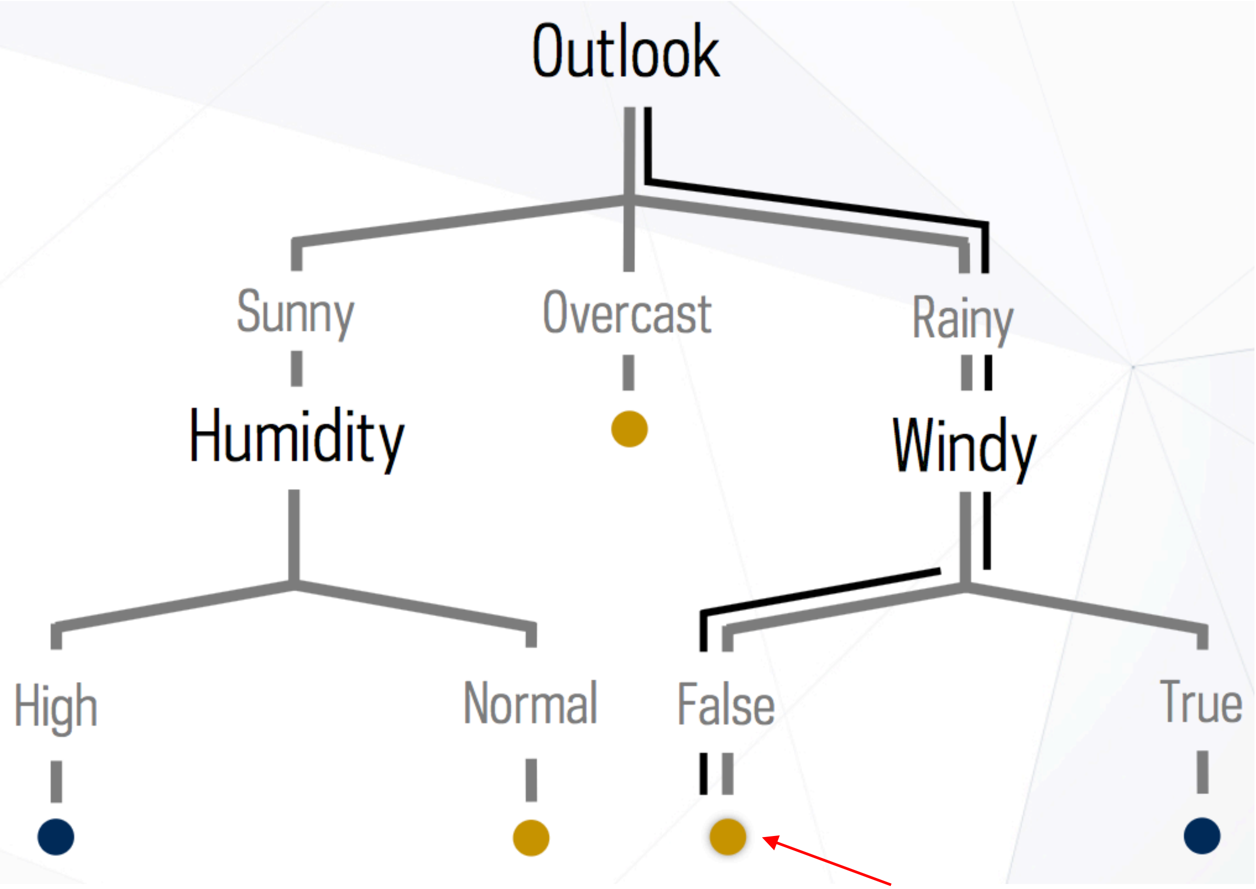
Good!

Final Decision Tree



Prediction

15	Rainy	Hot	High	"False"	?



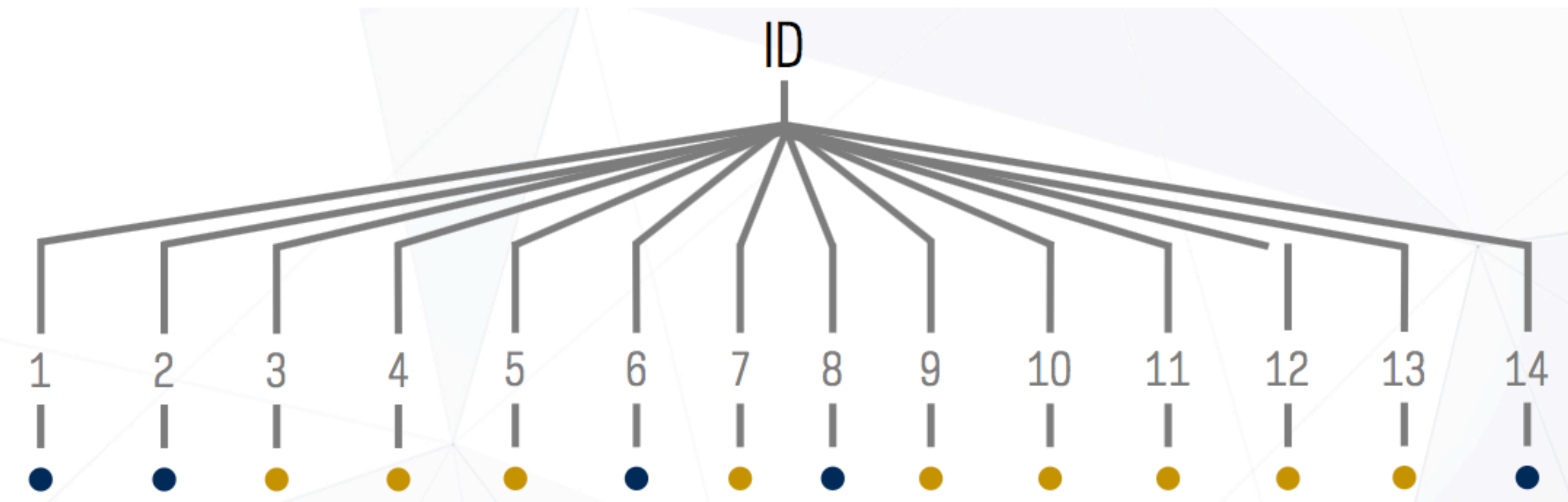
"Yes"

Highly-Branching Attribute

ID	Outlook	Temperature	Humidity	Windy	Label: Play?
1	Sunny	Hot	High	"False"	No
2	Sunny	Hot	High	"True"	No
3	Overcast	Hot	High	"False"	Yes
4	Rainy	Mild	High	"False"	Yes
5	Rainy	Cool	Normal	"False"	Yes
6	Rainy	Cool	Normal	"True"	No
7	Overcast	Cool	Normal	"True"	Yes
8	Sunny	Mild	High	"False"	No
9	Sunny	Cool	Normal	"False"	Yes
10	Rainy	Mild	Normal	"False"	Yes
11	Sunny	Mild	Normal	"True"	Yes
12	Overcast	Mild	High	"True"	Yes
13	Overcast	Hot	Normal	"False"	Yes
14	Rainy	Mild	High	"True"	No
15	Rainy	Hot	High	"False"	?

Highly-Branching Attribute

- Information gain measure is biased towards **highly-branching attributes** = with a large number of values
- Entropy of splitting on "ID" is 0. IG for "ID" is maximal.



Gain Ratio

- Corrects information by calculating the *intrinsic information* of a split
  - Information needed to identify branch
  - Accounts for number and size of branches
- Given entropy of instances distributed into branches

$$SplitInfo(S,F) = -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- Gain ratio is defined as

$$GainRatio(S,F) = \frac{IG(S,F)}{SplitInfo(S,F)}$$

S: "samples"
F: feature

Gain Ratio Calculation

$IG(Y|X_{Outlook}) = 0.25$

$\mathbf{IG(Y|X_{Temperature}) = 0.03}$

$IG(Y|X_{Humidity}) = 0.15$

$IG(Y|X_{Windy}) = 0.05$

$SplitInfo(X_{Temperature})$

$$= -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.56$$

$GainRatio(X_{Temperature})$

$=0.03/1.56 =$ **0.02**

Temperature
Hot
Hot
Hot
Mild
Cool
Cool
Cool
Mild
Cool
Mild
Mild
Mild
Hot
Mild

Splitting Criterion

- Information Gain (used in ID3)
  - Iterative Dichotomiser 3 invented by Ross Quinlan in 1986
- Gain Ratio (used in C4.5)
  - C4.5 is an extension of Quinlan's earlier ID3 algorithm, developed by Ross Quinlan
  - It became quite popular after ranking #1 in the Top 10 Algorithms in Data Mining pre-eminent paper published by Springer LNCS in 2008
- Gini Measure (used in CART)
  - Classification and Regression Trees by Breiman et al. in 1984

Gini Index (CART)

- Another splitting criteria. Defined as

$$Gini = 1 - \sum_{i=1}^{K} p_k^2$$

where p_k denotes the proportion of instances belonging to class k (k = 1… K).

Compared with Information Entropy (Info, or H):

$$Info = H = -\sum_{i=1}^{K} p_k \log p_k$$

IG vs Gini

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

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

$$Gain(A) = Info(D) - Info_A(D)$$

$$\Delta gini(A) = gini(D) - gini_A(D)$$

Maximize

Gini Index Calculation

Y = {Yes*9, No * 5}

$$\text{Gini}(Y) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.46$$

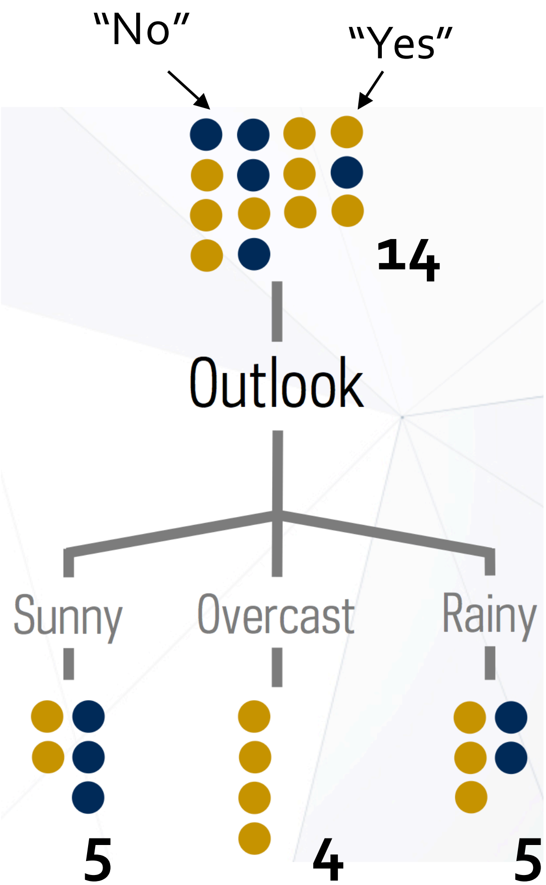
X_{Outlook} = {Sunny*5, Overcast*4, Rainy*5}

$$\text{Gini}(Y|X_{\text{Outlook}}) = \frac{5}{14} \times \left(1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right)$$

$$+ \frac{4}{14} \times \left(1 - \left(\frac{4}{4}\right)^2\right)$$

$$+ \frac{5}{14} \times \left(1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right) = 0.34$$

$$\Delta\text{Gini}(Y|X_{\text{Outlook}}) = 0.46 - 0.34 = 0.12$$



Gini Index Calculation

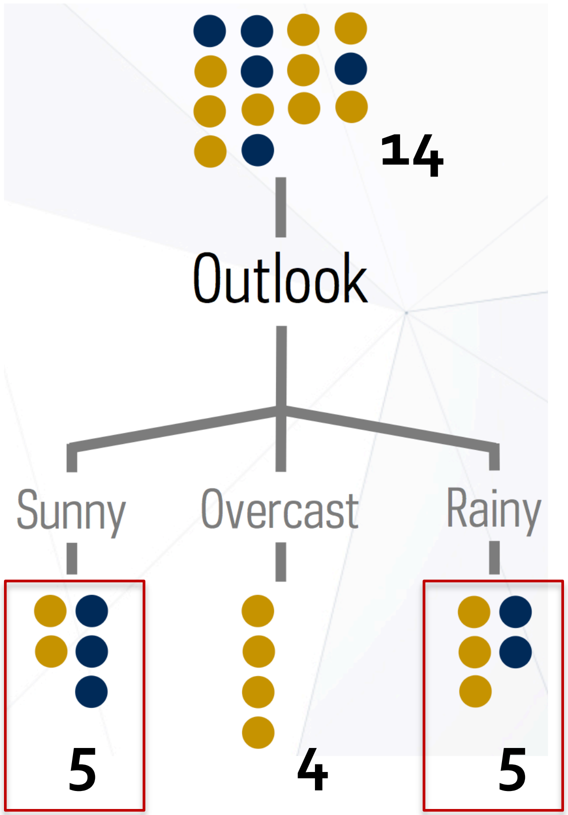
Δgini $(Y|X_{Outlook})$ = 0.12

Δgini$(Y|X_{Temperature})$ = 0.02

Δgini$(Y|X_{Humidity})$ = 0.09

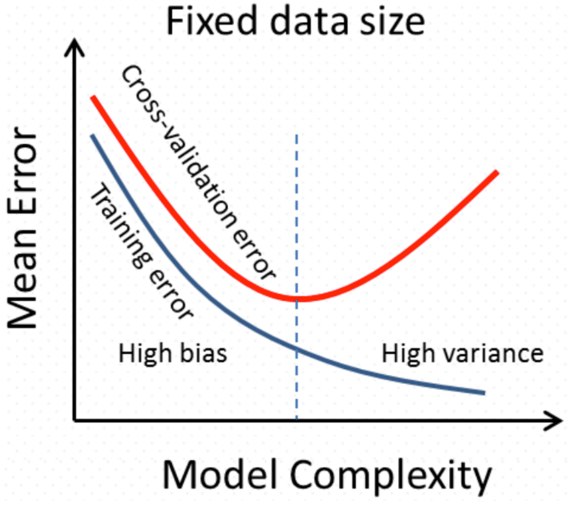
Δgini$(Y|X_{Windy})$ = 0.03

So the best feature is Outlook.



Overfitting

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples



Resolve Overfitting

- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early - do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a “fully grown” tree — get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the “best pruned tree”

Summary: Decision Tree for Classification

- Describe the difference between classification and clustering
- Describe two steps of the classification process

- Describe what is entropy; describe and compare the following "feature selection measures" or called "splitting criteria": information gain, gain ratio, and gini index.
- Given training instances and their attributes, construct by hand and implement using Python Decision Tree models:
  - ID3: information gain
  - C4.5: gain ratio
  - CART: gini index

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Comm. of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. SIGMOD'02
- H. V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001
- T. Redman. Data Quality: Management and Technology. Bantam Books, 1992
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995