Introduction to Machine Learning
Decision Trees

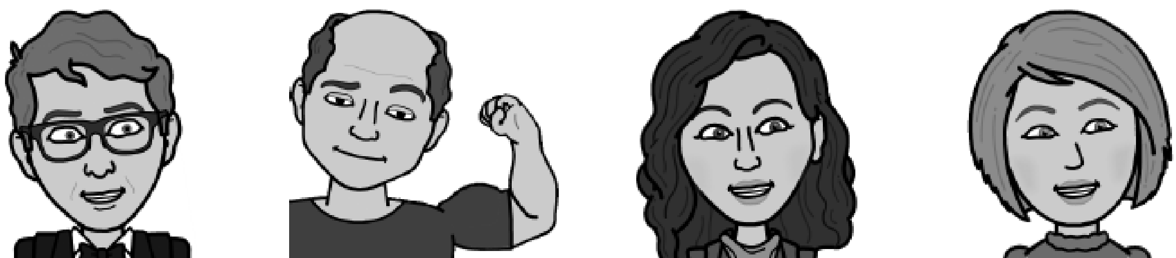
Prof. Chang-Chieh Cheng

Information Technology Service Center

National Chiao Tung University

Decision Tree

- Guess-who game



(a) Brian (b) John (c) Aphra (d) Aoife

Man	Long Hair	Glasses	Name
Yes	No	Yes	Brian
Yes	No	No	John
No	Yes	No	Aphra
No	No	No	Aoife

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Decision Tree

- Guess-who game
  - Build a decision tree from a set of observed data
  - Decision tree is not unique for the same dataset

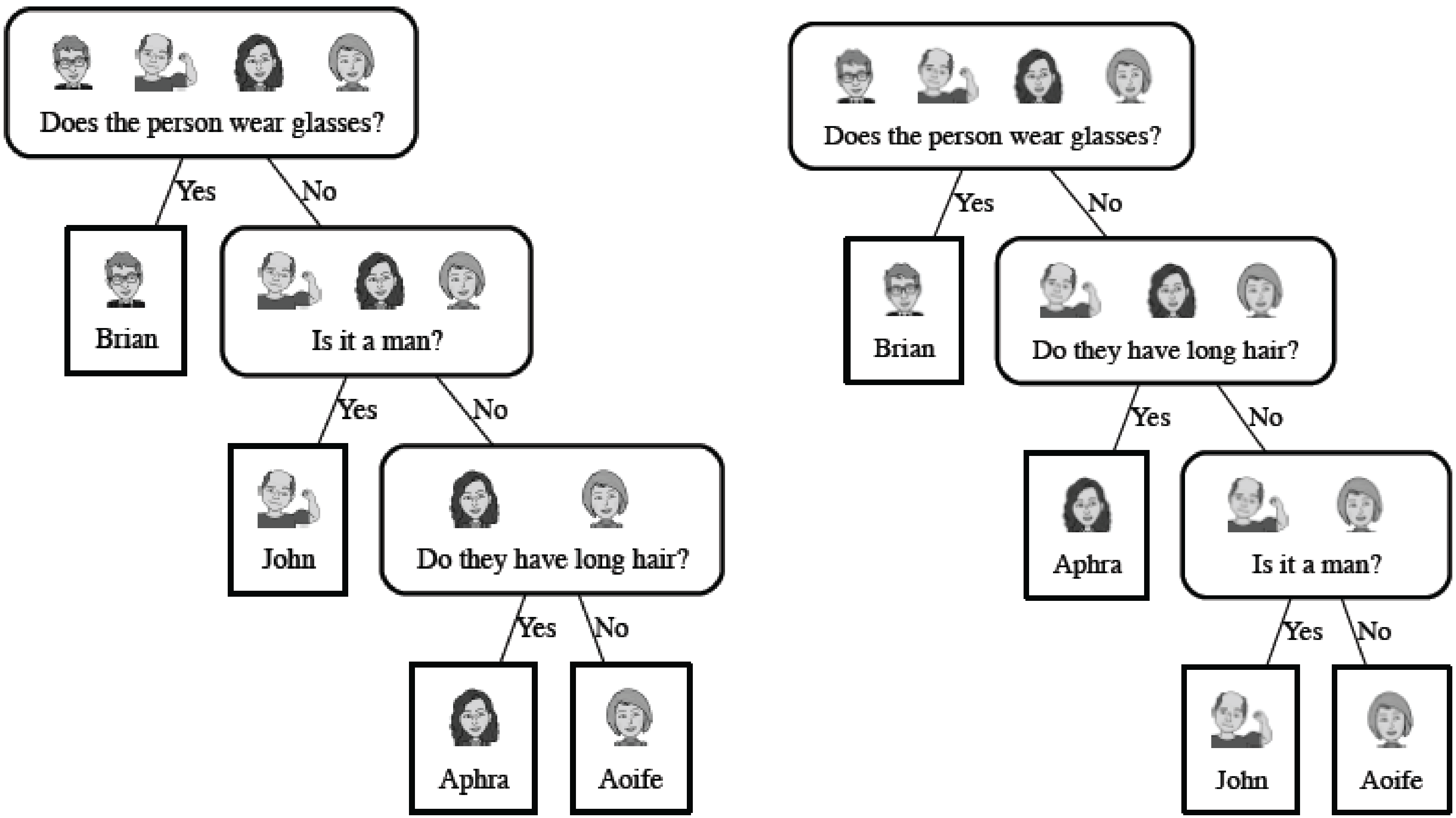
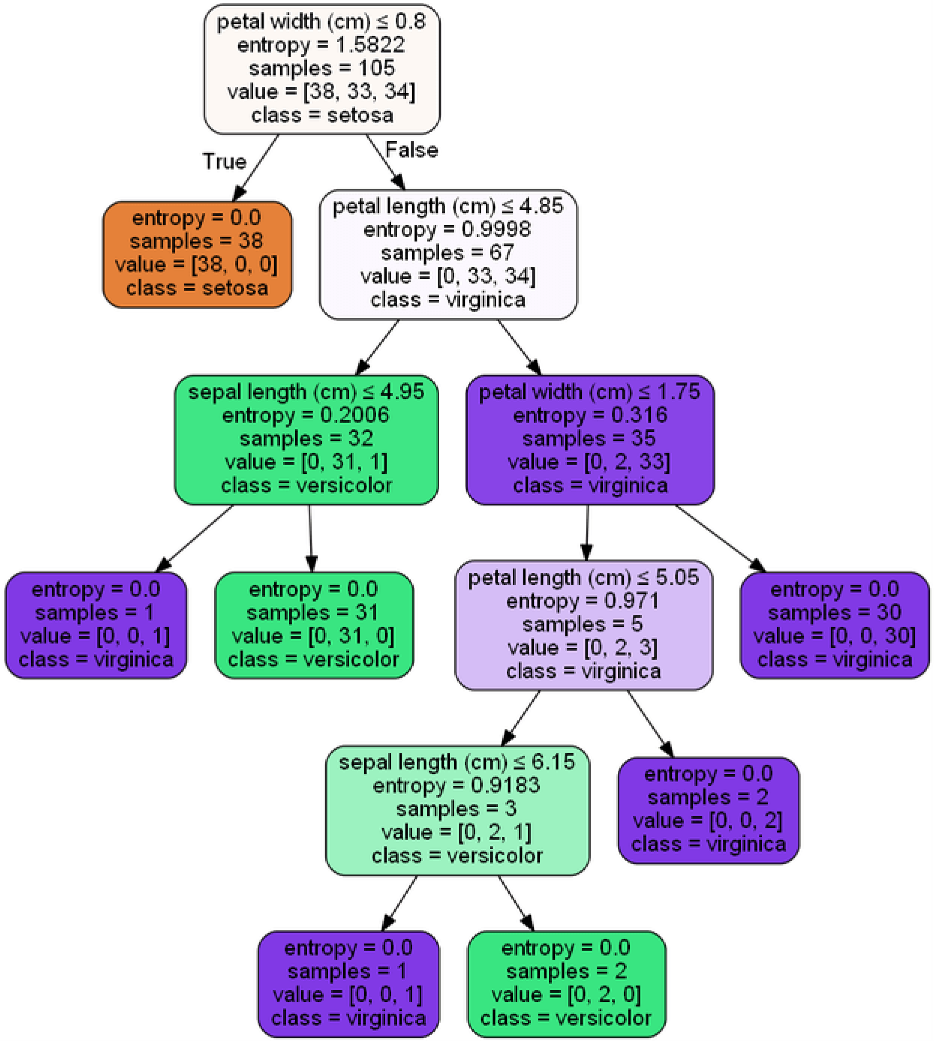


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Decision Tree

- For example, a sample of an iris
  - petal width = 1.2cm
  - petal length = 5.0cm
  - sepal length = 7.0cm
  - sepal width = 2.4cm
  - ➔ **versicolor**

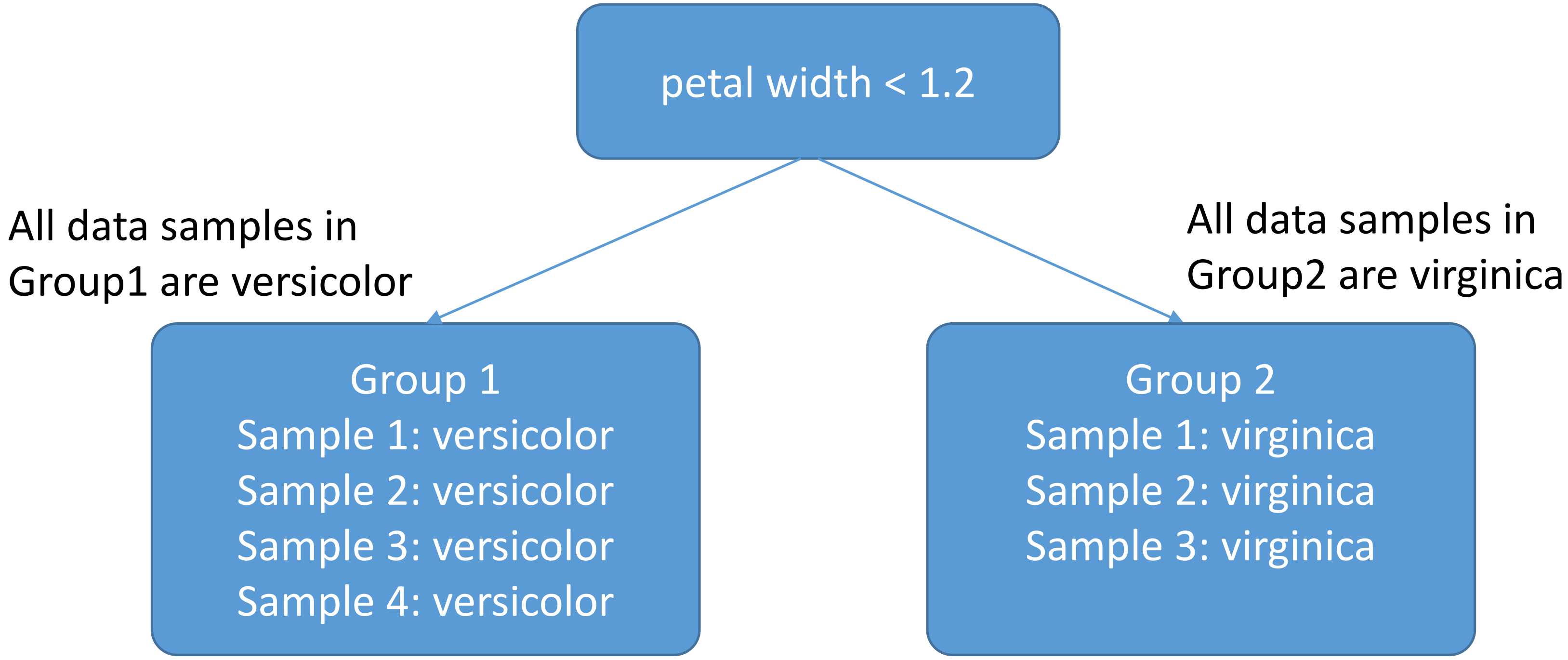


Decision Tree

- A decision tree consists of:
  - a root node (or starting node),
  - interior nodes
  - leaf nodes (or terminating nodes).
- Each of the non-leaf nodes (root and interior) in the tree specifies a test to be carried out on one of the query's descriptive features.
- Each of the leaf nodes specifies a predicted classification for the query.
- The height of a decision tree should be low
  - A shallow decision tree is good

Decision Tree

- The main essential of building a decision
  - Find a key feature for each node such that each separated subgroups has **lowest information complexity**
  - For example, an ideal situation of IRIS:



- How to decide the information complexity of a dataset?
  - **Entropy**

Entropy

- In physics, entropy is a measurement to describe that how chaotic of a system.
  - $Entropy = K \ln R$
    - *K*: a constant of the system
    - *R*: a state of the system
- In computer science, entropy is a measurement to describe that the complexity of a set of data

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

- $X = \{x_1, x_2, \dots, x_n\}$, set of random variables, types of data, or features.
- *b*: the base of log, commonly *b* is 2 or *e*

Entropy

- $H(X) = 0.81$
  - $p(\text{spare}) = 9/12 = 0.75$
  - $\log_2 p(\text{spare}) = -0.415$
  - 0.75 * (-0.415) =-0.3113

  - $p(\text{king}) = 3/12 = 0.25$
  - $\log_2 p(\text{spare}) = -2$
  - 0.25 * (-0.2) =-0.5

  - -((-0.5) + (-0.3113)) = 0.81

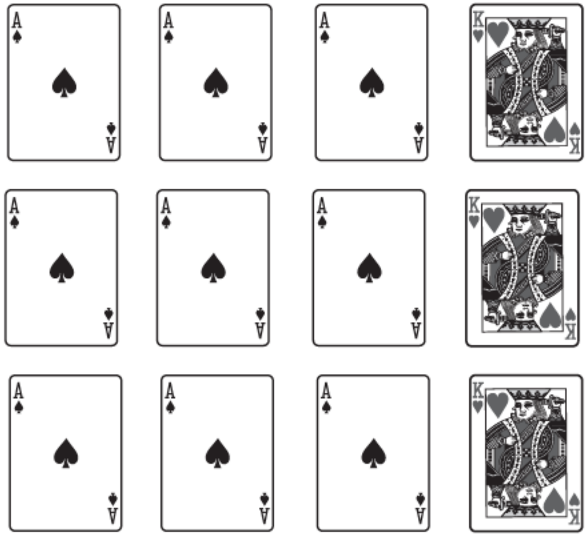


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Entropy

- $H(X) = 0$
  - $p(\text{spare}) = 12/12 = 1.0$
  - $\log_2 p(\text{spare}) = 0.0$

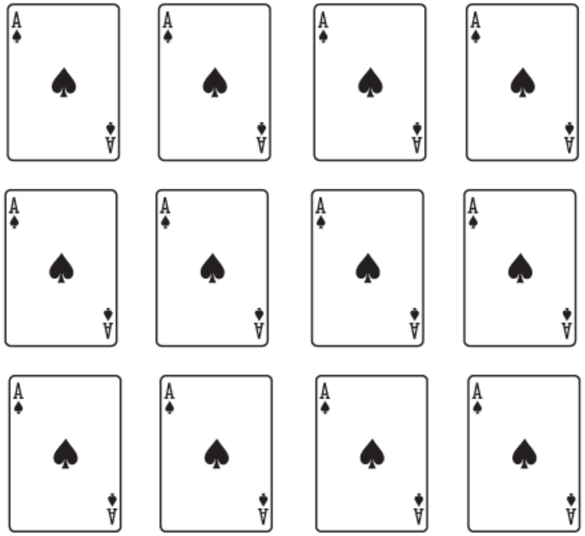


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Entropy

- $H(X) = 1.5$
  - p(spare) = 6/12 = 0.5
  - $\log_2 p$(spare) = -1
  - 0.5 * (-1) =-0.5

  - p(king) = 3/12 = 0.25
  - $\log_2 p$(spare) = -2
  - 0.25 * (-0.2) =-0.5

  - p(Diamond) = 3/12 = 0.25
  - $\log_2 p$(Diamond) = -2
  - 0.25 * (-0.2) =-0.5

  - -((-0.5) + (-0.5) + (-0.5)) = 1.5

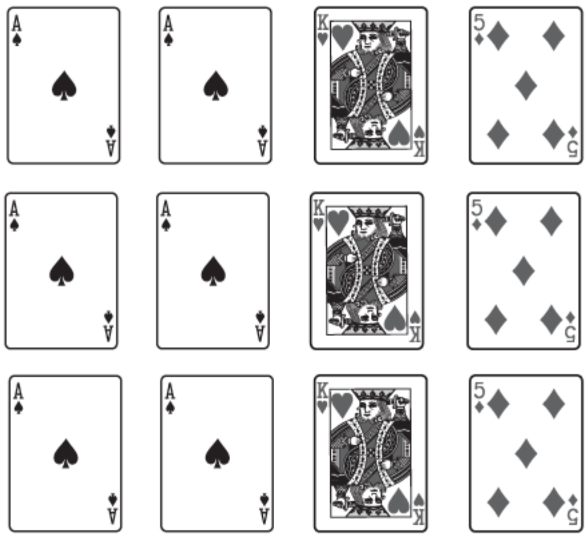


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Entropy

- High entropy:
  - Chaotic information
  - Many different kinds of information in a dataset
- Low entropy
  - Monotonous information
  - The data content near to invariance

Entropy

- Entropy of training data

$$H(T, D) = -\sum_{t \in T} p(t) \log_b p(t)$$

  - where *T* is the target set, *D* is the training dataset, and

$$p(t) = \frac{|\{D_t \subseteq D \mid \forall d \in D_t, T(d) = t\}|}{|D|}$$

Entropy

- Example: The vegetation classification dataset.

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

  - p(chaparral) = 3 / 7 = 0.42857
  - $\log_2 p$(chaparral) = -1.22239
  - 0.42857 * (-1.22239) =-0.52388

  - p(riparian) = p(conifer) 2 / 7 = 0.286
  - $\log_2 p$(riparian) = $\log_2 p$(conifer) = -1.80735
  - 0.286 * (-1.80735) =-0.51639
- $H(Vegetation, D)$ = -((-0.52388) + (-0.51639) + (-0.51639)) = 1.5567

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Information Gain

- The definition:
  - Select a **feature *F*** from the training **dataset *D***, the definition of **information gain** *G* as follows:

$$G(F, D) = H(T, D) - R(F, D)$$

  - where *R* is called the **remainder** that is defined as follows:

$$R(F, D) = \sum_{f \in F} \frac{\left|\left\{D_f \subseteq D \middle| \forall d \in D_f, F(d) = f\right\}\right|}{|D|} H\left(T, D_f\right)$$

Building a Decision Tree

- Example: The vegetation classification dataset.

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$-(\frac{2}{4}log_2\frac{2}{4}+2\left(\frac{1}{4}log_2\frac{1}{4}\right)) = 1.5$$

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	'true'	$\mathcal{D}_1$	$\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_7$	1.5	1.2507	0.3060
	'false'	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_5$	0.9183		
SLOPE	'flat'	$\mathcal{D}_3$	$\mathbf{d}_5$	0	0.9793	0.5774
	'moderate'	$\mathcal{D}_4$	$\mathbf{d}_2$	0		
	'steep'	$\mathcal{D}_5$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7$	1.3710		
ELEVATION	'low'	$\mathcal{D}_6$	$\mathbf{d}_2$	0	0.6793	0.8774
	'medium'	$\mathcal{D}_7$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		
	'high'	$\mathcal{D}_8$	$\mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_7$	0.9183		
	'highest'	$\mathcal{D}_9$	$\mathbf{d}_6$	0		

$$1.5 \times \frac{4}{7} + 0.9183 \times \frac{3}{7} = 1.2507$$

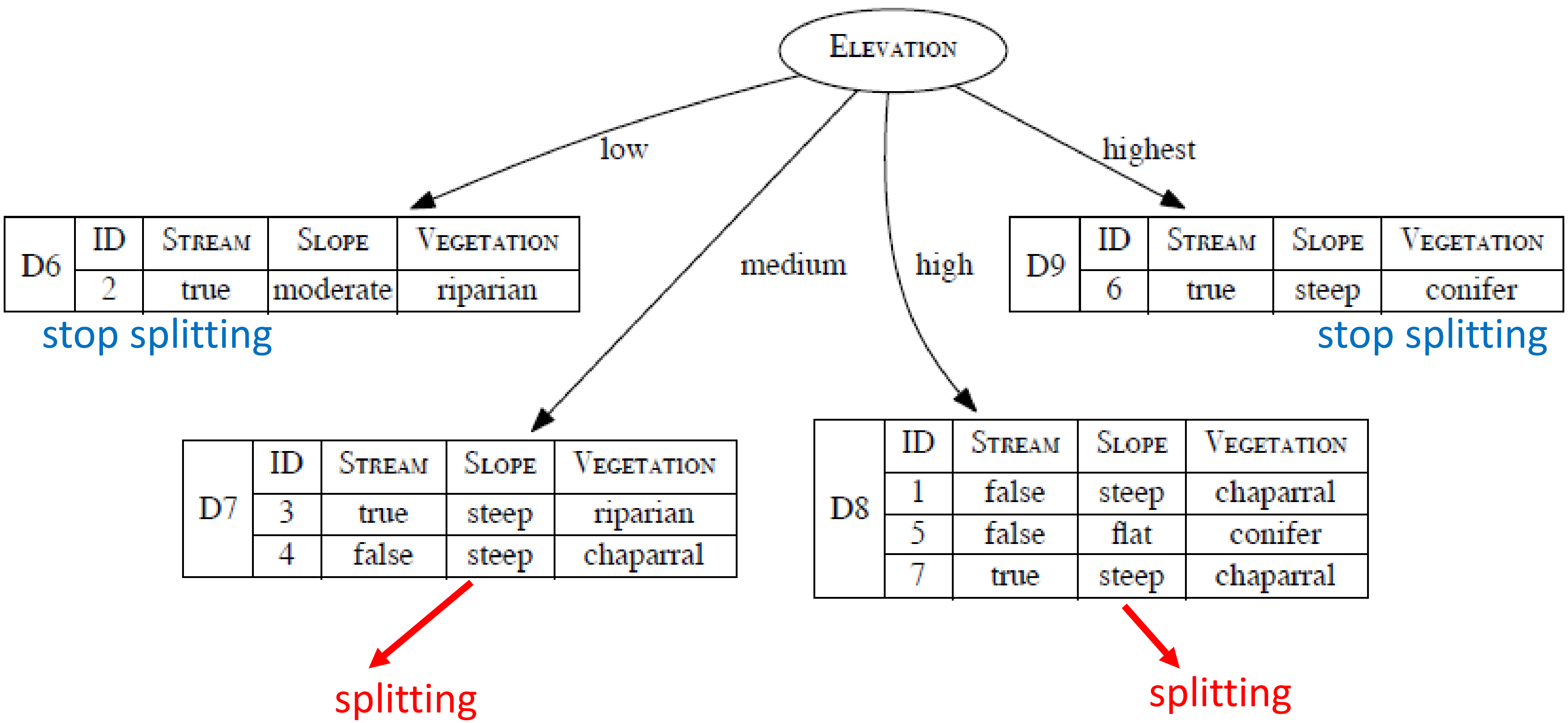
$$1.5567 - 1.2507 = 0.3060$$

$$-(\left(\frac{2}{3}log_2\frac{2}{3}\right)+\left(\frac{1}{3}log_2\frac{1}{3}\right)) = 0.9183$$

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Building a Decision Tree

- Select ELEVATION as the key feature to split data such that the information gain is the maximum.



D6	ID	STREAM	SLOPE	VEGETATION
	2	true	moderate	riparian

D7	ID	STREAM	SLOPE	VEGETATION
	3	true	steep	riparian
	4	false	steep	chaparral

D8	ID	STREAM	SLOPE	VEGETATION
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

D9	ID	STREAM	SLOPE	VEGETATION
	6	true	steep	conifer

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Building a Decision Tree

- Splitting D_7
  - $H(Vegetation, D_7) = -\left(\left(\frac{1}{2} log_2 \frac{1}{2}\right) + \left(\frac{1}{2} log_2 \frac{1}{2}\right)\right) = 1.0$
  - Information gains:

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	*'true'*	$\mathcal{D}_{10}$	$\mathbf{d}_3$	0	0	1.0
	'false'	$\mathcal{D}_{11}$	$\mathbf{d}_4$	0		
SLOPE	*'flat'*	$\mathcal{D}_{12}$		0	1.0	0
	'moderate'	$\mathcal{D}_{13}$		0		
	'steep'	$\mathcal{D}_{14}$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Building a Decision Tree

- Splitting D_7

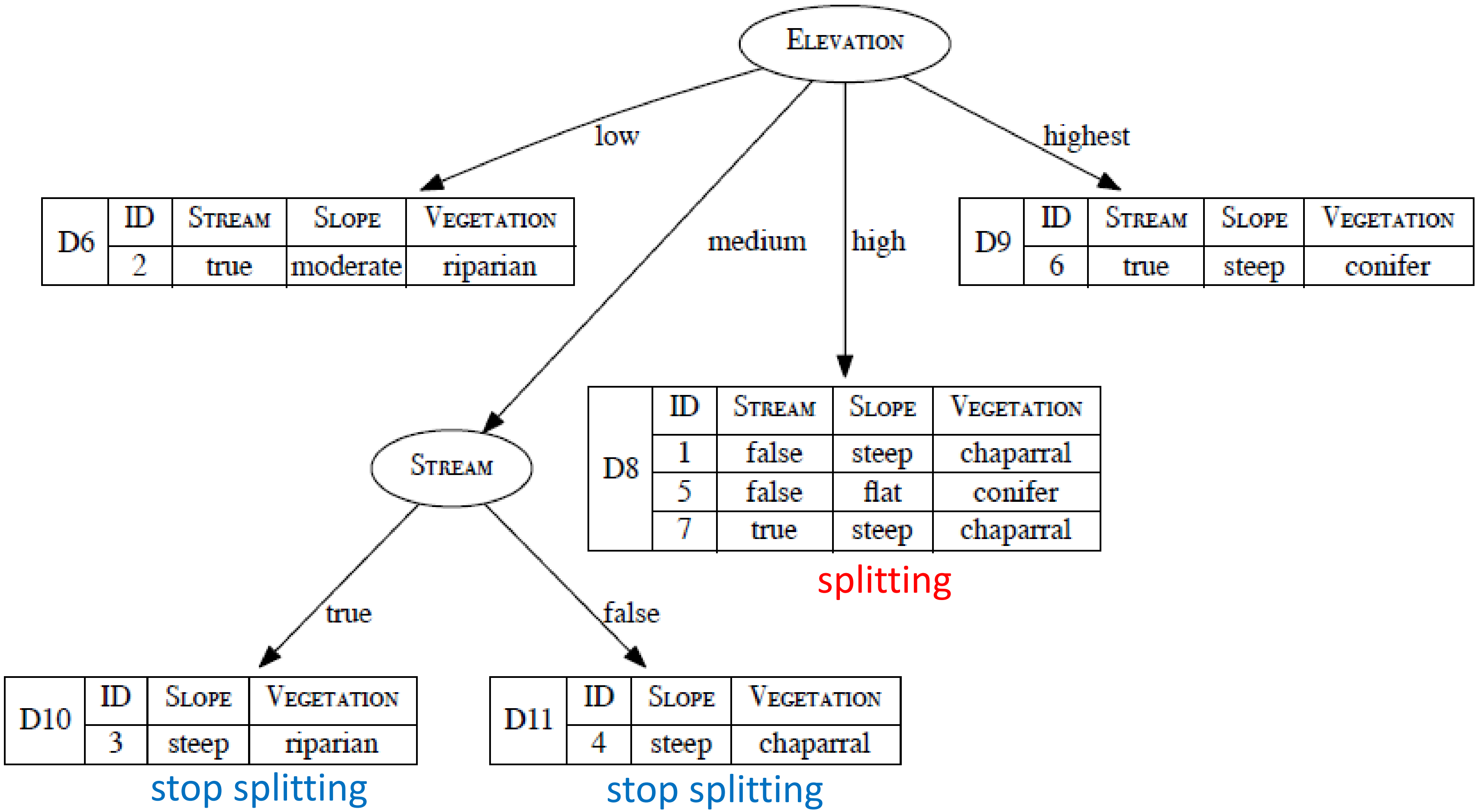


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Building a Decision Tree

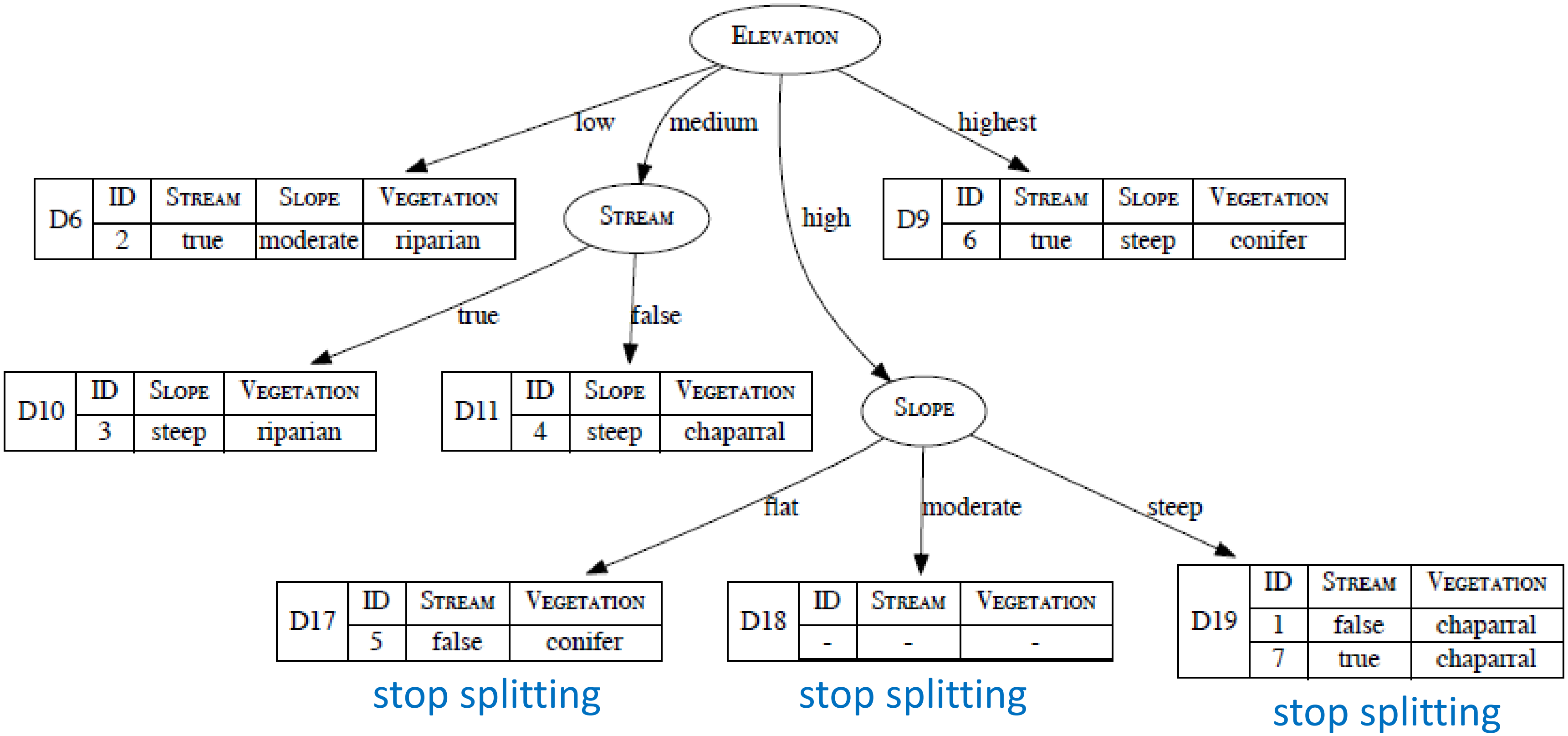
- Splitting D_8
  - $H(Vegetation, D_8) = -\left(\left(\frac{2}{3} log_2 \frac{2}{3}\right) + \left(\frac{1}{3} log_2 \frac{1}{3}\right)\right) = 0.9183$
  - Information gains:

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	*'true'*	$\mathcal{D}_{15}$	$\mathbf{d}_7$	0	0.6666	0.2517
	'false'	$\mathcal{D}_{16}$	$\mathbf{d}_1, \mathbf{d}_5$	1.0		
SLOPE	*'flat'*	$\mathcal{D}_{17}$	$\mathbf{d}_5$	0	0	0.9183
	'moderate'	$\mathcal{D}_{18}$		0		
	'steep'	$\mathcal{D}_{19}$	$\mathbf{d}_1, \mathbf{d}_7$	0		

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

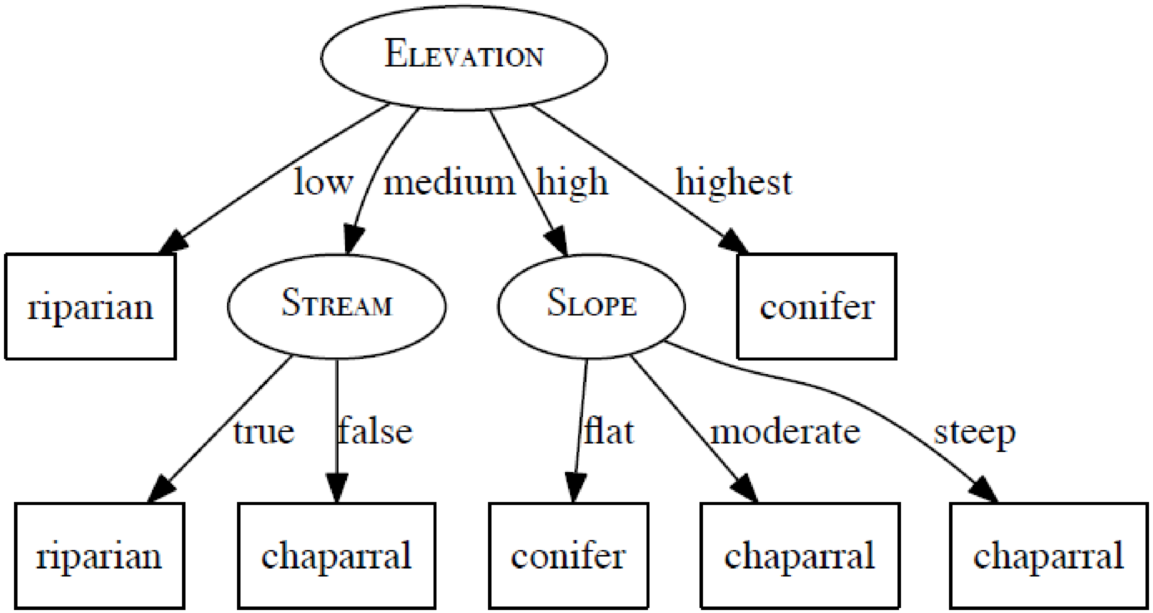
Building a Decision Tree

- Splitting D_8



Building a Decision Tree

- The final result



- What prediction will this decision tree model return for thefollowing query?
  - STREAM = ’true’, SLOPE=’Moderate’, ELEVATION=’High’
  - ➔ VEGETATION = ’Chaparral’

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Information Gain Ratio

- Information gain ratio, GR
- GR is also called weighted information gain
- GR can be considered as normalized information gain
- Select a **feature *F*** from the training **dataset *D***, the definition of *GR* as follows:

$$GR(F, D) = \frac{G(F, D)}{H(F, D)}$$

Information Gain Ratio

- Example:

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	'true'	$\mathcal{D}_1$	$\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_7$	1.5	1.2507	0.3060
	'false'	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_5$	0.9183		
SLOPE	'flat'	$\mathcal{D}_3$	$\mathbf{d}_5$	0	0.9793	0.5774
	'moderate'	$\mathcal{D}_4$	$\mathbf{d}_2$	0		
	'steep'	$\mathcal{D}_5$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7$	1.3710		
ELEVATION	'low'	$\mathcal{D}_6$	$\mathbf{d}_2$	0	0.6793	0.8774
	'medium'	$\mathcal{D}_7$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		
	'high'	$\mathcal{D}_8$	$\mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_7$	0.9183		
	'highest'	$\mathcal{D}_9$	$\mathbf{d}_6$	0		

$$H(\mathrm{STREAM}, D) = -\left(\left(\frac{4}{7} log_2 \frac{4}{7}\right) + \left(\frac{3}{7} log_2 \frac{3}{7}\right)\right) = 0.9852$$

$$H(\mathrm{SLOPE}, D) = -\left(2\left(\frac{1}{7} log_2 \frac{1}{7}\right) + \left(\frac{5}{7} log_2 \frac{5}{7}\right)\right) = 1.1488$$

$$H(\mathrm{ELEVATION}, D) = -\left(2\left(\frac{1}{7} log_2 \frac{1}{7}\right) + \left(\frac{2}{7} log_2 \frac{2}{7}\right) + \left(\frac{3}{7} log_2 \frac{3}{7}\right)\right) = 1.8424$$

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Information Gain Ratio

- Example:

$$GR(\textsc{Stream}, \mathcal{D}) = \frac{0.3060}{0.9852} = 0.3106$$

$$GR(\textsc{Slope}, \mathcal{D}) = \frac{0.5774}{1.1488} = 0.5026$$

$$GR(\textsc{Elevation}, \mathcal{D}) = \frac{0.8774}{1.8424} = 0.4762$$

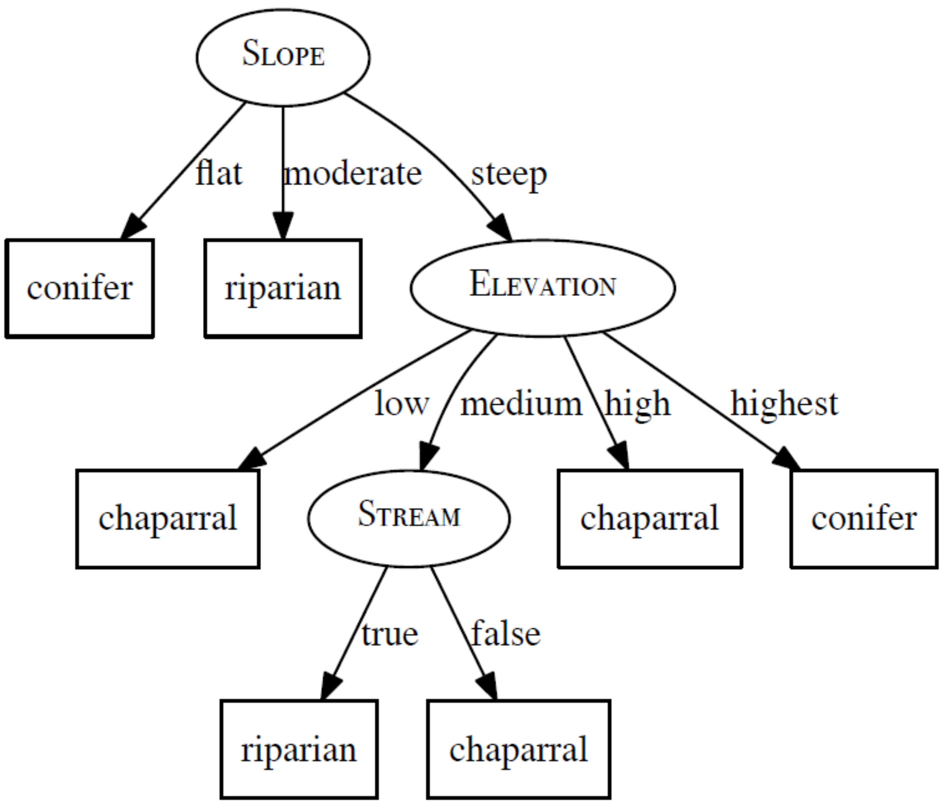


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Gini Index

- Given a training dataset *D* with target set *T*, the definition of Gini index, *GI*, is as follows:

$$GI(T,D) = 1 - \sum_{t \in T} p(t)^2$$

  - The Gini index can be thought of as calculating how often you would misclassify an instance in the dataset if you classified it based on the distribution of classifications in the dataset.
  - Information gain can be calculated using the Gini index by replacing the entropy measure with the Gini index.

Gini Index

- Example:

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$GI(\text{VEGETATION}, D) = 1 - \left(\left(\frac{3}{7}\right)^2 + \left(\frac{2}{7}\right)^2 + \left(\frac{2}{7}\right)^2\right) = 0.6531$$

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Gini Index

- Computing the information gain by Gini index

$$0.625 \times \frac{4}{7} + 0.4444 \times \frac{3}{7} = 0.5476$$

Split by Feature	Level	Part.	Instances	Partition Gini Index	Rem.	Info. Gain
STREAM	'true'	$\mathcal{D}_1$	$\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_7$	0.625	0.5476	0.1054
	'false'	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_5$	0.4444		
SLOPE	'flat'	$\mathcal{D}_3$	$\mathbf{d}_5$	0	0.4	0.2531
	'moderate'	$\mathcal{D}_4$	$\mathbf{d}_2$	0		
	'steep'	$\mathcal{D}_5$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7$	0.56		
ELEVATION	'low'	$\mathcal{D}_6$	$\mathbf{d}_2$	0	0.3333	0.3198
	'medium'	$\mathcal{D}_7$	$\mathbf{d}_3, \mathbf{d}_4$	0.5		
	'high'	$\mathcal{D}_8$	$\mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_7$	0.4444		
	'highest'	$\mathcal{D}_9$	$\mathbf{d}_6$	0		

$$0.6531 - 0.5476 = 0.1054$$

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Continuous Features

- What if ELEVATION is a continuous feature?

ID	STREAM	SLOPE	ELEVATION	VEGETATION
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
$\geq$750	$\mathcal{D}_1$	$\mathbf{d}_2$	0.0	1.2507	0.3060
	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.4591		
$\geq$1 350	$\mathcal{D}_3$	$\mathbf{d}_2, \mathbf{d}_4$	1.0	1.3728	0.1839
	$\mathcal{D}_4$	$\mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.5219		
$\geq$2 250	$\mathcal{D}_5$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3$	0.9183	0.9650	0.5917
	$\mathcal{D}_6$	$\mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.0		
$\geq$4 175	$\mathcal{D}_7$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1$	0.9710	0.6935	0.8631
	$\mathcal{D}_8$	$\mathbf{d}_5, \mathbf{d}_6$	0.0		

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Continuous Features

- Finding a threshold for a continuous feature such the information gain is maximum

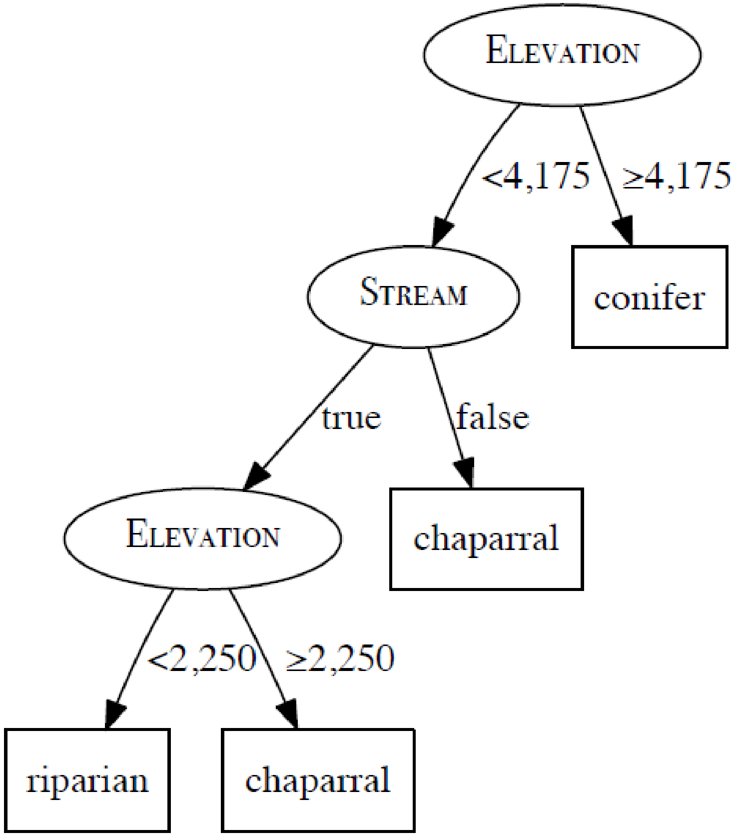


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Continuous Targets

- A dataset listing the number of bike rentals per day

ID	SEASON	WORK DAY	RENTALS
1	winter	false	800
2	winter	false	826
3	winter	true	900
4	spring	false	2 100
5	spring	true	4 740
6	spring	true	4 900

ID	SEASON	WORK DAY	RENTALS
7	summer	false	3 000
8	summer	true	5 800
9	summer	true	6 200
10	autumn	false	2 910
11	autumn	false	2 880
12	autumn	true	2 820

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Continuous Targets

- Variance, *V*

$$V(T,D) = \frac{\sum_{i=1}^{n}(t_i - \bar{t})^2}{n-1}$$

  - where *D* is the training dataset with m data instances, *T* is the target set, $T = \{t_1, t_2, \dots, t_n\}$

- Weighted variance, *U*

$$U(T,F,D) = \sum_{f \in F} \frac{\left|\left\{D_f \subseteq D \middle| \forall d \in D_f, F(d) = f\right\}\right|}{|D|} V\left(T, D_f\right)$$

  - **selecting the feature that minimizes the weighted variance**

Continuous Targets

- Computing the variance for each value of each feature

ID	Season	Work Day	Rentals
1	winter	false	800
2	winter	false	826
3	winter	true	900
4	spring	false	2 100
5	spring	true	4 740
6	spring	true	4 900

ID	Season	Work Day	Rentals
7	summer	false	3 000
8	summer	true	5 800
9	summer	true	6 200
10	autumn	false	2 910
11	autumn	false	2 880
12	autumn	true	2 820

Split by Feature	Level	Part.	Instances	$\frac{\lvert\mathcal{D}_{d=l}\rvert}{\lvert\mathcal{D}\rvert}$	$var(t, \mathcal{D})$	Weighted Variance
Season	'winter'	$\mathcal{D}_1$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.25	2 692	$1\,379\,331\frac{1}{3}$
	'spring'	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.25	$2\,472\,533\frac{1}{3}$	
	'summer'	$\mathcal{D}_3$	$\mathbf{d}_7, \mathbf{d}_8, \mathbf{d}_9$	0.25	3 040 000	
	'autumn'	$\mathcal{D}_4$	$\mathbf{d}_{10}, \mathbf{d}_{11}, \mathbf{d}_{12}$	0.25	2 100	
Work Day	'true'	$\mathcal{D}_5$	$\mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_8, \mathbf{d}_9, \mathbf{d}_{12}$	0.50	$4\,026\,346\frac{1}{3}$	$2\,551\,813\frac{1}{3}$
	'false'	$\mathcal{D}_6$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_7, \mathbf{d}_{10}, \mathbf{d}_{11}$	0.50	1 077 280	

The minimum *U*

- For D_1, SEASON = 'winter'
  - $\bar{t} = \frac{800+826+900}{3} = 842$
  - $V(T, D_1) = \frac{(800-842)^2+(826-842)^2+(900-842)^2}{3} = 2692$

Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Continuous Targets

- Selecting the feature that minimizes *U*

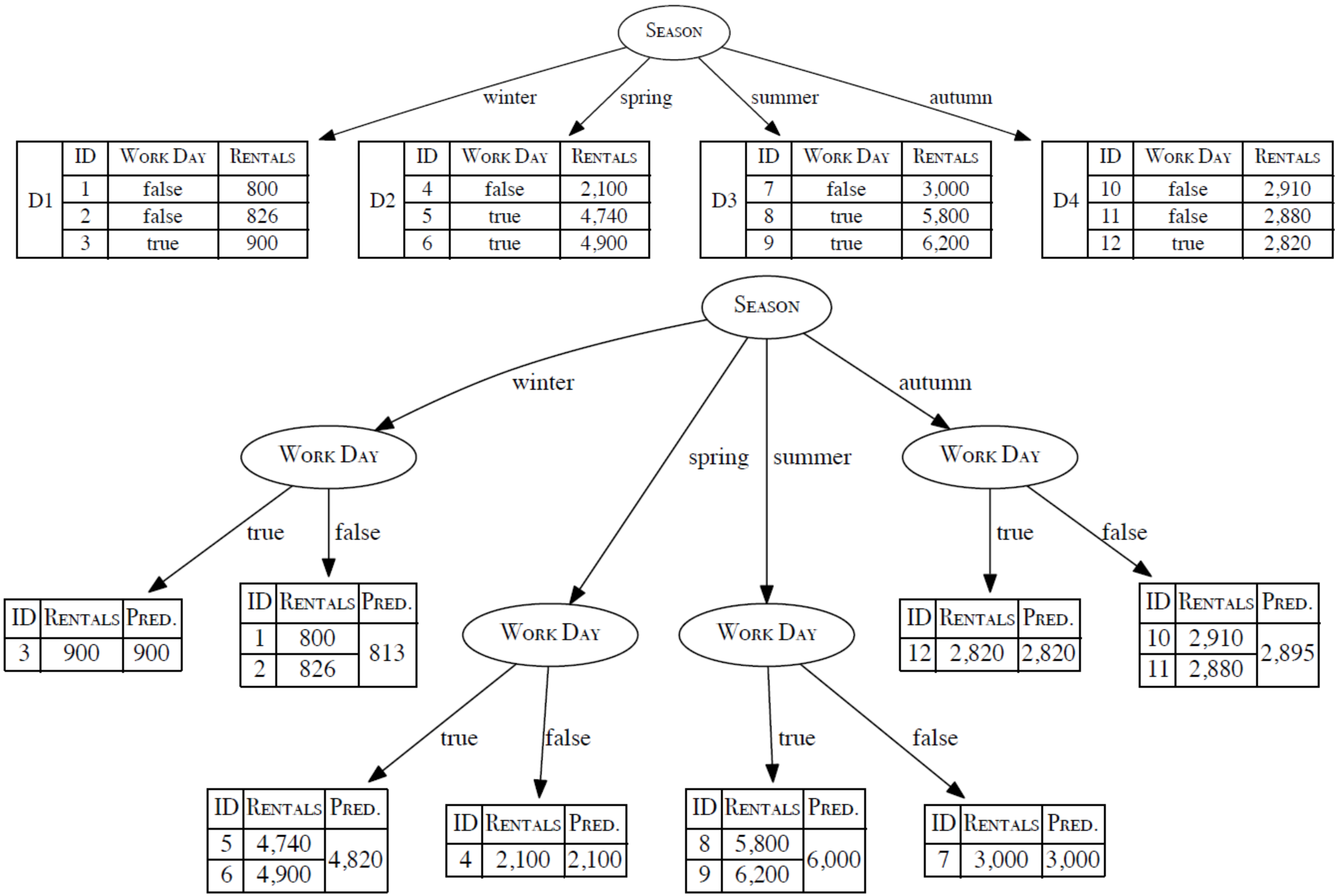


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Tree Pruning

- Overfitting in decision tree
  - A wrong splitting caused by some noisy data
  - The height of tree too high
  - An inaccurate decision tree built
- Tree pruning
  - Pre-pruning
    - Early stopping
      - If the entropy of subset is lower than a threshold ➔ stop splitting
    - χ^2 **pruning**
      - use statistical significance tests to determine the importance of subtrees
  - Post-pruning

Post-pruning

- Pruning a built decision tree by a **validation dataset**.
- Example:
  - A decision tree for the post-operative patient routing task.
    - Target: ICU(intensive care unit), GEN(general ward for recovery)

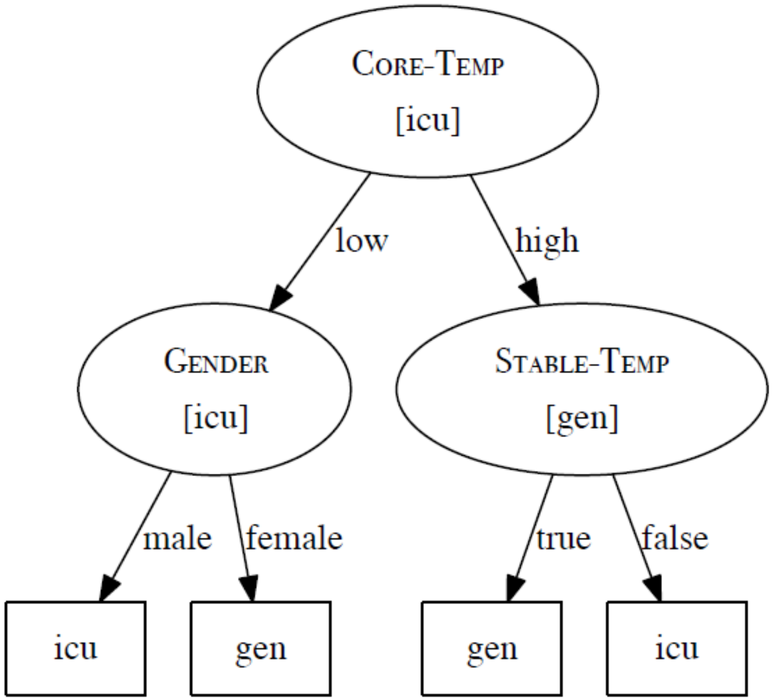


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Post-pruning

- Example: The validation data

ID	CORE-TEMP	STABLE-TEMP	GENDER	DECISION
1	high	true	male	gen
2	low	true	female	icu
3	high	false	female	icu
4	high	false	male	icu
5	low	false	female	icu
6	low	true	male	icu

- Core-Temp = Low
  - The error of Gender node: 0
    - The error of male: 0
    - The error of female: 2
  - Gender node has a child with larger error
  - ➔Remove!

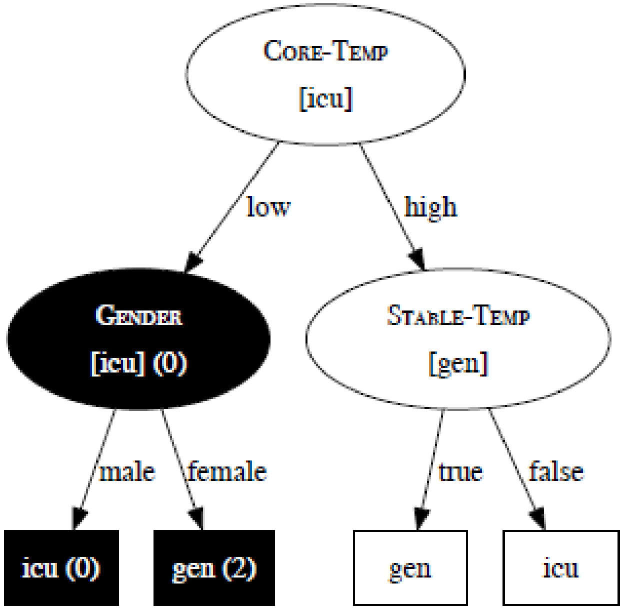


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Post-pruning

- Core-Temp = high
  - The error of Stable-Temp: 2
    - The error of true: 0
    - The error of false: 0
  - Stable-Temp node doesn't have a child with larger error
  - ➔Keep!

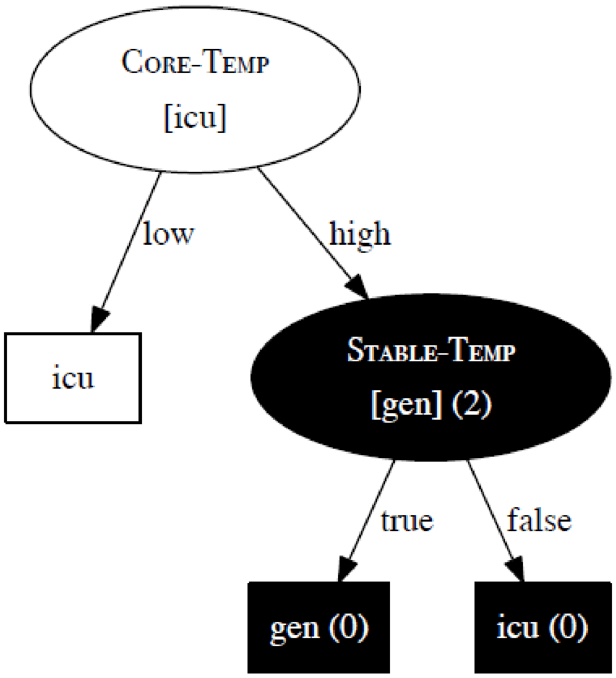


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Post-pruning

- The error of Core-Temp: 1
  - Core-Temp node doesn't have a leave child with larger error
  - ➔Keep!

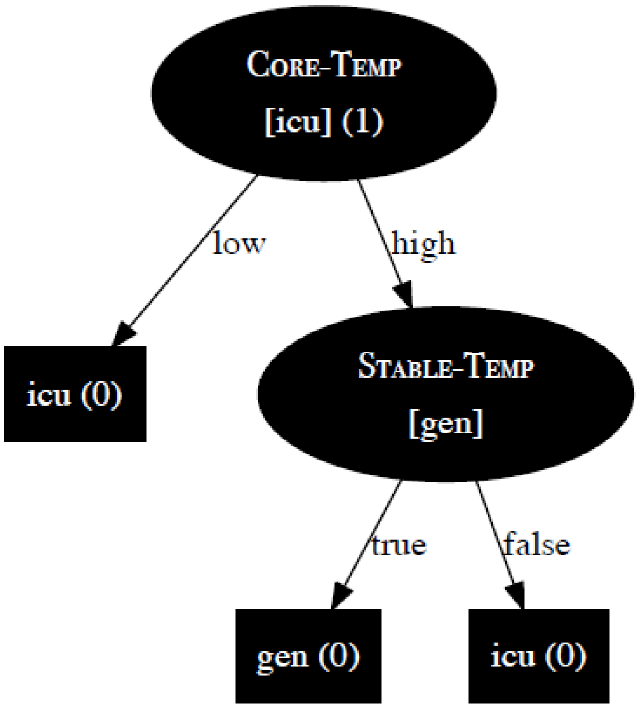


Figure from: John D. Kelleher, et al, "Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies," *MIT Press*, 2015.

Random Forest

- Random forest creation
  1. Randomly select *k* data instances from total *m* data instances, where *k* << *m*.
  2. Create a decision tree for the *k* instances.
  3. Repeat step 1 and 2 until *n* decision trees are created.

- Random forest prediction
  - Given a query *q*.
  - Each decision tree predicts a target for *q*
  - Calculate the votes for each predicted target.
  - The high voted predicted target is the final prediction

Random Forest

- Sampling with replacement
  - Randomly choose a ball from a box and then put it back to the box
- Bootstrap sample
  - For a training data set *D*
  - *D*' = Applying sampling with replacement from D, and |*D*'| = *D*
  - The fraction of the unique samples of D' is 1 – 1/e ≈ 0.632
  - Using *D*' to create a decision tree
- Out-of-bag error (OOB error)
  - The mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample.

Decision Tree in scikit-learn

- DecisionTreeClassifier
  - https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- RandomForestClassifier
  - https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html