Welcome to Online Engineering at George Washington University

Class will begin shortly

Audio: To eliminate background noise, please be sure your audio is muted. To speak, please click the hand icon at the bottom of your screen (**Raise Hand**). When instructor calls on you, click microphone icon to unmute. When you've finished speaking, ***be sure to mute yourself again***.

Chat: Please type your questions in Chat.

Recordings: Please note the recording of this class meeting will be available to download later today. The class recordings are to be used exclusively by registered students in this particular class. Releasing these recordings is strictly prohibited.

THE GEORGE
WASHINGTON
UNIVERSITY

WASHINGTON, DC

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

SEAS 8510
Analytical Methods for Machine Learning

Lecture 5

Dr. Zachary Dennis

Adapted from Rodriguez, Etemadi, Fall 2023, George Washington University

Agenda

9:00 – 10:15	Probability, Conditional Probabilty, Bayes’ Theorem
10:15 – 10:25	*BREAK (10 min)*
10:25 – 11:30	Independence, Naïve Bayes, Decision Trees
11:30 – 11:45	Homework and Discussion Look Ahead
11:45 – 12:00	Midterm Q&A

Assignments

Last week: No Homework or Discussion Due

This week: Midterm opens on 4/20 at 8 PM Eastern

Homework 4 and Discussion 4 due on 4/27 at 9 AM Eastern

Probability

How would you define probability?

Probability

- Concerns the study of uncertainty
- Fraction of times an event occurs
- Degree of belief about an event

Probability arises in two contexts:

1. In actual repeated experiments

   a) Ex: You record the color of 1,000 cars driving by and 57 of them are green. You estimate the probability of a car being green as 57/1,000 = 0.057

2. In idealized conceptions of a repeated process

   Ex. You consider flipping a coin. The expected probability of a head is 1/2 = 0.5.

   Ex. You need a model for how people’s heights are distributed. You choose a normal distribution to represent the expected relative probabilities.

Why Probability?

Solving machine learning problems requires to deal with uncertain quantities, as well as with stochastic (non-deterministic) quantities

- Probability theory provides a mathematical framework for representing and quantifying uncertain quantities

There are different sources of uncertainty:

- Inherent stochasticity in the system being modeled
- For example, most interpretations of quantum mechanics describe the dynamics of subatomic particles as being probabilistic
- Incomplete observability
- Even deterministic systems can appear stochastic when we cannot observe all of the variables that drive the behavior of the system
- Incomplete modeling
- When we use a model that must discard some of the information we have observed, the discarded information results in uncertainty in the model’s predictions
- E.g., discretization of real-numbered values, dimensionality reduction, etc.

Why Probability?

In supervised learning, want to predict something unknown (target) given something known (features). Depending on our objective, you might:

- Predict most likely value
- Predict value with smallest expected distance
- Quantify our uncertainty

In unsupervised learning, you often care about uncertainty

- Learn what's “normal” in order to detect anomalies

Random Variables

- Quantifying uncertainty requires the idea of a random variable
- A **random variable** is a function that maps outcomes of random experiments to a set of properties we are interested in
- A **probability distribution** is a description of how likely a random variable is to take on each of its possibles states
- Random Variables can be **discrete** or **continuous**

Sample Spaces and Events

- An **experiment** is any activity or process whose outcome is subject to

uncertainty

- The **sample space** is the set of all possible outcomes of the

experiment, denoted by

- The **event space** is the space of potential results of the experiment. Any collection (subset) of outcomes contained in the sample space ***S***
    - *An event is simple if it consists of exactly one outcome, it is compound if it consists of more than one outcome*

Example

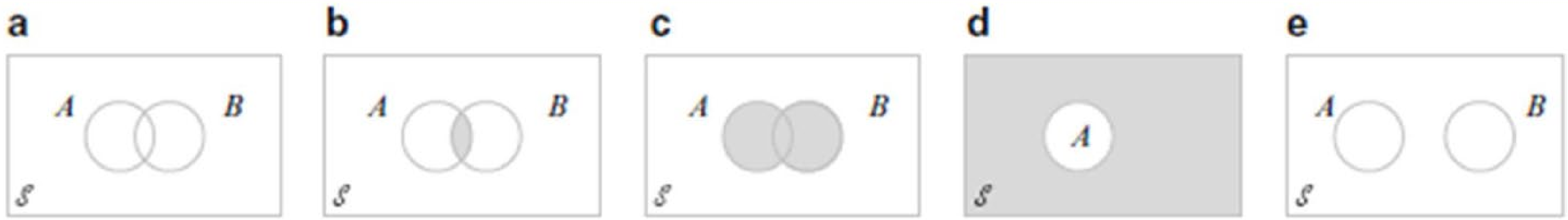
- **Experiment:**
  - Rolling an unbiased 6-sided die
- **Random Variable:**
  - # on the die facing up
- **Sample Space**
  - {1 , 2, 3, 4, 5, 6}
- **Probability of rolling a 6:**
  - P(6) = 1 / 6

Set Theory Related to Events

Definitions

- **Union** of two events A and B – denoted by A U B
  - A “or” B
- **Intersection** of two events A and B – denoted by A∩B
  - A “and” B
- **Complement** of an event A – denoted by A’ is the set of all outcomes in the sample space that are not contained in A
- When A and B have no outcomes in common, they are said to be disjoint or **mutually exclusive** events

Set Theory



a: Venn diagram of events A and B

b: Shaded region is $A \cap B$

c: Shaded region is $A \cup B$

d: Shaded region is A'

e: Mutually exclusive events

Axioms of Probability

Given an experiment with sample space $\mathcal{S}$, the objective of probability is to assign to each event A a number P(A), called the probability of event A, which will give a precise measure of the chance that A will occur.

To ensure that the probability assignments will be consistent with our intuitive notions of probability, all assignments should satisfy the following axioms (basic properties) of probability

Axiom 1 For any event A, P(A) >= 0

Axiom 2 P($\mathcal{S}$) = 1

Axiom 3 If A1, A2, A3,…. Is an infinite collection of disjoint events, then

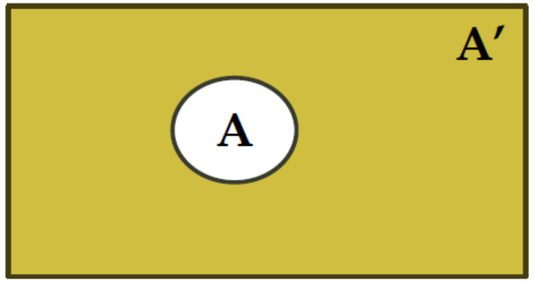
$$P(A_1 \cup \; A_2 \cup \; A_3 \ldots) = \sum_{i=1}^{\infty} P(A_i)$$

Probability Properties

Proposition

For any event A, $P(A) + P(A') = 1$, from which

$P(A) = 1 - P(A')$



- Useful because there are many situations where P(A') is more easily obtained than P(A)

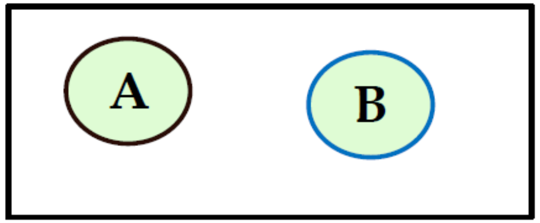
Probability Properties

Proposition

For any event A, $P(A) \leq 1$

When two events A and B are mutually exclusive,

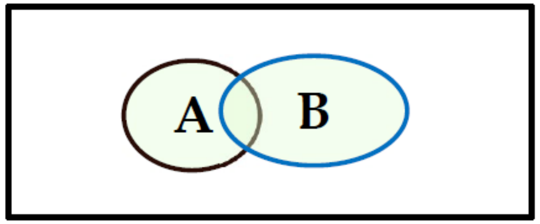
$$P(A \cup B) = \mathrm{P(A)} + \mathrm{P(B)}$$



Proposition

For any two events A and B

$$P(A \cup B) = \mathrm{P(A)} + \mathrm{P(B)} - \mathrm{P}((A \cap B)$$



Conditional Probability

The probabilities assigned to various events depend on what is known about the experimental situation when the assignment is made.

Subsequent to the initial assignment, partial information relevant to the outcome of the experiment may become available. Such information may cause us to revise some of our probability assignments.

For a particular event A we have used P(A) to represent the probability, assigned to A ; we now think of P(A) as the original, or unconditional probability, of the event A

Conditional Probability

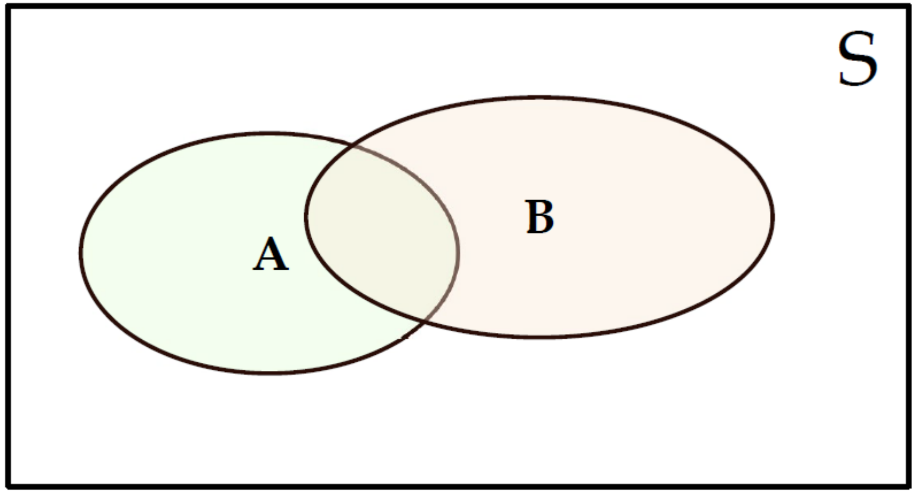
Now we’ll examine how the information “event B has occurred” affects the probability assigned to A.

We will use the notation **P(A|B)** to represent the **conditional probability of A given that the event B has occurred**. B is the “conditioning event”

What is the complement to P(A|B)?

Conditional Probability

Think of conditioning on B as redefining the sample space from S to B



Given that B has occurred, the relevant sample space is not longer S, but consists of outcomes in B. Event A has occurred if and only if one of the outcomes in the intersection occurred

Definition of Conditional Probability

Definition

For any two events A and B with P(B) > 0, the conditional probability of A given the B has occurred is defined by

$$P(A|B) = \frac{P\ (A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P\ (A \cap B)}{P(A)}$$

Note that in the first, P(B)>=0

In the second, P(A)>=0

Conditional Probability Example

Components are assembled in a plant that uses 2 different assembly lines: A and A’

Components are either Defective (B) or Nondefective (B’)

	Condition	
Line	**B**	**B'**
A	2	6
A'	1	9

P(A) =

P(B) =

P(A and B) =

P(A given B) =

$$P(A|B) = \frac{P\ (A \cap B)}{P(B)}$$

Conditional Probability Example

Suppose that of all individuals buying a certain digital camera:

- 60% include an optional memory card in their purchase
- 40% include an extra battery
- 30% include both a card and battery

Consider randomly selecting a buyer and let A = [memory card purchased] and B=[battery purchased]

Given that the selected buyer purchased an extra battery, what is the probability an optional memory card was purchased?

Given that the selected buyer purchased an optional memory card, what is the probability they purchased an extra battery?

Multiplication Rule

Since,

$$P(A|B) = \frac{P\ (A \cap B)}{P(B)}$$

$$P\ (A \cap B) = \mathrm{P(A|B)} * \mathrm{P(B)}$$

This rule is important because it is often the case that $P(A \cap B)$ is desired, whereas both $P(B)$ and P(A|B) can be specified from the problem description.

Multiplication Rule Example

Four individuals have responded to a request by a blood bank for blood donations. None of them has donated before, so their blood types are unknown.

Suppose only type O+ is desired and only one of the four actually has this type. If the potential donors are selected in random order for typing, **what is the probability that at least three individuals must be typed to obtain the desired type?**

Expanding the Multiplication Rule

The multiplication rule can be expanded to more than 2 events.

For example,

$$
\begin{aligned}
P\,(A_1 \cap A_2 \cap A_3) &= \mathrm{P}(A_3|A_2 \cap A_1) * \mathrm{P}(A_1 \cap A_2) \\
&= \mathrm{P}(A_3|A_2 \cap A_1) \ast \mathrm{P}(A_2|A_1) \ast \mathrm{P}(A_1)
\end{aligned}
$$

Where A1 occurs first, followed by A2, and finally A3

Bayes' Theorem

$$\mathrm{P(A|B)} = \frac{P(B|A)P(A)}{P(B)}$$

The terms are referred to as:

P(A), the prior probability, initial degree of belief for A

P(A|B), the posterior probability, the degree of belief after incorporating the knowledge of B

P(B|A), the likelihood of B given A

P(B), the evidence

$$Posterior\ Probability = \frac{likelihood * prior\ probability}{evidence}$$

Bayes’ Theorem

Let A_1, ….., A_k be a collection of mutually exclusive and exhaustive events with $P(A_i)>0$ for i=1…., k. Then for any other event B, for which P(B) > 0

$$\boxed{P(A_j \mid B) = \frac{P(A_j \cap B)}{P(B)} = \frac{P(B \mid A_j)P(A_j)}{\sum_{i=1}^{k} P(B \mid A_i)P(A_i)} \quad j = 1, \ldots, k}$$

Bayes' Theorem Example

Incidence of a rare disease: Only 1 in 1000 adults is afflicted with a rare disease for which a diagnostic test has been developed. The test is such that when an individual actually has the disease, a positive result will occur 99% of the time, whereas an individual without the disease will show a positive test result only 2% of the time. **If a randomly selected individual is tested and the result is positive, what is the probability that the individual has the disease?**

Break

Independence

In our examples, it was frequently the case that $P(A|B)$ differed from the unconditional probability $P(A)$, indicating that the information "B has occurred" resulted in a change in the chance of A occurring. Often the chance that A will occur or has occurred is not affected by knowledge that B has occurred

Definition

Two events A and B are independent if $P(A|B) = P(A)$ and are dependent otherwise.

If A and B are independent, then so are the following pairs of events:

- A' and B
- A and B'
- A' and B'

Naïve Bayes

Family of classification algorithms

- Set of features (X1, X2, X3, …. Xn)
- Predicting Class Y

Naïve Bayes “naively” assumes all features are independent.

Algorithm Steps:

- Create frequency tables
- Create likelihood tables
- Calculate posterior probability using Bayes’ theorem
- Predicts higher probability

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Decision Trees

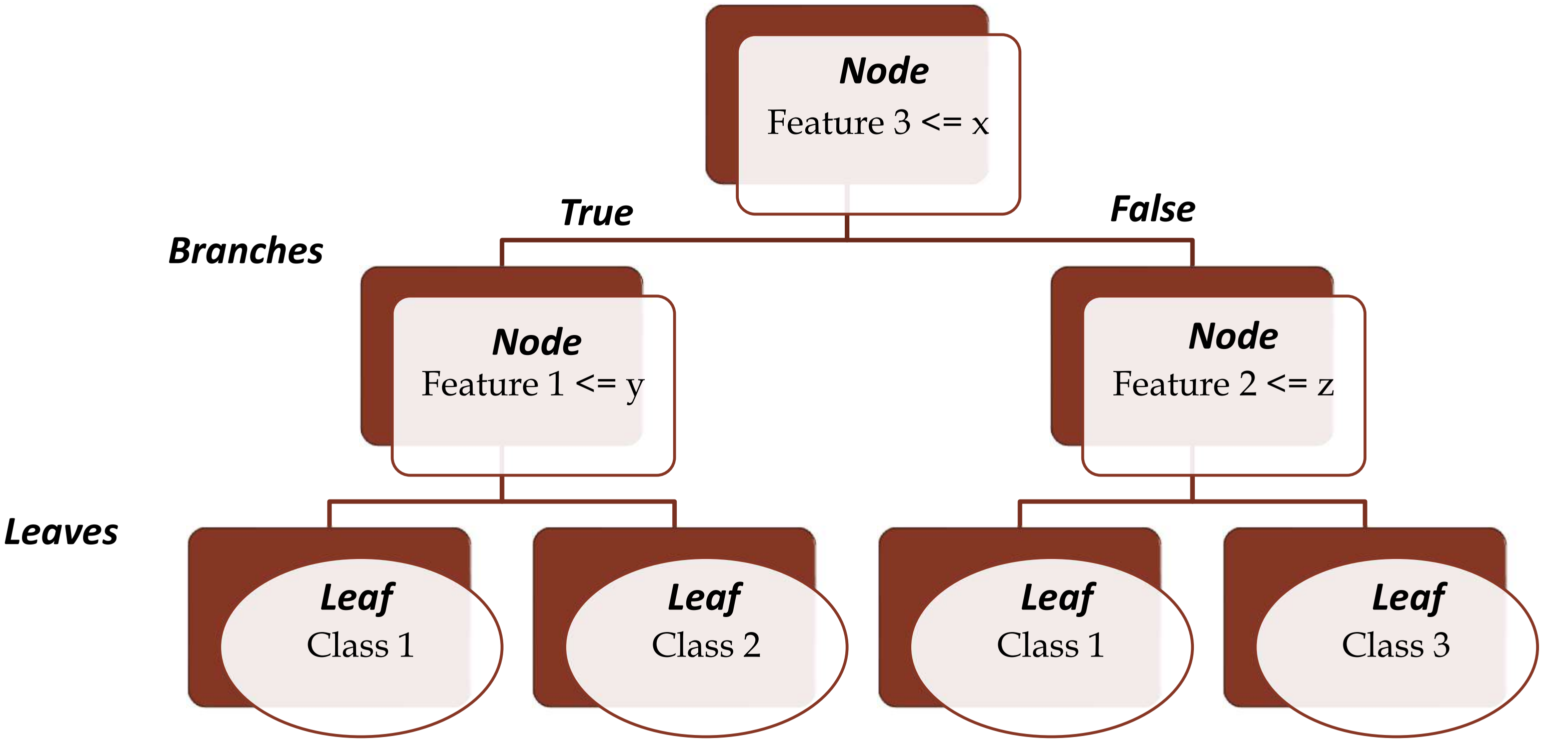
Decision Trees

- Decision Trees are versatile Machine Learning algorithms that can perform both classification and regression tests, and even multioutput tasks
- Powerful algorithms capable of fitting complex data sets
- Decision Trees are also the fundamental components of Random Forests, which are among the most powerful Machine Learning algorithms available today.

(Geron, 2019, p. 175)

Decision Tree Terminology

- Tree-shaped diagram
- Each branch represents a possible decision, occurrence or reaction



Construction of Decision Trees

- **Top Down Approach:**
  - Evaluate each attribute for usefulness classifying
  - Select best attribute as root of the tree
  - Split based on attribute to produce a subset of data
  - Repeat on subsets to find next nodes considering only attributes that have not been selected already

- **How do you decide which attribute is most useful?**

Attribute Selection Measures

- **How do you decide which attribute is most useful?**
  - Entropy
  - Information Gain
  - Gini Index

- **Common Algorithms and their Methods:**
  - ID3 (Iterative Dichotomizer 3) – Entropy, Information Gain
  - C4.5 (Successor of ID3) – Entropy, Information Gain
  - CART (Classification and Regression) – Gini Index

Entropy

- Entropy is a measure of impurity or randomness

- Node is pure when value 0 or 1

$$E(S) = \sum_{i=1}^{n} -p_i log_2 p_i$$

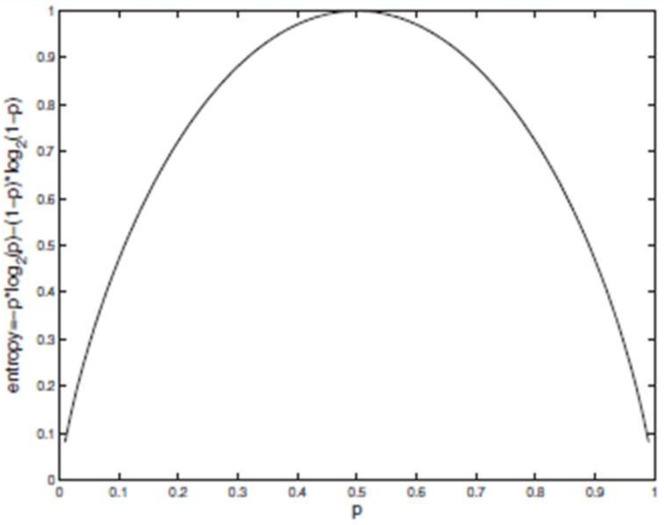


Figure 9.2 Entropy function for a two-class problem.

  - S is the current state
  - pi is the probability of an event i of state S or percentage of class i in a node of state S

(Alpaydin, 2014, p. 217)

Information Gain

- How much information an attribute provides for the target variable

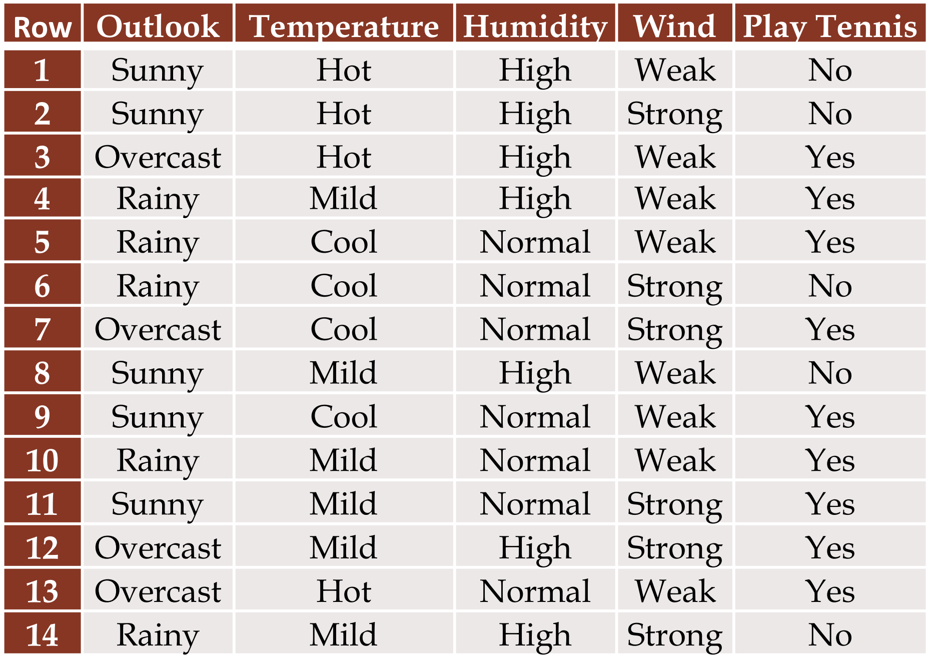
$$Information\ Gain = Entropy_{parent} - \ Entropy_{children}$$

$$Info\ Gain\ (A; B) = E(A) - \sum_{b} P(B = b) * E(A|B = b)$$

ID3 Decision Tree Example - Data

Row	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

ID3 Decision Tree Example



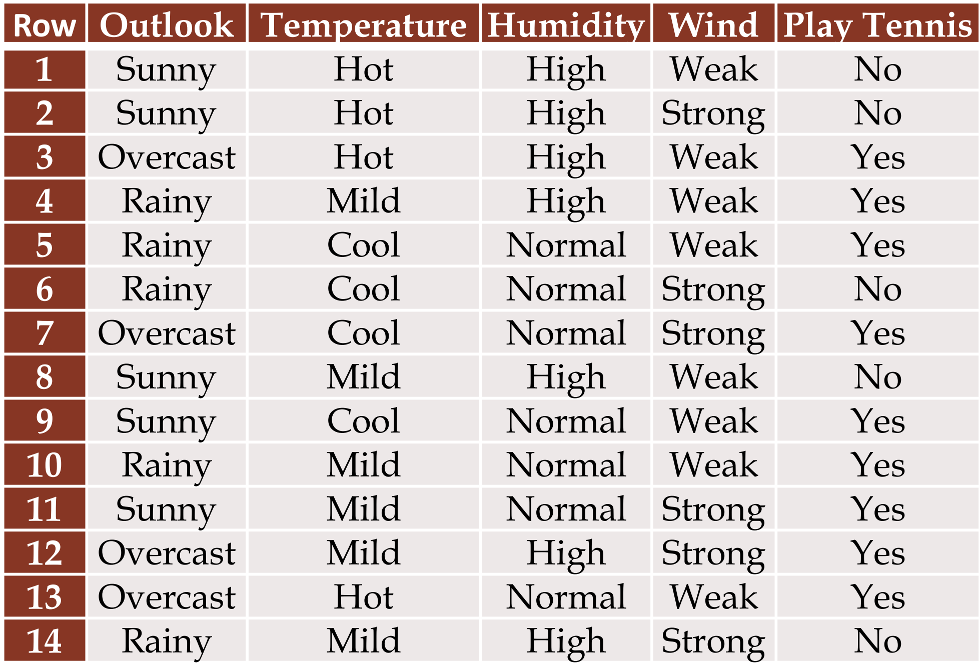
Row	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

Steps:

1. Calculate Entropy for Target Variable
2. Calculate Entropy for Each Attribute
3. Calculate Information Gain for Each Attribute
4. Select Attribute with Highest Information Gain for Root Node
5. Split and continue….

- E(Temperature)=

ID3 Decision Tree Example



Row	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

Steps:

1. Calculate Entropy for Target Variable

Tennis = Yes:

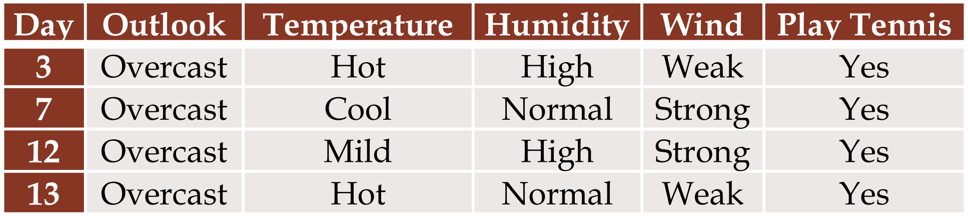
Tennis = No:

- E(Play Tennis) =

ID3 Decision Tree Example

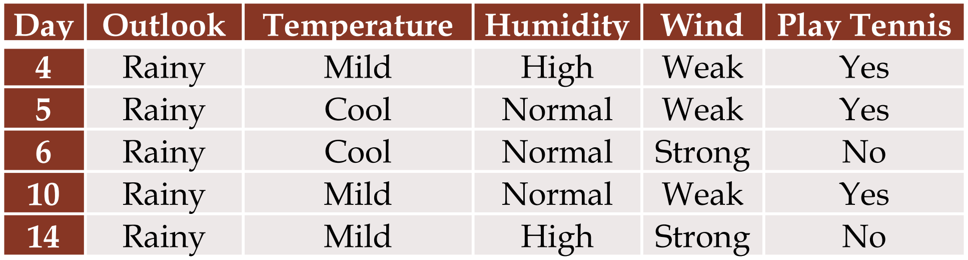
Steps:

2. Calculate Entropy for Each Attribute



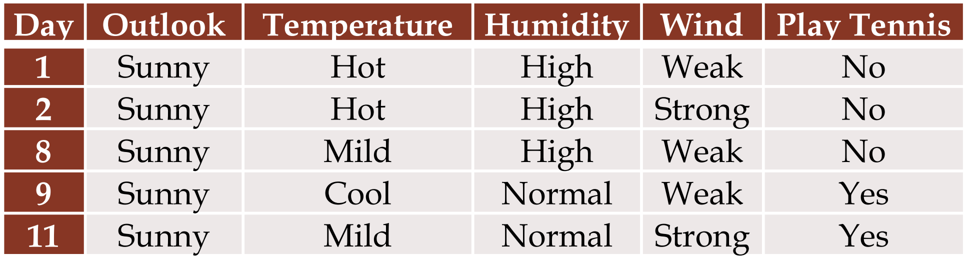
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

E(Play|Outlook= Overcast) =



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
10	Rainy	Mild	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

E(Play|Outlook= Rainy) =



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

E(Play|Outlook= Sunny) =

ID3 Decision Tree Example

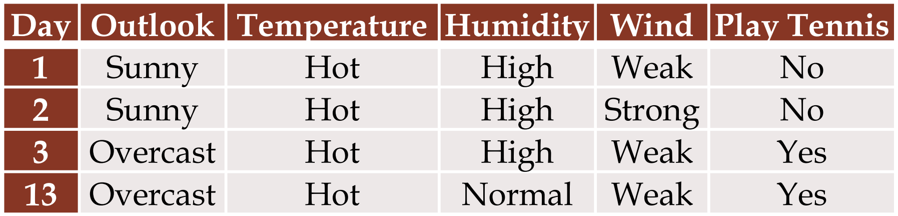
Steps:

2. Calculate Entropy for Each Attribute

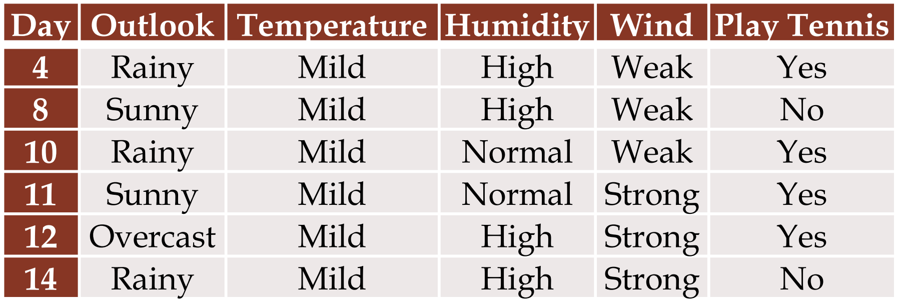
E(Play|Temp = Hot) =

E(Play|Temp = Mild) =

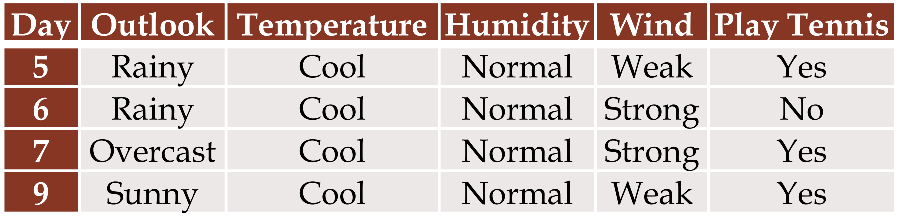
E(Play|Temp = Cool) =



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
4	Rainy	Mild	High	Weak	Yes
8	Sunny	Mild	High	Weak	No
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rainy	Mild	High	Strong	No



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
9	Sunny	Cool	Normal	Weak	Yes

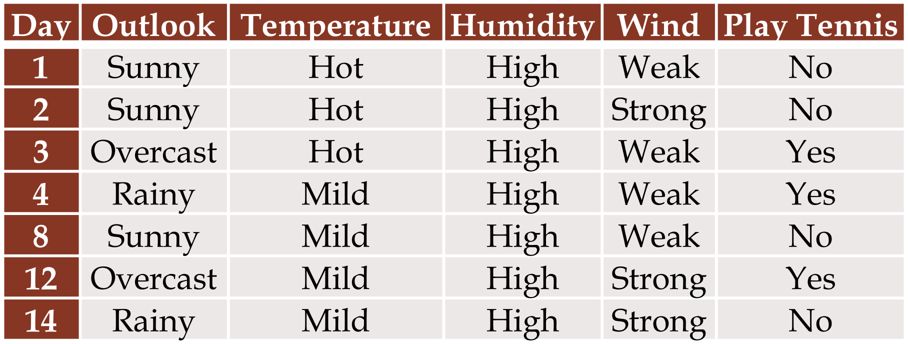
ID3 Decision Tree Example

Steps:

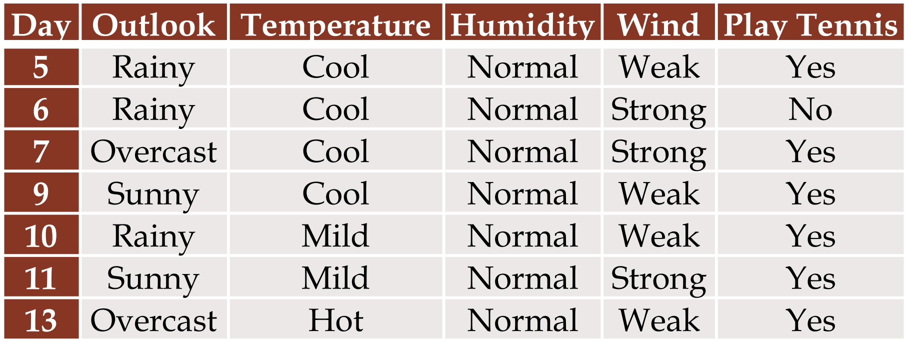
2. Calculate Entropy for Each Attribute

E(Play|Humidity=High) =

E(Play|Humidity=Normal) =



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
8	Sunny	Mild	High	Weak	No
12	Overcast	Mild	High	Strong	Yes
14	Rainy	Mild	High	Strong	No



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

ID3 Decision Tree Example

Steps:

2. Calculate Entropy for Each Attribute

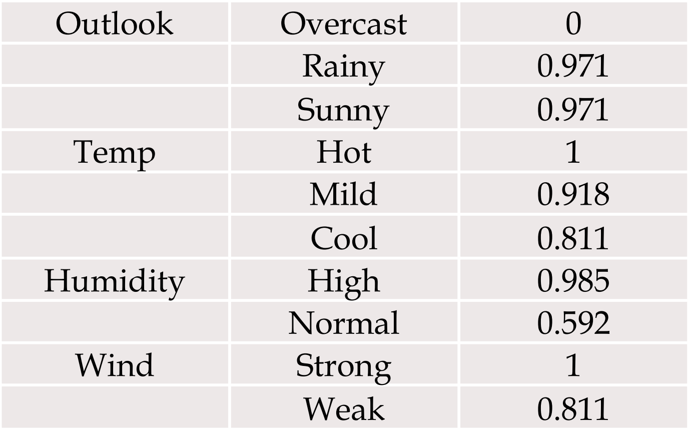
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
2	Sunny	Hot	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rainy	Mild	High	Strong	No

E(Play|Wind=Strong) =

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

E(Play|Wind=Weak) =

ID3 Decision Tree Example



Outlook	Overcast	0
	Rainy	0.971
	Sunny	0.971
Temp	Hot	1
	Mild	0.918
	Cool	0.811
Humidity	High	0.985
	Normal	0.592
Wind	Strong	1
	Weak	0.811

Steps:

3. Calculate Information Gain for Each Attribute
    - **IG (Outlook)** = E(Play Tennis) - [P(Outlook = Sunny* E(Play Tennis | Outlook = Sunny) +P(Outlook=Overcast * E(Play Tennis | Outlook = Overcast) + P(Outlook =Rainy)* E(Play Tennis | Outlook = Rainy)
    - **IG (Temperature)=**

ID3 Decision Tree Example

Steps:

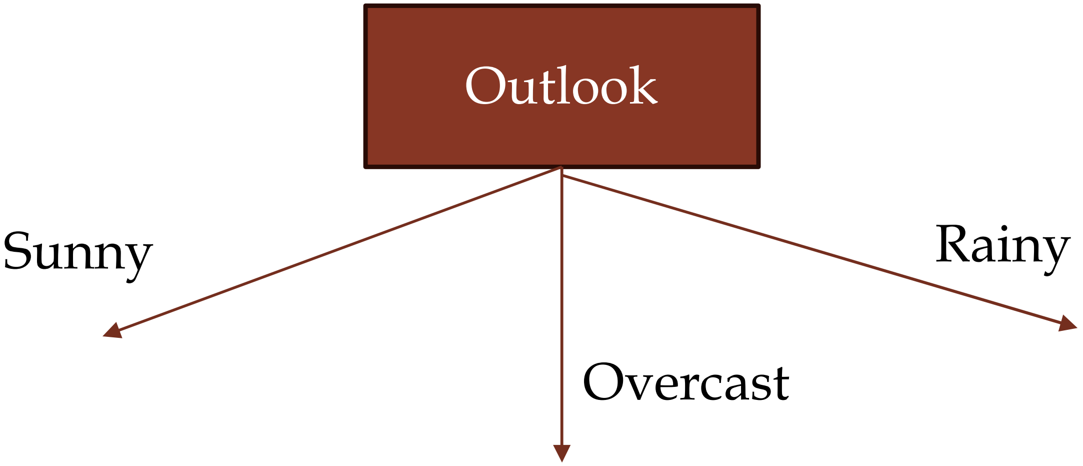
3. Calculate Information Gain for Each Attribute
   - **IG (Humidity) =**
   - **IG (Wind) =**

ID3 Decision Tree Example

Outlook	0.246
Temp	0.029
Humidity	0.1515
Wind	0.048

Steps:

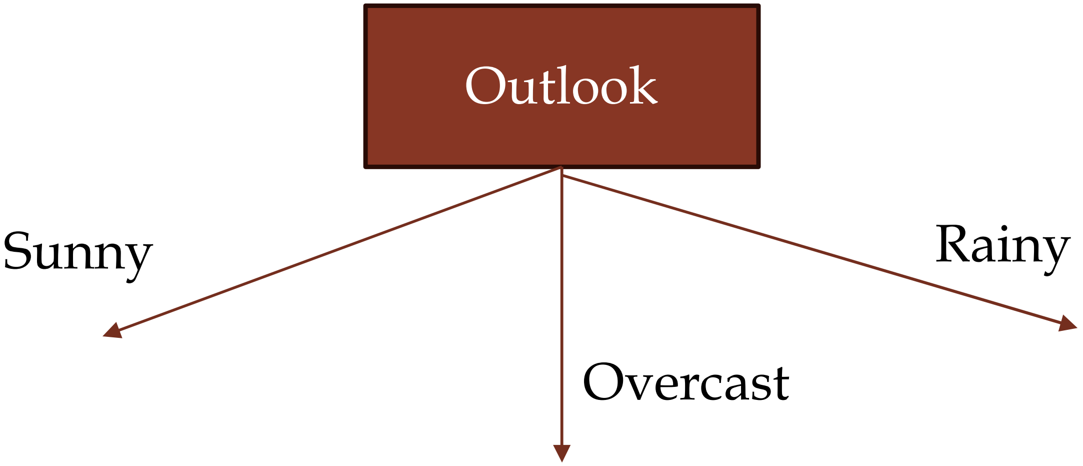
4. Select Attribute with Highest Information Gain for Root Node
5. Split and continue….



ID3 Decision Tree Example

Steps:

4. Select Attribute with Highest Information Gain for Root Node
5. Split and continue….



Outlook

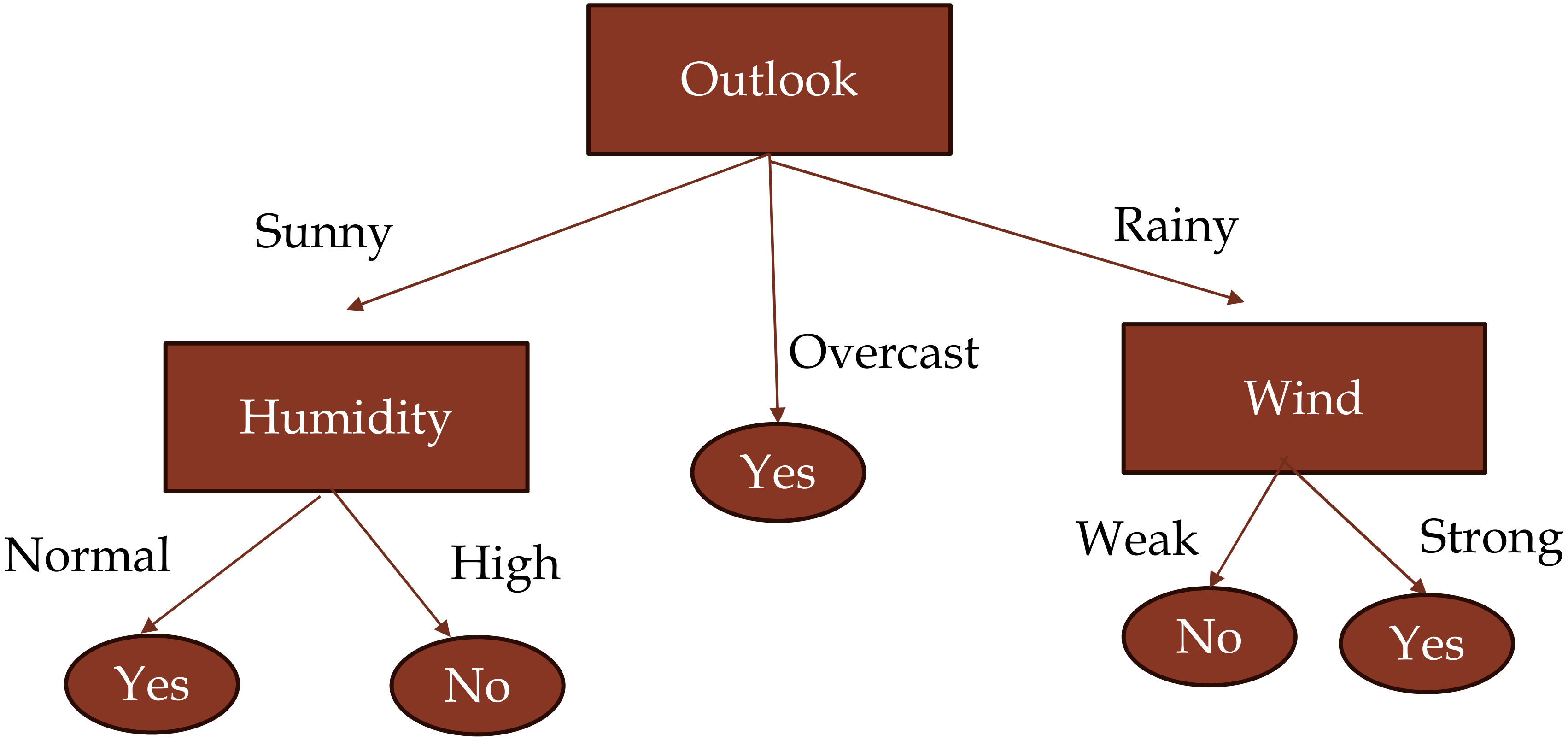
Sunny

Overcast

Rainy

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

ID3 Decision Tree Example



Using Decision Trees to Make Predictions

Start at the root node (depth 0, at the top)

Is the petal length of the flower smaller than 2.45 cm?

- If it is, then you move down to the left (depth 1, left). This is a leaf node and the decision tree predicts that your flower is an *Iris Setosa*

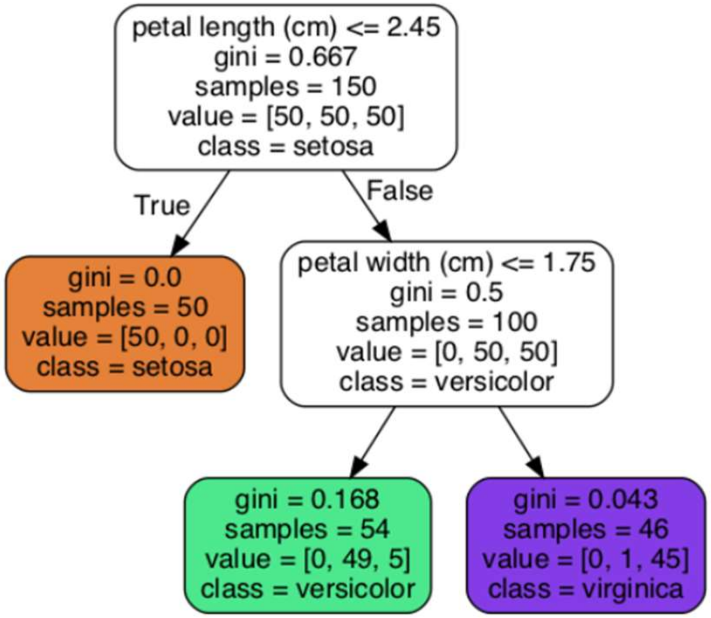


Figure 6-1. Iris Decision Tree

If it is greater than 2.45 cm…

Move down to the right to child node (depth 1, right)

Is the petal width smaller than 1.75 cm?

- Yes > then your flower is most likely an *Iris Versicolor* (depth 2, left)
- No > then your flower is likely *Iris Virginica* (depth 2, right)

(Geron, 2019, p. 176)

Using Decision Trees to Make Predictions

- A node's **samples** attribute is how many training instances it applies to

- How many training instances have petal length of more than 2.45 cm?

  100

- A node's **value** attribute tells you how many training instances of each class this node applies to
  - This node (petal length greater than 2.45 cm and width less than 1.75 cm) applies to: 0 setosa, 49 versicolor, and 5 virginica

- A node's **gini** attribute measures its impurity
  - A node is "pure" and has gini = 0 when all training instances it applies to belong to the same class

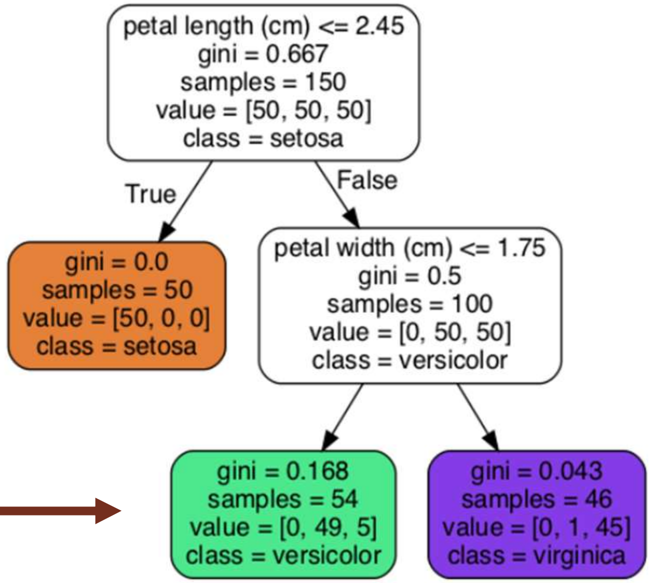


Figure 6-1. Iris Decision Tree

(Geron, 2019, p. 177)

Gini Impurity

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

- $p_{i,k}$ is the ratio of class *k* instances among the training instances in the *ith* node
- Gini = 0 when the node is “pure” and all training instances it applies to belong to the same class

(Geron, 2019, p. 177)

Gini Impurity Examples

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

- $p_{i,k}$ is the ratio of class *k* instances among the training instances in the *ith* node

What is the gini score of node 0?

What is the gini score of depth-1 left node?

What is the gini score of depth-2 right node?

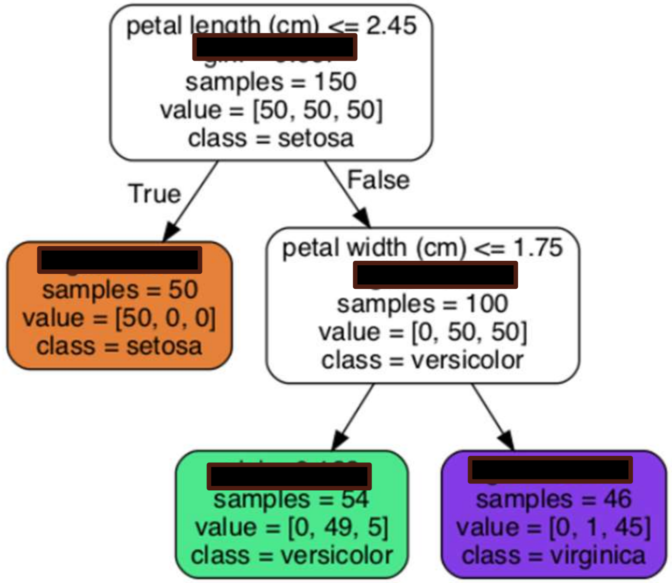


Figure 6-1. Iris Decision Tree

(Geron, 2019, p. 177)

Estimating Class Probabilities

A Decision Tree can also estimate the probability that an instance belongs to a particular class k

First it traverses the tree to find the leaf node for this instance, and then it returns the ratio of training instances of class k in this node.

If you ask it to predict the class, it outputs the class with the highest probability

Example instance: Petal length = 5 cm, Petal width = 1.5 cm

- What is probability the flower is Iris Setosa?

= 0 / 54 = 0%

- Iris Versicolor?

= 49 / 54 = 90.7%

- Iris Virginica?

= 5 / 54 = 9.3%

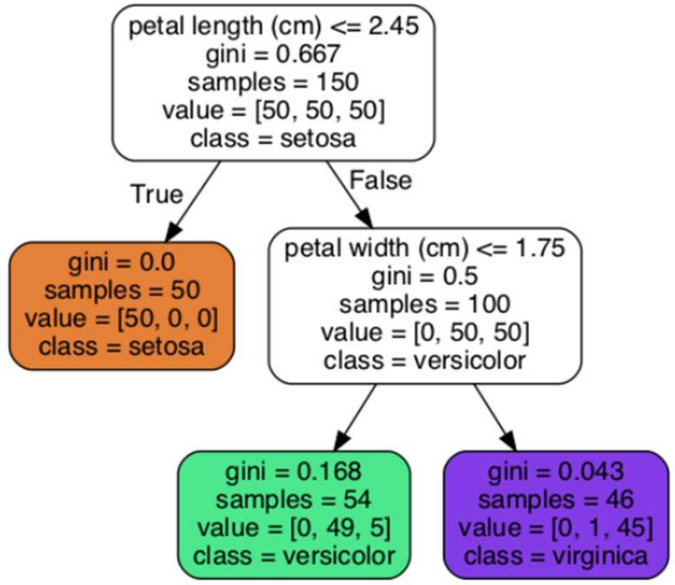
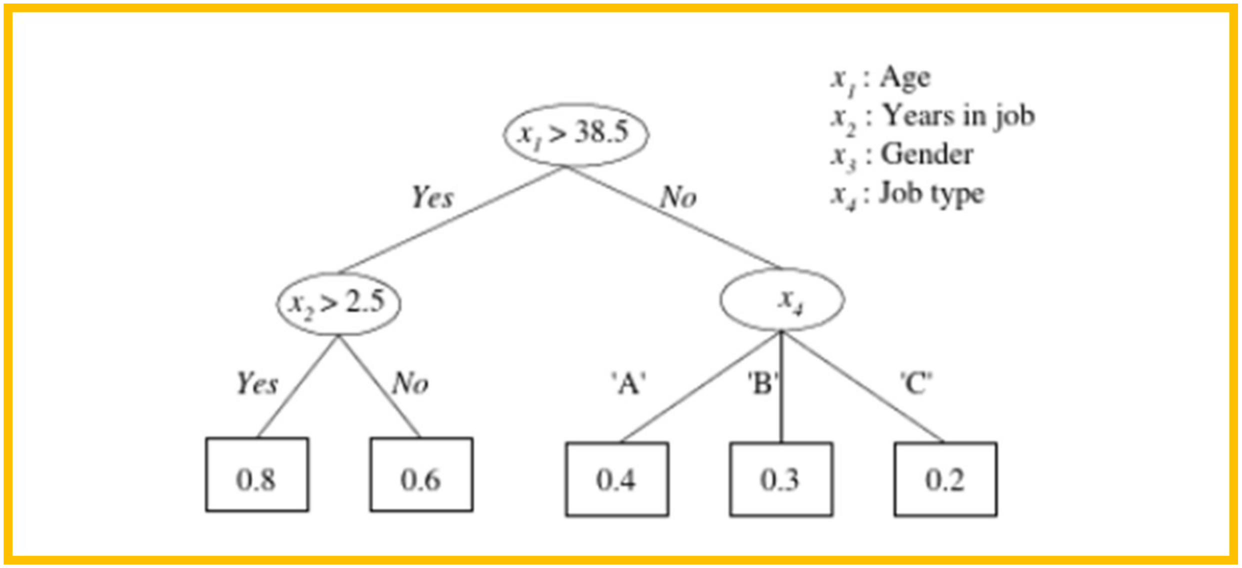


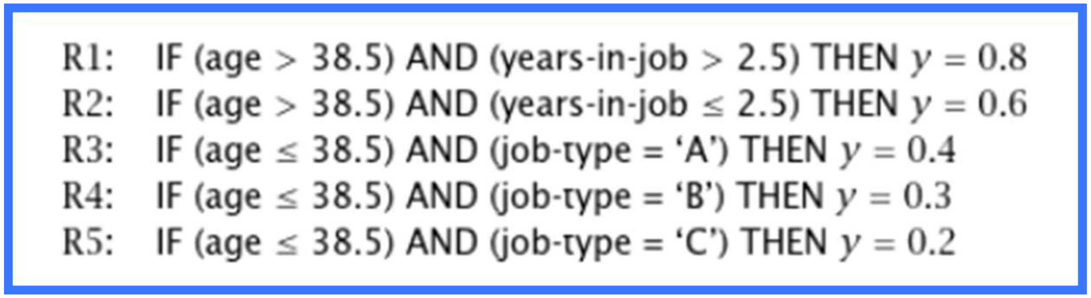
Figure 6-1. Iris Decision Tree

(Geron, 2019, p. 178)

Rule Extraction from Trees

- The decision tree can be converted to IF-THEN rules by tracing the path from the root node to each leaf node in the tree




R1: IF (age > 38.5) AND (years-in-job > 2.5) THEN $y = 0.8$
R2: IF (age > 38.5) AND (years-in-job ≤ 2.5) THEN $y = 0.6$
R3: IF (age ≤ 38.5) AND (job-type = 'A') THEN $y = 0.4$
R4: IF (age ≤ 38.5) AND (job-type = 'B') THEN $y = 0.3$
R5: IF (age ≤ 38.5) AND (job-type = 'C') THEN $y = 0.2$

(Alpaydin, 2014, p. 225)

Regularization Hyperparameters

Decision Trees:

- Make very few assumptions about the training data
- Tend to overfit if they are not constrained
  - The tree structure will adapt itself to the training data, fitting it very closely
- To avoid overfitting to the training data, you need to restrict the decision tree's freedom during training (i.e. regularization)
- The regularization hyperparameters depend on the algorithm used, but generally you can at least restrict the maximum depth of the decision tree
  - Default max_depth = None allows for unlimited depth. Reducing max_depth will regularize the model and reduce the risk of overfitting

(Geron, 2019, p. 181)

Resources

- Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for Machine Learning.
- Modern Mathematical Statistics with Applications Second Edition by Jay L. Devore and Kenneth N. Berk
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, Aurelien Geron, 2019
- Introduction to Machine Learning (2014) by Ethem Alpayadin