Classification:
Decision Tree & Naïve Bayes

Spring 2018

Review

- Last week:
  - Regression Applications
  - Linear Regression
  - Polynomial Regression
  - Regularization (Ridge Regression and Lasso Regression)
  - Evaluating Regression Models

- Assignments (Canvas)
  - Lab assignment due yesterday
  - New problem set out and due next week

- Questions?

Today’s Topics

- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

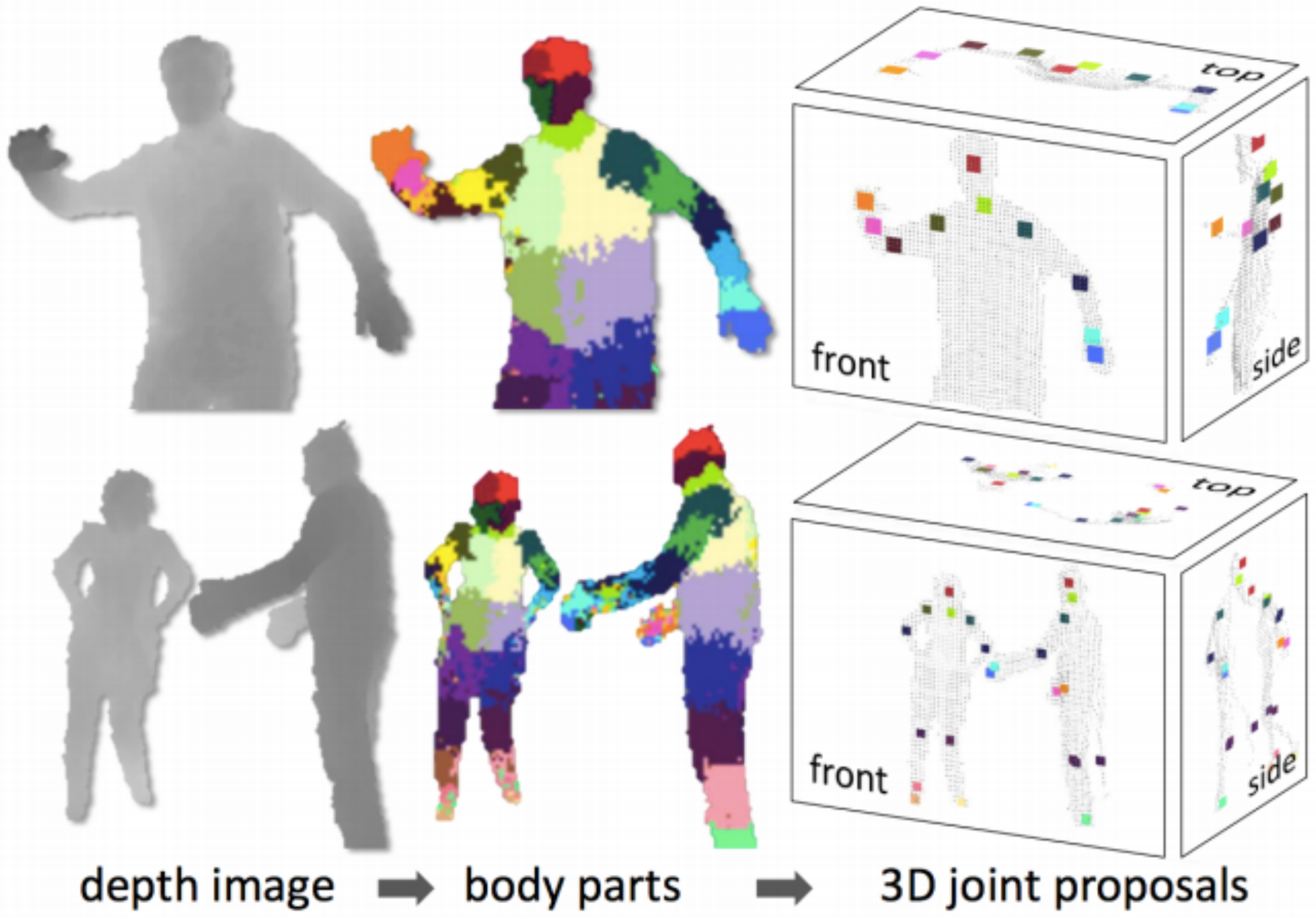
Today’s Topics

- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

Today’s Focus: Classification

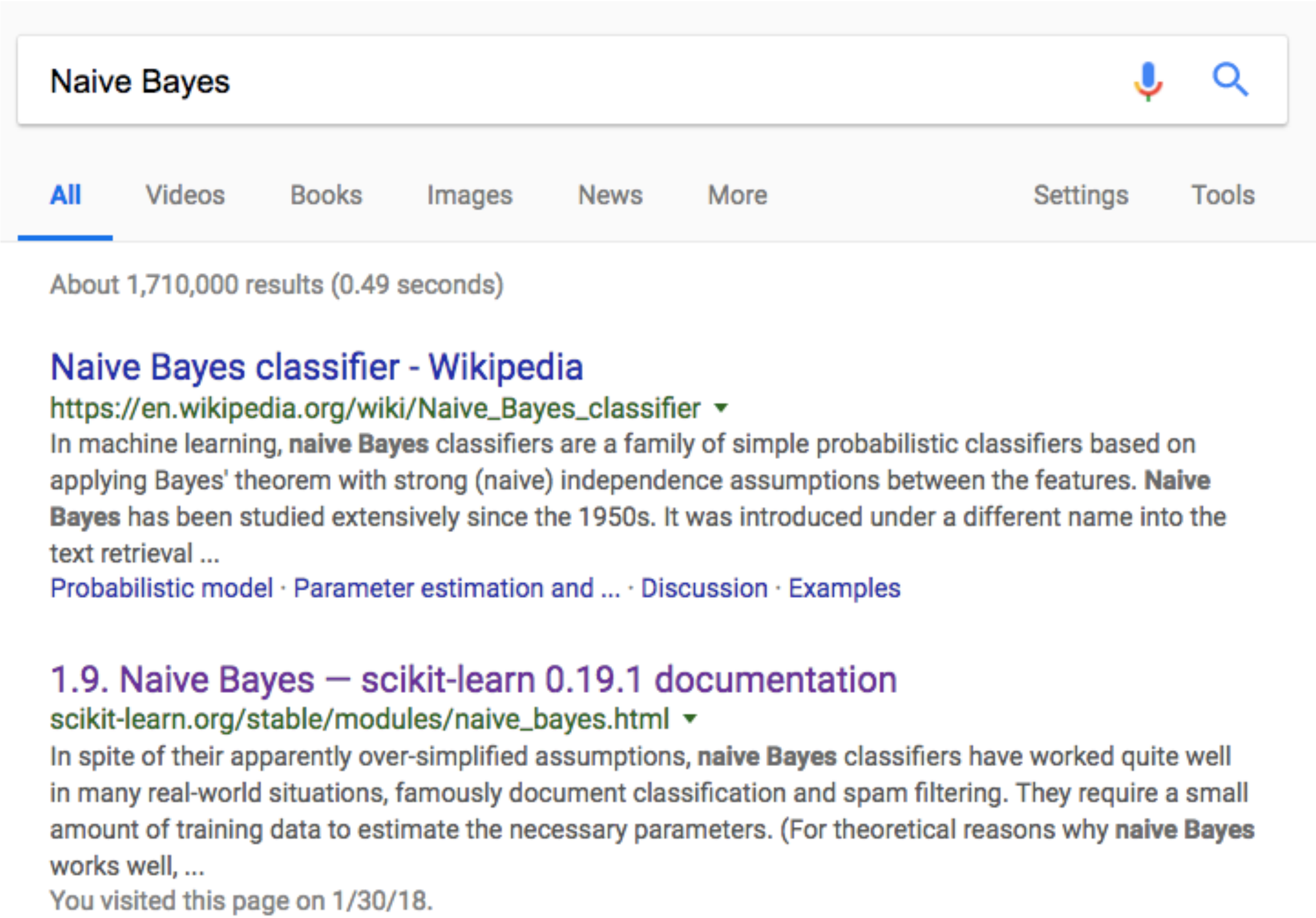
Predict **discrete** value

Entertainment: Classifying Body Parts in XBox

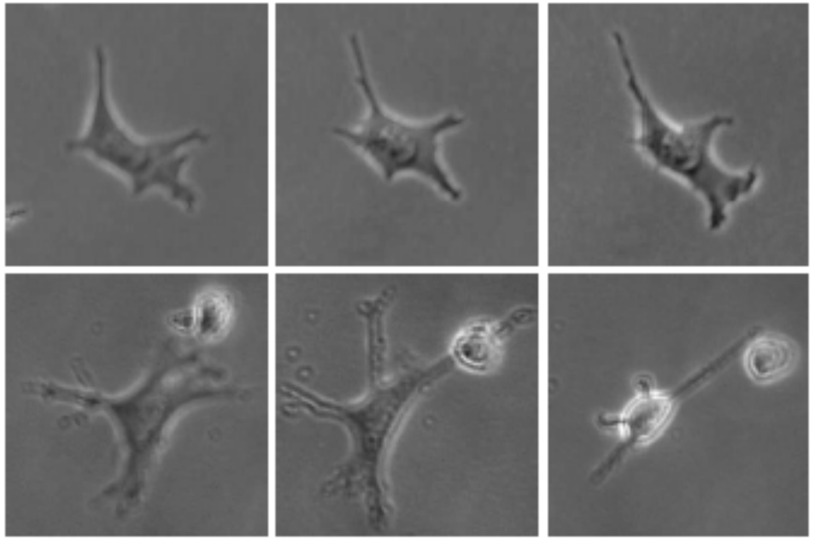


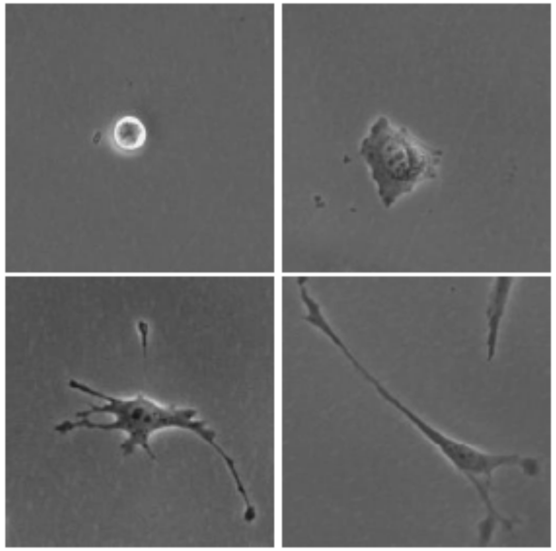
Slide adapted from Sanja Fidler

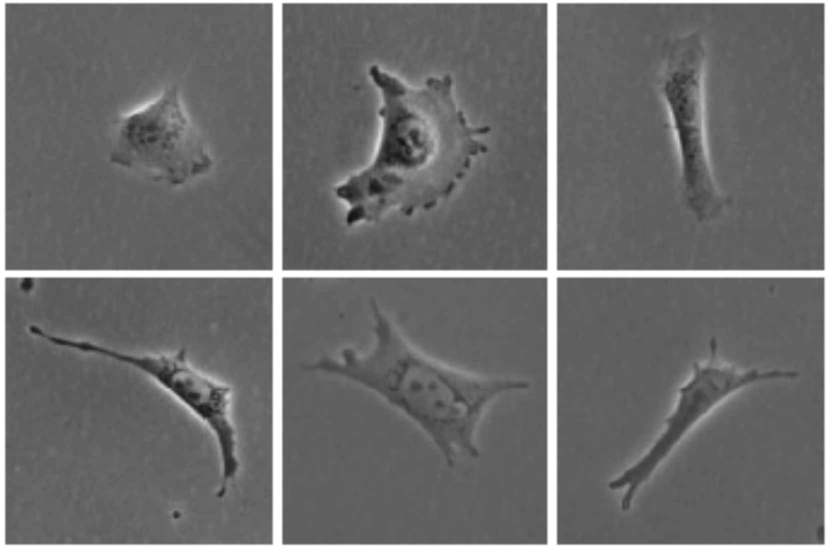
Information Retrieval: Relevant Document or Not?



Biology: Classify Cell Shapes for Long Term Goal of Biomaterial Creation







Theriault et al; Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning; 2012.

Today’s Topics

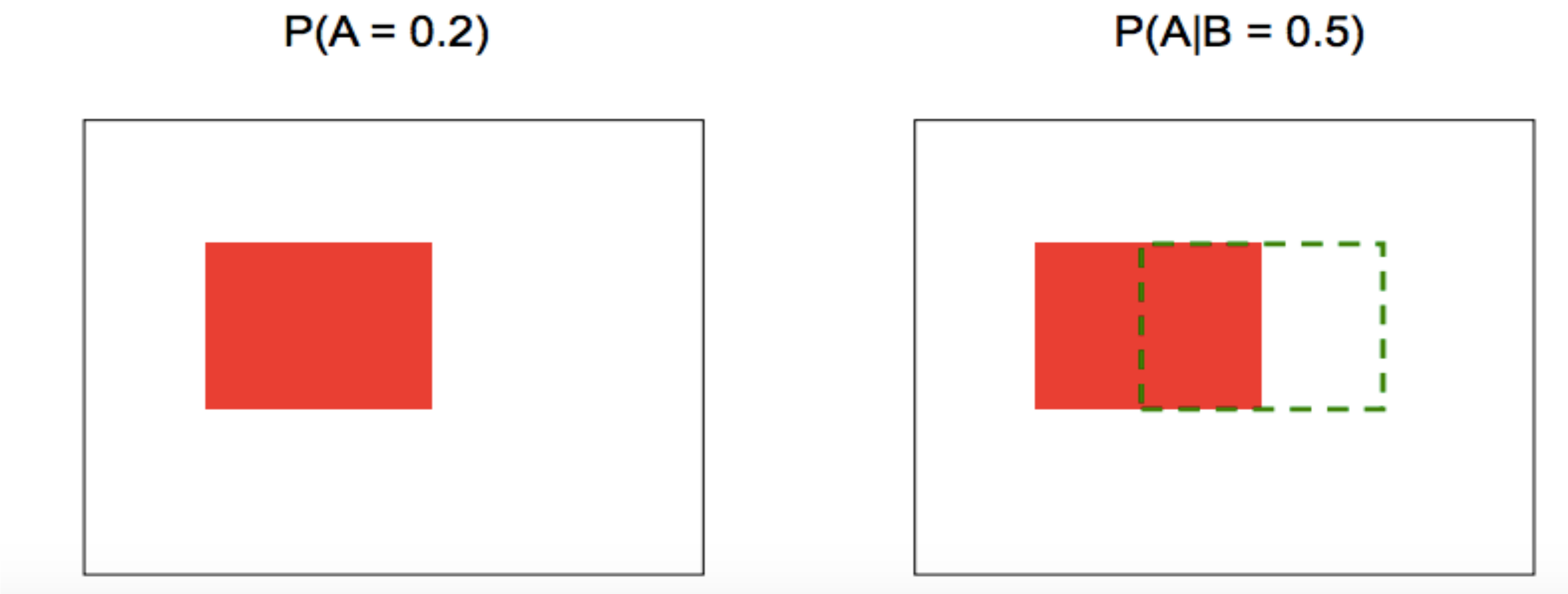
- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

Basic Ingredient for ML Today: Probability

- Notation: P(A)
- e.g., P(Rain)

Conditional Probability

- P(A = 1 | B =1): fraction of cases where A is true if B is true



Conditional Probability

- Knowledge of additional random variables can improve our prior belief of another random variable
- P(Slept in movie) = ?
  - 0.5
- P(Slept in movie | Like Movie) = ?
  - ¼
- P(Didn't sleep in movie | Like Movie) = ?
  - 3/4

Slept	Liked
1	0
0	1
1	1
1	0
0	0
1	0
0	1
0	1

Joint Distribution

- P(A, B): probability a set of random variables will take a specific value

If we assume independence then

$$P(A,B)=P(A)P(B)$$

However, in many cases such an assumption maybe too strong (more later in the class)

Joint Distribution

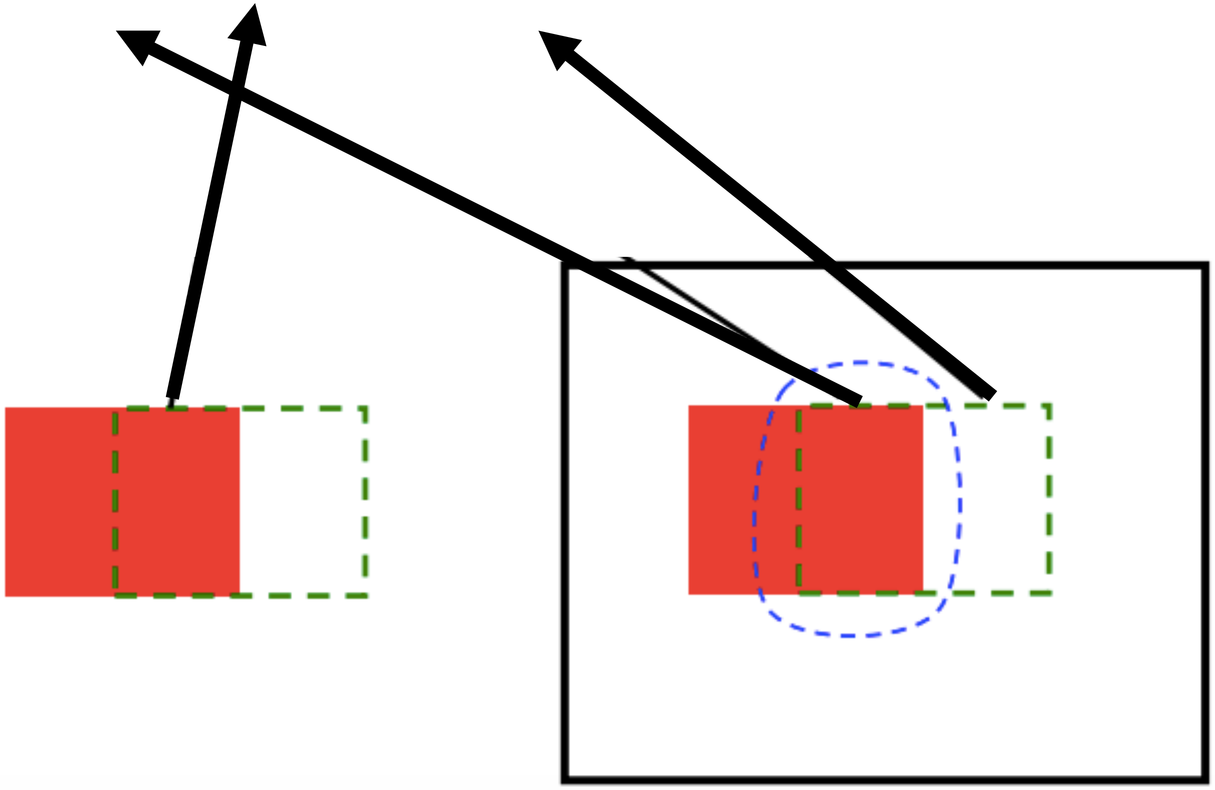
- P(class size > 20) = ?
  - 0.6
- P(summer) = ?
  - 0.4
- P(class size > 20, summer) = ?
  - 0.1

Evaluation of classes

Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

Chain Rule

- Joint probability can be represented with conditional probability
- P(A, B) = P(A|B)*P(B)



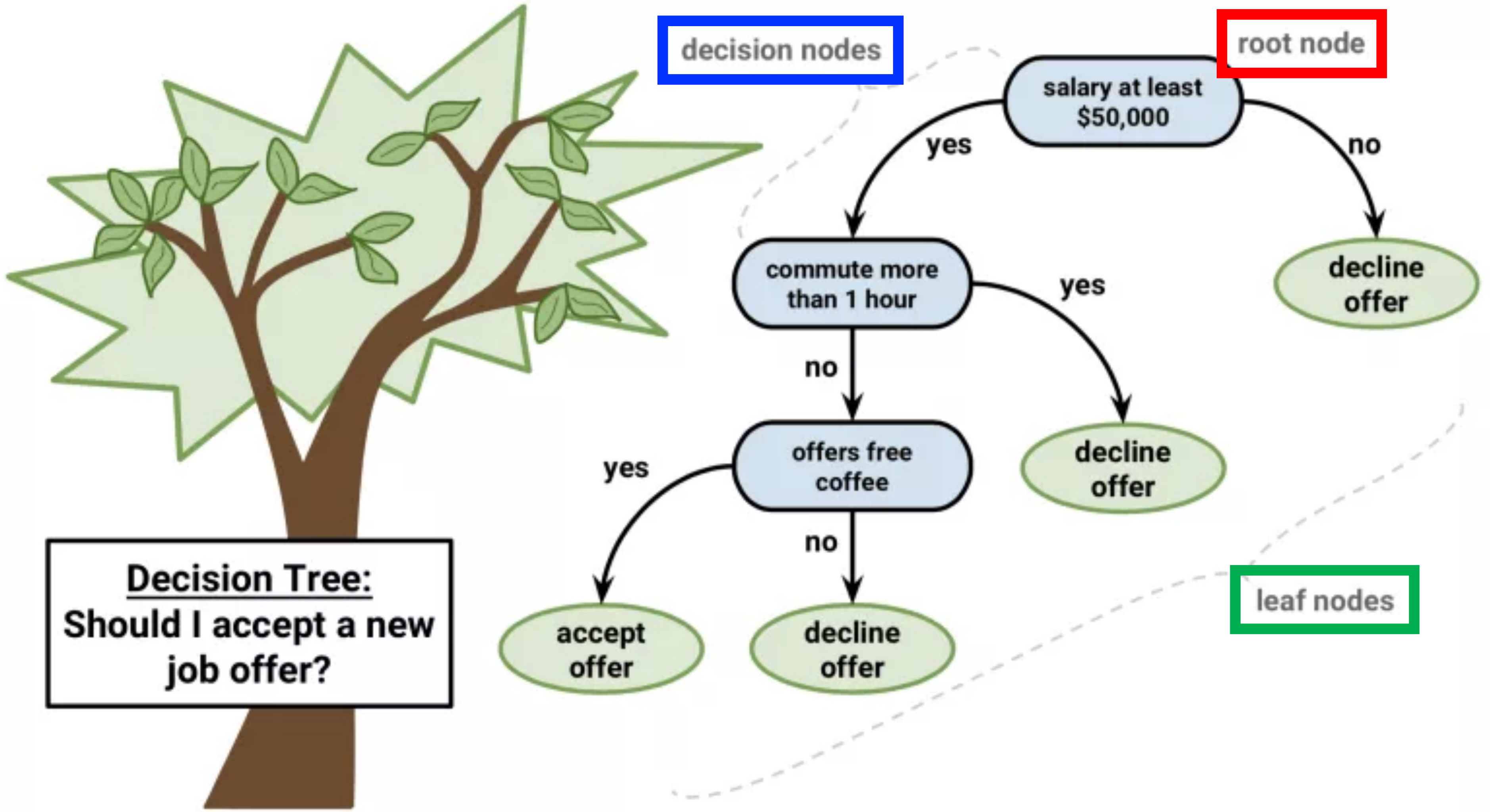
Today’s Topics

- Classification applications
- Discussion: Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

Decision Tree: Discriminative Classifier

- Learns mapping from input features to class label

Decision Tree

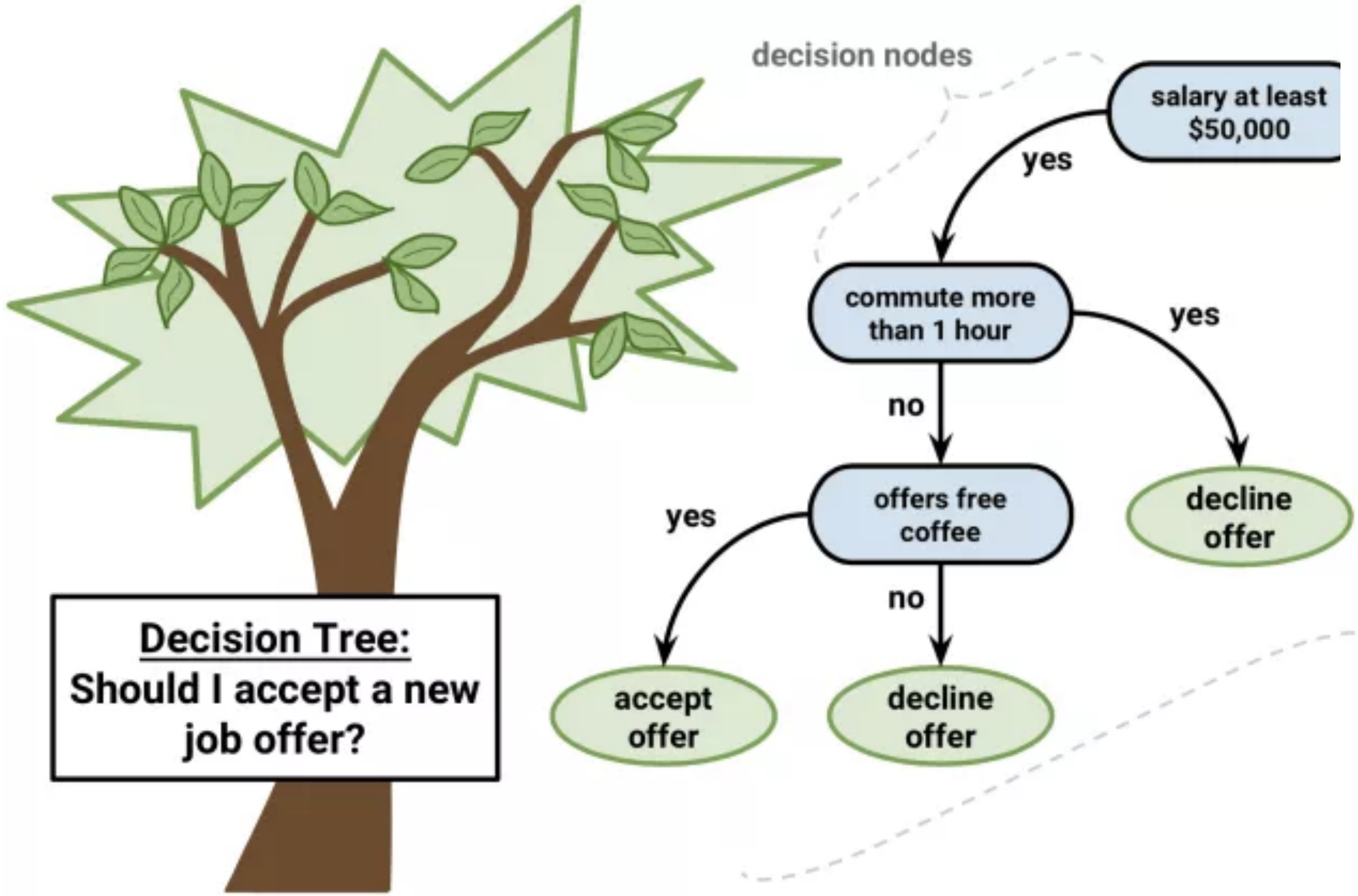


http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

Decision Tree

Test Example

Salary: $44,869
Commute: 35 min
Free Coffee: Yes

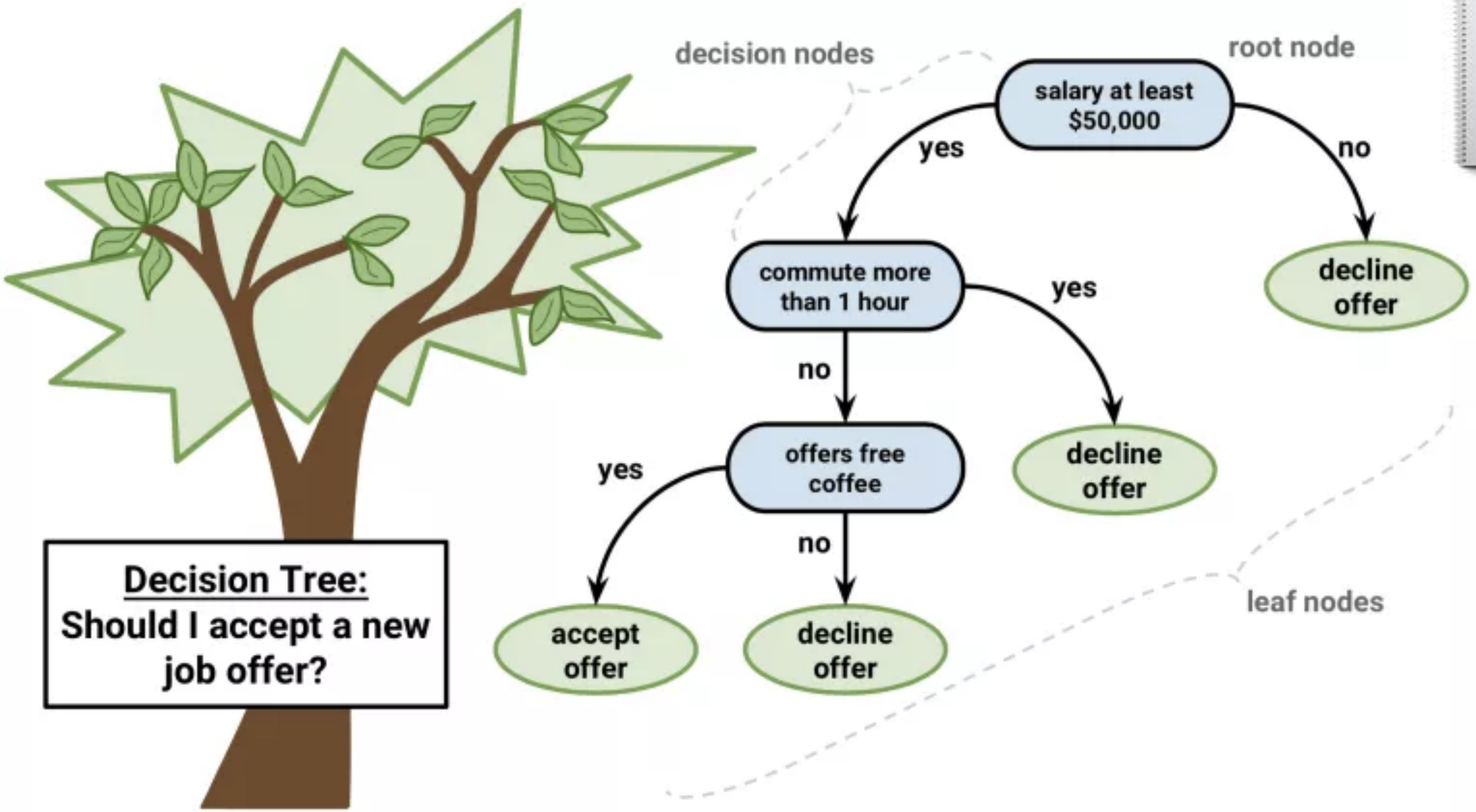


http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

Decision Tree

Test Example

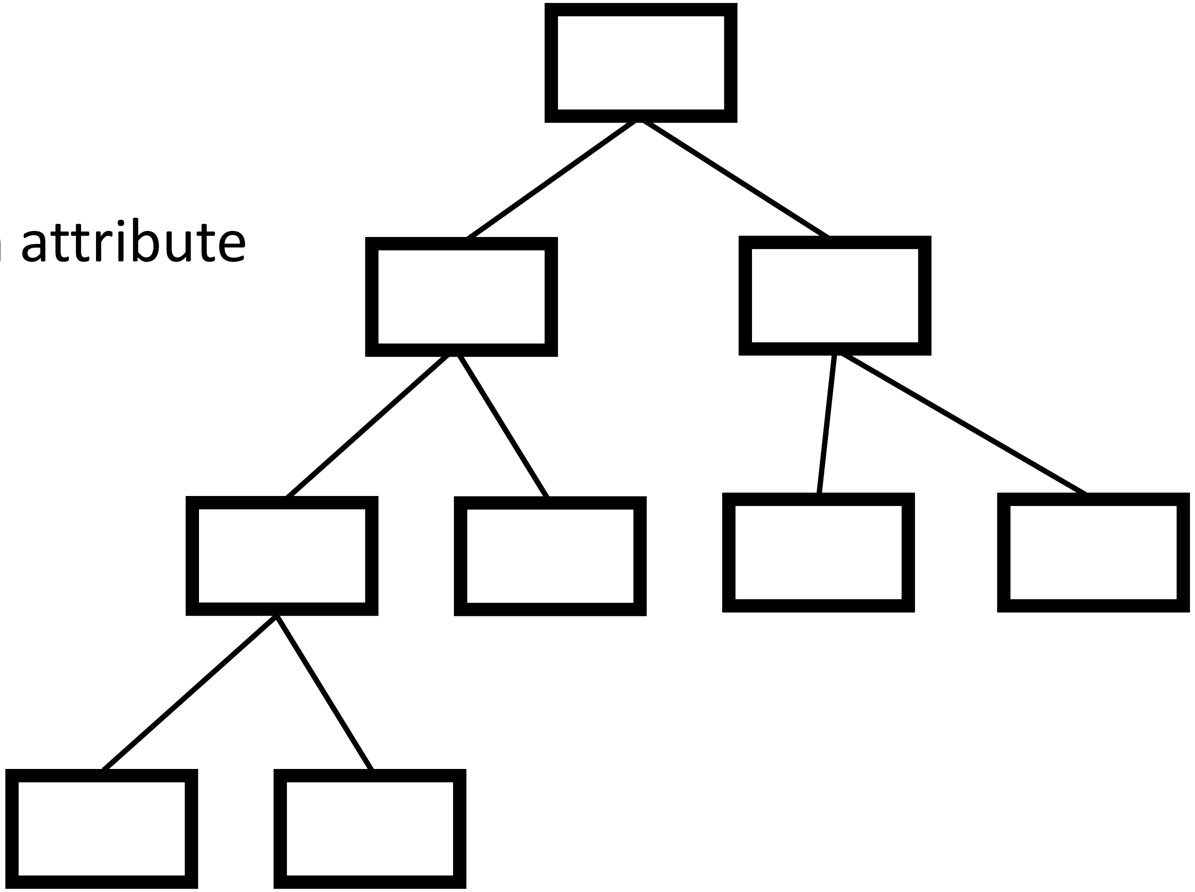
Salary: $62,200
Commute: 45 min
Free Coffee: Yes



http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

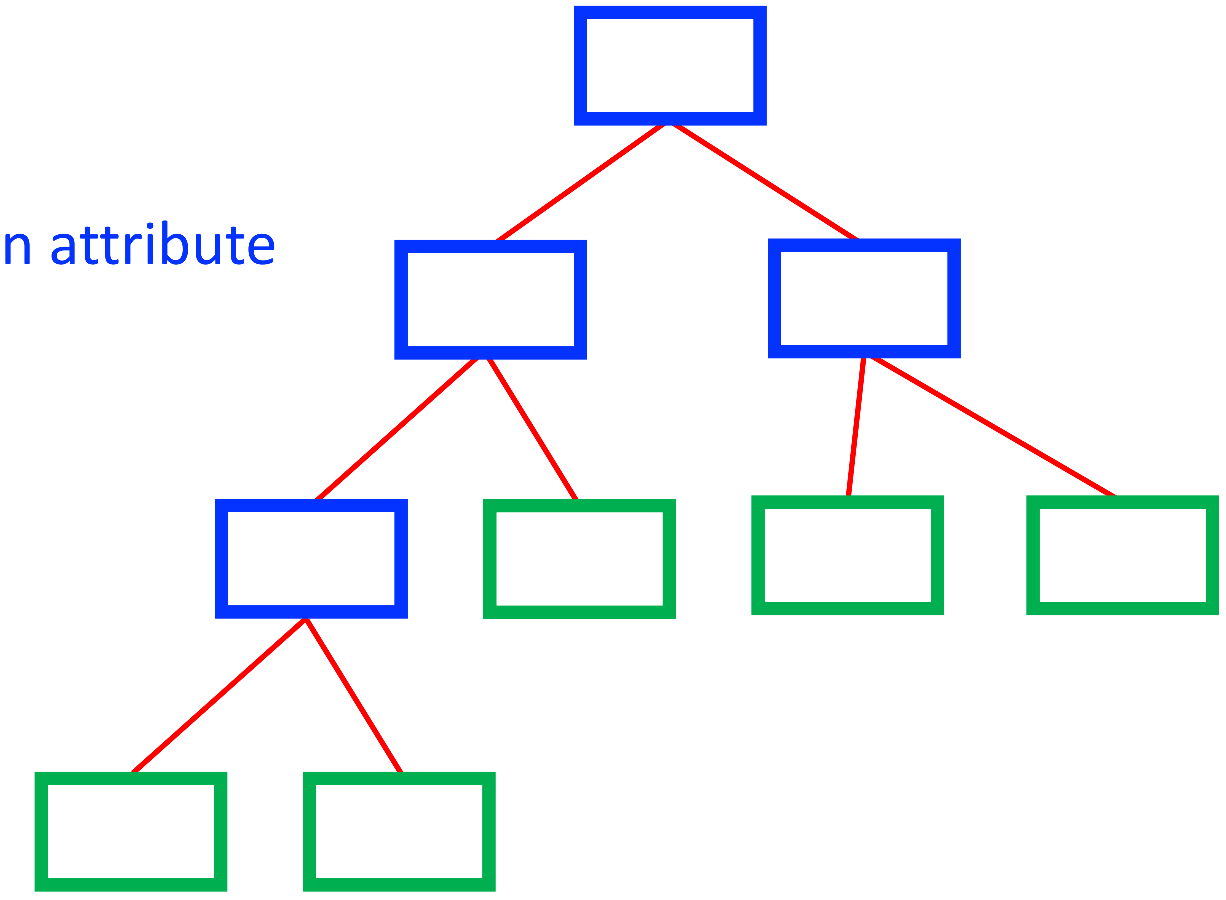
Decision Tree: Generic Structure

- Goal: predict class label
- Representation: Tree
  - Internal (non-leaf) nodes = tests an attribute
  - Branches = attribute value
  - Leaf = classification label



Decision Tree: Generic Structure

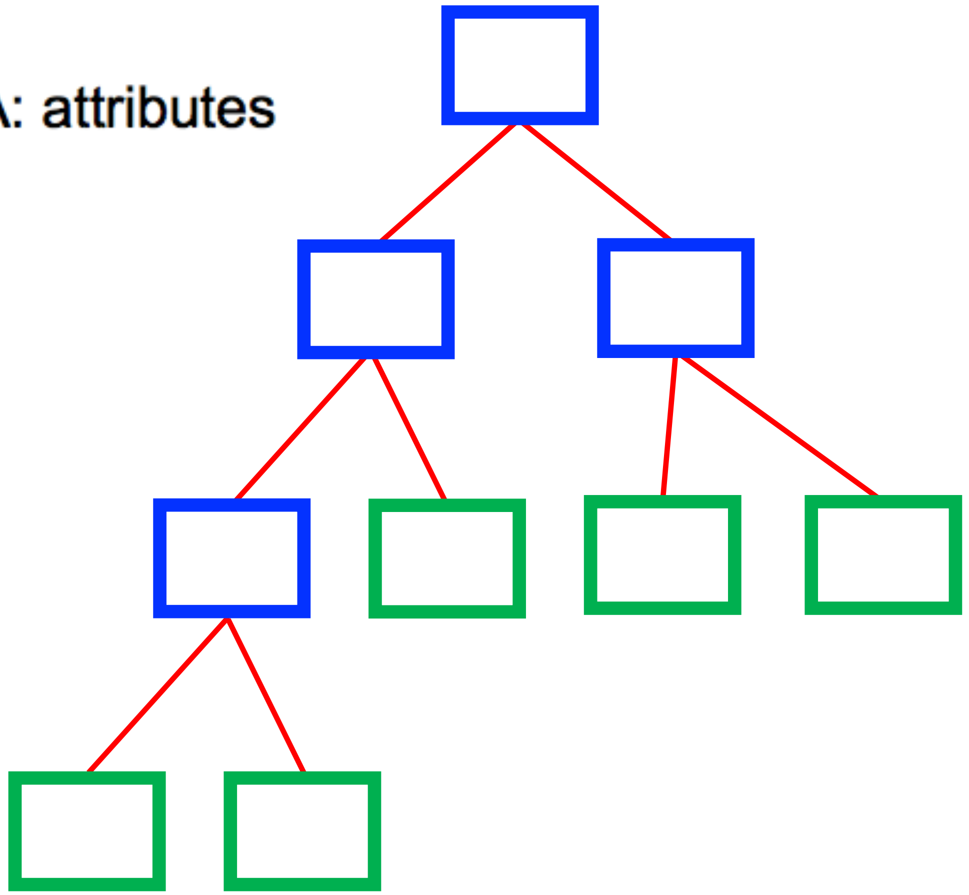
- Goal: predict class label
- Representation: Tree
  - Internal (non-leaf) nodes = tests an attribute
  - Branches = attribute value
  - Leaf = classification label



Decision Tree: Generic Learning Algorithm

- Greedy approach (NP complete problem)

```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
    status = leaf
    class = most common class in n(L)
  else
    status = internal
    a ⇐ bestAttribute(n,A)      Key Decision
    LeftNode = BuildTree(n(a=1), A \ {a})
    RightNode = BuildTree(n(a=0), A \ {a})
  end
end
```



Next “Best” Attribute: Use Entropy

Number of classes

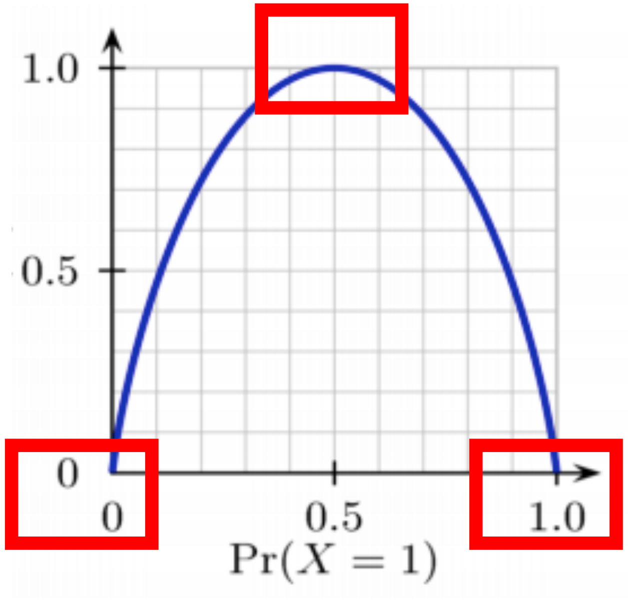
Encodes in bits

$$Entropy = -\sum_{i=1}^{\boxed{n}} \boxed{p_i} \log_{\boxed{2}} p_i$$

Fraction of examples belonging to class *i*

In a binary setting,

- Entropy is 0 when fraction of examples belonging to a class is 0 or 1
- Entropy is 1 when fraction of examples belonging to each class is 0.5



https://www.logcalculator.net/log-2

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Current entropy?

$$Entropy = -\left(\boxed{\tfrac{5}{8}\log_2 \tfrac{5}{8}} + \qquad\qquad \right)$$

Next "Best" Attribute: Use Entropy

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

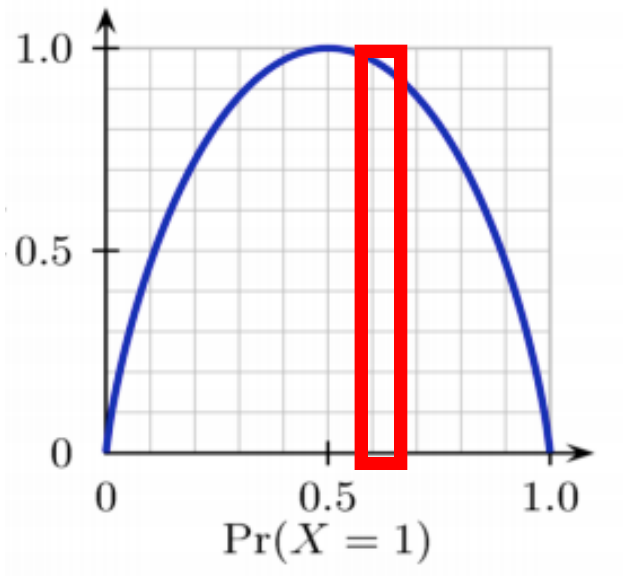
e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Current entropy?

$$Entropy = -(\tfrac{5}{8}\log_2\tfrac{5}{8} + \boxed{\tfrac{3}{8}\log_2\tfrac{3}{8}})$$

Next "Best" Attribute: Use Entropy



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

e.g., Will you like a movie?

Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Current entropy?

$$Entropy = -\left(\tfrac{5}{8}\log_2 \tfrac{5}{8} + \tfrac{3}{8}\log_2 \tfrac{3}{8}\right)$$

$$Entropy = -(-0.42 - 0.53) = \boxed{0.95}$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Liked?
m1	$\boxed{\text{Comedy}}$	Yes
m2	Drama	Yes
m3	$\boxed{\text{Comedy}}$	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	$\boxed{\text{Comedy}}$	Yes
m8	Drama	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Type”?
  - Left tree: “Comedy” = ?

$$Entropy = -(\tfrac{2}{3}\log_2 \tfrac{2}{3} + \tfrac{1}{3}\log_2 \tfrac{1}{3})$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Type”?
  - Left tree: “Comedy” = ?

$$Entropy = -(\tfrac{2}{3}\log_2 \tfrac{2}{3} + \tfrac{1}{3}\log_2 \tfrac{1}{3})$$

$$Entropy = -(-0.53 - 0.39) = 0.92$$

Next "Best" Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Type"?
  - Left tree: "Comedy" = 0.92
  - Right tree: "Drama" = ?

$$Entropy = -(\tfrac{3}{5}\log_2 \tfrac{3}{5} + \tfrac{2}{5}\log_2 \tfrac{2}{5})$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Type”?
  - Left tree: “Comedy” = 0.92
  - Right tree: “Drama” = ?

$$Entropy = -(\tfrac{3}{5}\log_2\tfrac{3}{5} + \tfrac{2}{5}\log_2\tfrac{2}{5})$$

$$Entropy = -(-0.44 - 0.53) = 0.97$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Type	Liked?
m1	Comedy	Yes
m2	Drama	Yes
m3	Comedy	No
m4	Drama	No
m5	Drama	Yes
m6	Drama	No
m7	Comedy	Yes
m8	Drama	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Type”?
  - Left tree: “Comedy” = 0.92
  - Right tree: “Drama” = 0.97
- Information gain by split on “Type”?

$$IG = 0.95 - (\tfrac{3}{8} * 0.92 + \tfrac{5}{8} * 0.97)$$
$$IG = 0$$

Next "Best" Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = ?

$$Entropy = -(\tfrac{2}{3}\log_2 \tfrac{2}{3} + \tfrac{1}{3}\log_2 \tfrac{1}{3})$$

$$Entropy = -(-0.53 - 0.39) = 0.92$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Length”?
  - Left tree: “Short” = 0.92
  - Middle tree: “Medium” = ?

$$Entropy = -(\tfrac{3}{4}\log_2\tfrac{3}{4} + \tfrac{1}{4}\log_2\tfrac{1}{4})$$

$$Entropy = -(-0.32 - 0.5) = 0.82$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “Length”?
  - Left tree: “Short” = 0.92
  - Middle tree: “Medium” = 0.82
  - Right tree: “Long” = ?

Next "Best" Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	Length	Liked?
m1	Short	Yes
m2	Medium	Yes
m3	Medium	No
m4	Long	No
m5	Medium	Yes
m6	Short	No
m7	Short	Yes
m8	Medium	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "Length"?
  - Left tree: "Short" = 0.92
  - Middle tree: "Medium" = 0.82
  - Right tree: "Long" = 0
- Information gain by split on "Length"?

$$IG = 0.95 - (\tfrac{3}{8} * 0.92 + \tfrac{4}{8} * 0.82 + \tfrac{1}{8} * 0)$$
$$IG = 0.19$$

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “IMDb Rating”?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “IMDb Rating”?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “IMDb Rating”?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next "Best" Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

$$IG = 0.95 - \left(\tfrac{5}{8} * \left(\tfrac{5}{5}\log_2\tfrac{5}{5}\right) + \tfrac{3}{8} * \left(\tfrac{3}{3}\log_2\tfrac{3}{3}\right)\right)$$

$$IG = 0.95$$

Next "Best" Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = "Yes" and C2 = "No"
- Entropy if we split on "IMDb Rating"?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

Split at midpoint and measure entropy

Next “Best” Attribute: Use Entropy

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

e.g., Will you like a movie?

Movie	IMDb Rating	Liked?
m1	7.2	Yes
m2	9.3	Yes
m3	5.1	No
m4	6.9	No
m5	8.3	Yes
m6	4.5	No
m7	8.0	Yes
m8	7.5	Yes

- Let C1 = “Yes” and C2 = “No”
- Entropy if we split on “IMDb Rating”?
  - Order attribute values:

    {4.5, 5.1, 6.9, 7.2, 7.5, 8.0, 8.3, 9.3}

    …

Split at midpoint and measure entropy

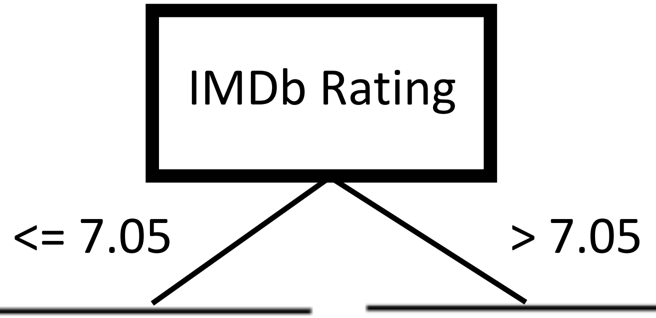
Decision Tree: What is Our First Split?

- Greedy approach (NP complete problem)

```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
     status = leaf
     class = most common class in n(L)
  else
     status = internal
     a ⇐ bestAttribute(n,A)
     LeftNode = BuildTree(n(a=1), A
     RightNode = BuildTree(n(a=0),
  end
end
```

	IG = 0	IG = 0.19	IG = 0.95	
Movie	Type	Length	IMDb Rating	Liked?
m1	Comedy	Short	7.2	Yes
m2	Drama	Medium	9.3	Yes
m3	Comedy	Medium	5.1	No
m4	Drama	Long	6.9	No
m5	Drama	Medium	8.3	Yes
m6	Drama	Short	4.5	No
m7	Comedy	Short	8.0	Yes
m8	Drama	Medium	7.5	Yes

Decision Tree: What Tree Results?

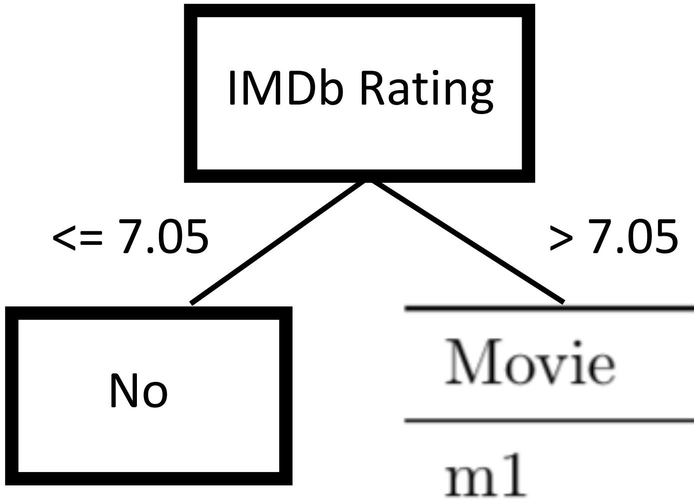


Movie	Type	Length	Liked?
m3	Comedy	Medium	No
m4	Drama	Long	No
m6	Drama	Short	No

Recurse on this tree?

Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m5	Drama	Medium	Yes
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

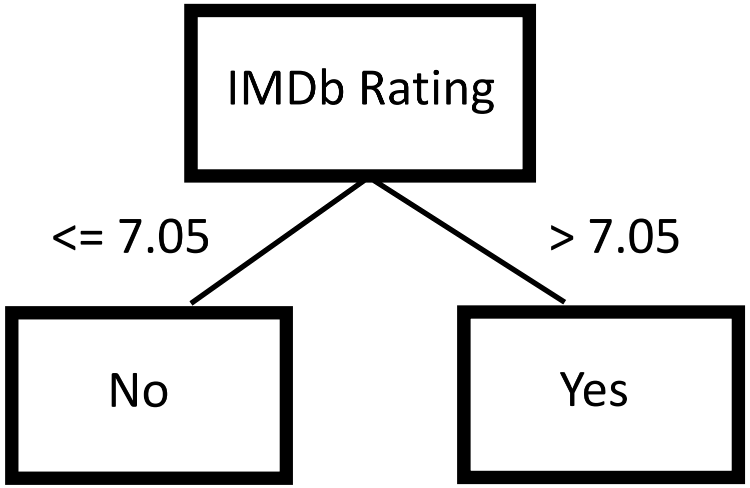
Decision Tree: What Tree Results?



Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m5	Drama	Medium	Yes
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

Recurse on this tree?

Decision Tree: What Tree Results?



Decision Tree: Generic Learning Algorithm

- Greedy approach (NP complete problem)

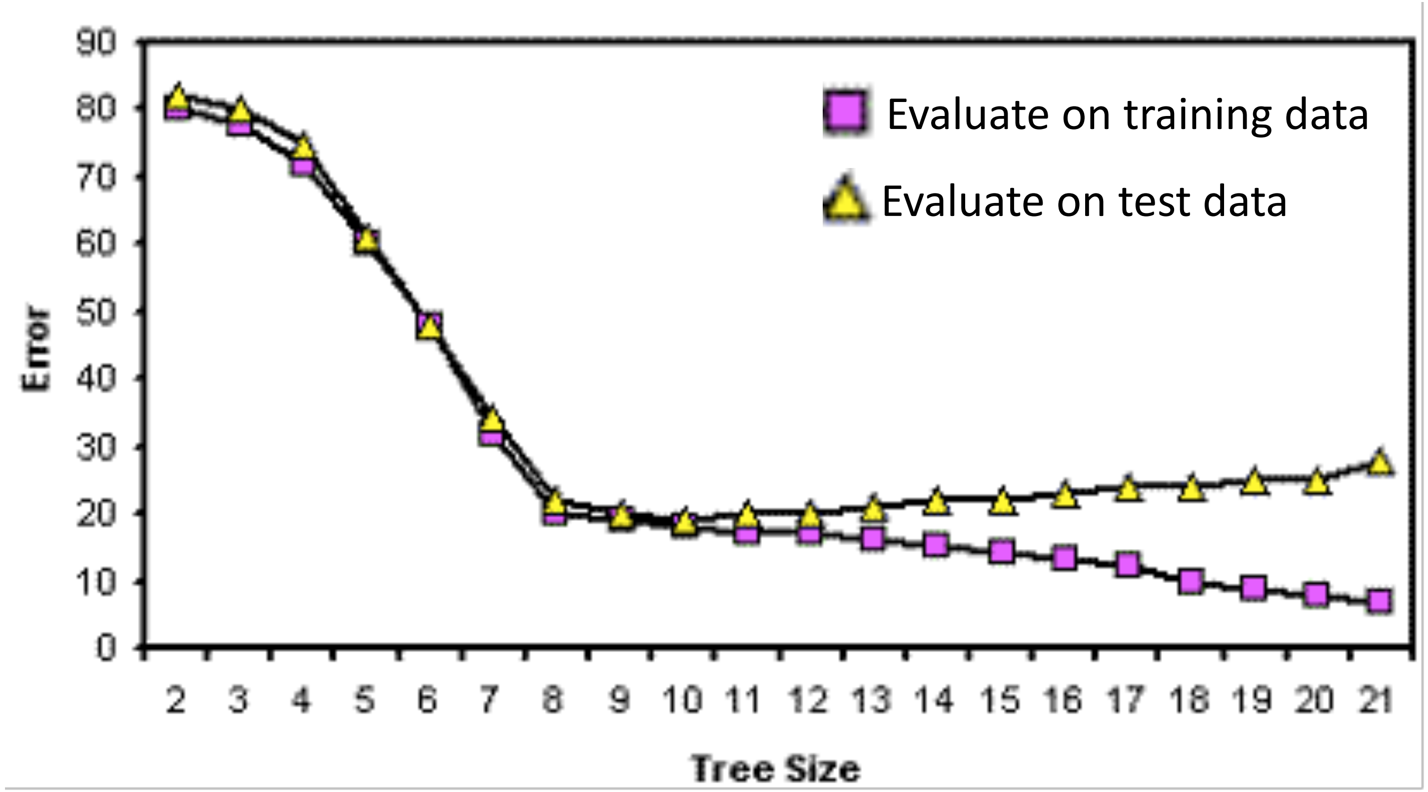
```
Function BuildTree(n,A)    // n: samples (rows), A: attributes
  If empty(A) or all n(L) are the same
     status = leaf
     class = most common class in n(L)
  else
     status = internal
     a ⇐ bestAttribute(n,A)
     LeftNode = BuildTree(n(a=1), A \ {a})
     RightNode = BuildTree(n(a=0), A \ {a})
  end
end
```

Key Decision

- Entropy (maximize information gain)
- Gini Index
- Gain ratio
- Mean squared error
- …

Overfitting

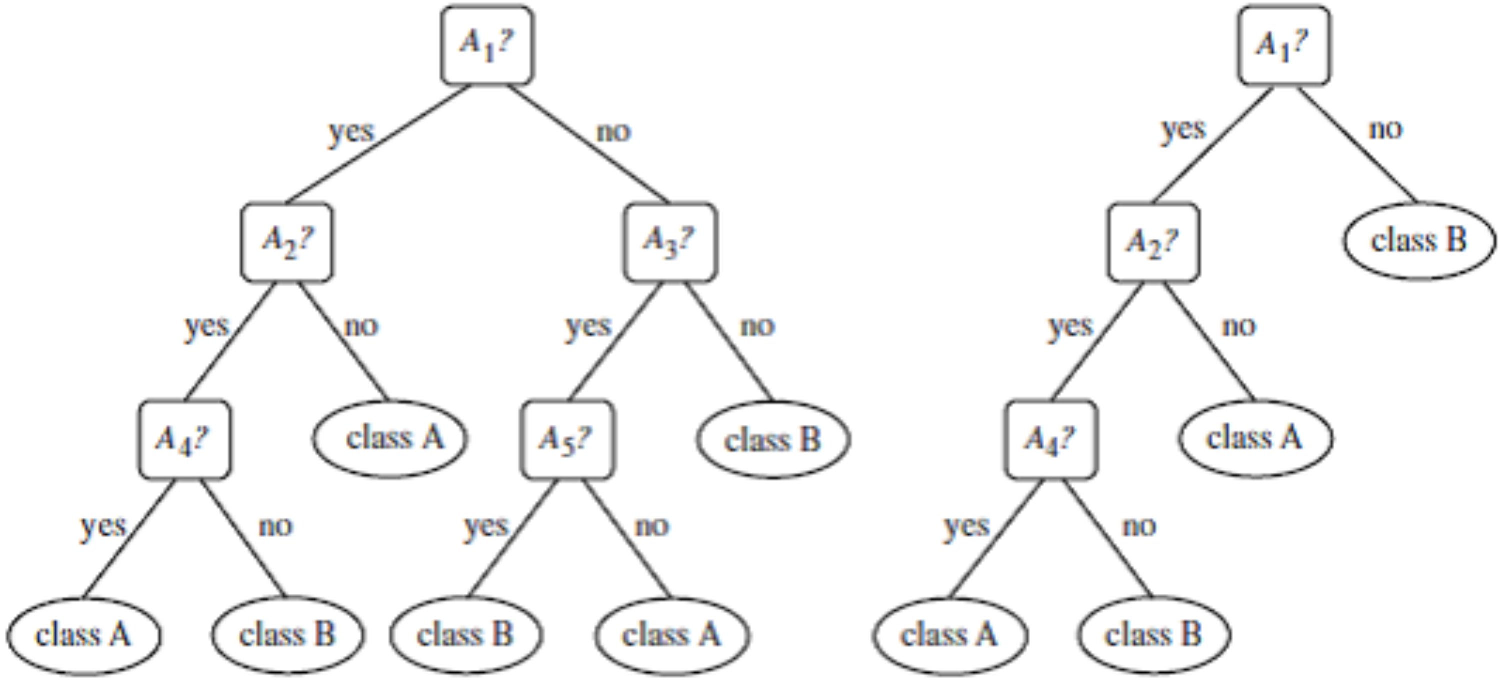
- At what tree size, does training error and testing error grow?



http://www.alanfielding.co.uk/multivar/crt/dt_example_04.htm

Regularization to Avoid Overfitting

- Pruning
  - Pre-pruning: stop tree growth earlier
  - Post-pruning: prune tree afterwards



https://www.clariba.com/blog/tech-20140811-decision-trees-with-sap-predictive-analytics-and-sap-hana-emilio-nieto

Today’s Topics

- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

Naïve Bayes: Generative Classifier

- Learns a model of the joint probability of the input features and each class, and then picks the most probable class

Naïve Bayes: Derivation of Formula

- Recall Chain Rule:
  - P(A, B) = P(A|B) * P(B)
  - P(A, B) = P(B|A) * P(A)
- Therefore:
  - P(A|B) * P(B) = P(B|A) * P(A)
- Rearranging:
  - P(A|B) = (P(B|A) * P(A))/P(B)
- Rewriting:

Need to solve this…
more to follow

Need to solve this…
but how?

$$\boxed{P(C_i|features)} = \frac{\boxed{P(features|C_i)} * \boxed{P(C_i)}}{\boxed{P(features)}}$$

Want to find class with the largest probability

Constant for all classes… so can ignore this!

Naïve Bayes: Assumes Conditionally Independent Features Given Class

- Recall:

$$P(C_i|features) = \boxed{P(features|C_i)} * P(C_i)$$

If we assume independence then

P(A,B)=P(A)P(B)

However, in many cases such an assumption maybe too strong (more later in the class)

$$P(features|C_i) = \prod_{j=1}^{m} P(x_j|C_i)$$

$$P(features|C_i) = P(x_1|C_i) * P(x_2|C_i) * \ldots * P(x_m|C_i)$$

$$P(C_i|features) = \boxed{P(x_1|C_i) * P(x_2|C_i) * \ldots * P(x_m|C_i)} * P(C_i)$$

Naïve Bayes: Different Generative Models Can Yield the Observed Features

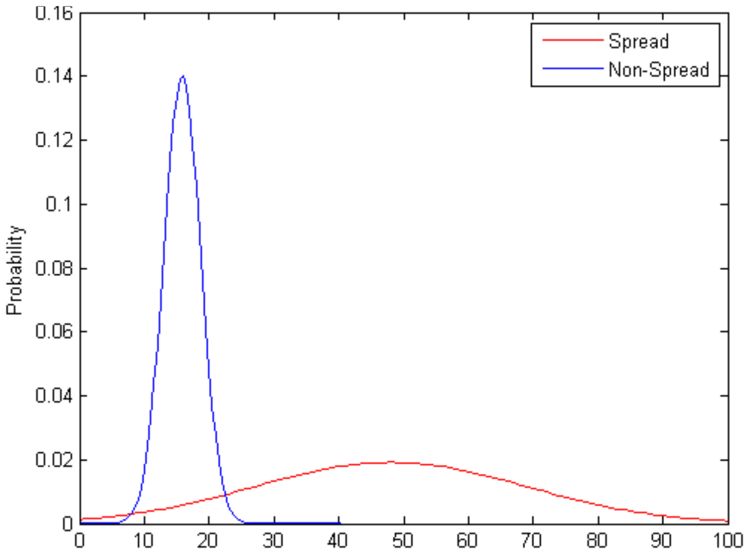
Key Decision: How to compute probability of each feature given the class?

Recall: Want to find class with the largest probability

$$\boxed{P(C_i|features)} = \boxed{P(x_1|C_i)} * \boxed{P(x_2|C_i)} * \ldots * \boxed{P(x_m|C_i)} * P(C_i)$$

Naïve Bayes: Different Generative Models Can Yield the Observed Features

- **Gaussian** Naïve Bayes (typically used for "continuous"-valued features)
  - Assume data drawn from a Gaussian distribution: mean + standard deviation



$$P(C_i|features) = \boxed{P(x_1|C_i) * P(x_2|C_i) * \ldots * P(x_m|C_i)} * P(C_i)$$

Naïve Bayes: Different Generative Models Can Yield the Observed Features

- **Multinomial** Naïve Bayes (typically used for "discrete"-valued features)
  - Assume count data and computes fraction of entries belonging to the category

e.g.,

Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m3	Comedy	Medium	No
m4	Drama	Long	No
m5	Drama	Medium	Yes
m6	Drama	Short	No
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

$$P(C_i|features) = \boxed{P(x_1|C_i) * P(x_2|C_i) * \ldots * P(x_m|C_i)} * P(C_i)$$

Gaussian Naïve Bayes: Example

x_1

e.g.,

IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = ?
  - 5/8 = 0.625

$$P(C_i|features) = P(x_1|C_i) * \boxed{P(C_i)}$$

Gaussian Naïve Bayes: Example

e.g.,

x_1

IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = ?
  - 5/8 = 0.625
- P(Not Liked) = ?
  - 3/8 = 0.375

$$P(C_i|features) = P(x_1|C_i) * \boxed{P(C_i)}$$

Gaussian Naïve Bayes: Example

e.g.,

x_1	
IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked): Mean and Standard Deviation?
  - Mean = 8.06
  - Standard Deviation = 0.81

$$P(C_i|features) = \boxed{P(x_1|C_i)} * P(C_i)$$

Gaussian Naïve Bayes: Example

x_1

e.g.,

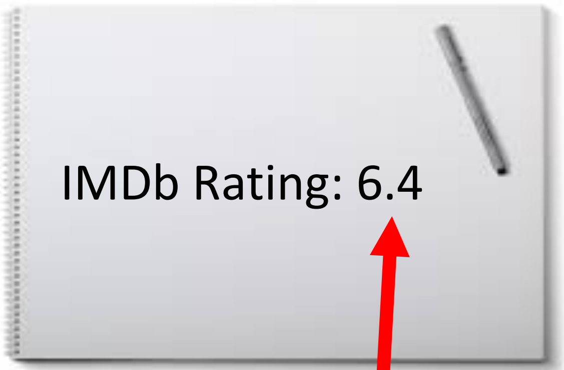
IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked)
  - Mean = 8.06
  - Standard Deviation = 0.81
- P(IMDb Rating | Not Liked): Mean and Standard Deviation?
  - Mean = 5.5
  - Standard Deviation = 1.25

$$P(C_i|features) = \boxed{P(x_1|C_i)} * P(C_i)$$

Gaussian Naïve Bayes: Example

Test Example



- P(Liked | Features)

e.g.,

x_1

IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked)
  - Mean = 8.06
  - Standard Deviation = 0.81
- P(IMDb Rating | Not Liked)
  - Mean = 5.5
  - Standard Deviation = 1.25

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(Can Use: https://planetcalc.com/4986/)

$$P(C_i|features) = \boxed{P(x_1|C_i)} * P(C_i)$$

Test Example

- P(Liked | Features)
  - = 0.06 * 0.625

Gaussian Naïve Bayes: Example

e.g.,

x_1

IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked)
  - Mean = 8.06
  - Standard Deviation = 0.81
- P(IMDb Rating | Not Liked)
  - Mean = 5.5
  - Standard Deviation = 1.25

$$P(C_i|features) = P(x_1|C_i) * P(C_i)$$

Test Example

Gaussian Naïve Bayes: Example

e.g.,

x_1	
IMDb Rating	Liked?
7.2	Yes
9.3	Yes
5.1	No
6.9	No
8.3	Yes
4.5	No
8.0	Yes
7.5	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked)
  - Mean = 8.06
  - Standard Deviation = 0.81
- P(IMDb Rating | Not Liked)
  - Mean = 5.5
  - Standard Deviation = 1.25

- P(Liked | Features)
  - = 0.06 * 0.625
  - = 0.0375
- P(Not Liked | Features)
  - = 0.25 * 0.375
  - = 0.09

Which class is the most probable?

$$P(C_i|features) = P(x_1|C_i) * P(C_i)$$

Multinomial Naïve Bayes: Example

	x_1	x_2	
Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m3	Comedy	Medium	No
m4	Drama	Long	No
m5	Drama	Medium	Yes
m6	Drama	Short	No
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(Comedy | Liked) = ?
  - 2/5 = 0.4
- P(Comedy | Not Liked) = ?
  - 1/3 = 0.333
- P(Drama | Liked) = ?
  - 3/5 = 0.6
- P(Drama | Not Liked) =
  - 2/3 = 0.666

$$P(C_i|features) = \boxed{P(x_1|C_i)} * P(x_2|C_i) * P(C_i)$$

Multinomial Naïve Bayes: Example

	x_1	x_2	
Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m3	Comedy	Medium	No
m4	Drama	Long	No
m5	Drama	Medium	Yes
m6	Drama	Short	No
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

- P(Short | Liked) = ?
  - 2/5 = 0.4
- P(Short | Not Liked) = ?
  - 1/3 = 0.333
- P(Medium | Liked) = ?
  - 3/5 = 0.6
- P(Medium | Not Liked) = ?
  - 1/3 = 0.333
- P(Long | Liked) = ?
  - 0/5 = 0
- P(Long | Not Liked) = ?
  - 1/3 = 0.333

$$P(C_i|features) = P(x_1|C_i) * \boxed{P(x_2|C_i)} * P(C_i)$$

Multinomial Naïve Bayes: Example

Test Example

	x_1	x_2	
Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m3	Comedy	Medium	No
m4	Drama	Long	No
m5	Drama	Medium	Yes
m6	Drama	Short	No
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

Which class is the most probable?

- P(Liked) = 0.63
- P(Not Liked) = 0.38
- P(Comedy | Liked) = 0.4
- P(Comedy | Not Liked) = 0.33
- P(Drama | Liked) = 0.6
- P(Drama | Not Liked) = 0.67

- P(Short | Liked) = 0.4
- P(Short | Not Liked) = 0.33
- P(Medium | Liked) = 0.6
- P(Medium | Not Liked) = 0.33
- P(Long | Liked) = 0
- P(Long | Not Liked) = 0.33

$$P(C_i|features) = P(x_1|C_i) * P(x_2|C_i) * P(C_i)$$

Multinomial Naïve Bayes: Example

Test Example

	x_1	x_2	
Movie	Type	Length	Liked?
m1	Comedy	Short	Yes
m2	Drama	Medium	Yes
m3	Comedy	Medium	No
m4	Drama	Long	No
m5	Drama	Medium	Yes
m6	Drama	Short	No
m7	Comedy	Short	Yes
m8	Drama	Medium	Yes

Which class is the most probable?

Liked) = 0.63

Not Liked) = 0.38

Comedy | Liked) = 0.4

Comedy | Not Liked) = 0.33

Drama | Liked) = 0.6

- P(Drama | Not Liked) = 0.67

- P(Short | Liked) = 0.4
- P(Short | Not Liked) = 0.33
- P(Medium | Liked) = 0.6
- P(Medium | Not Liked) = 0.33
- P(Long | Liked) = 0
- P(Long | Not Liked) = 0.33

To avoid zero, assume training data is so large that adding one to each count makes a negligible difference

$$P(C_i|features) = P(x_1|C_i) * P(x_2|C_i) * P(C_i)$$

Today’s Topics

- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

Evaluating classifiers: confusion matrix

Confusion Matrix: e.g.,

		Actual	
		Spam	Trusted
Predicted	Spam	TP	FP
	Trusted	FN	TN

TP = true positive
TN = true negative
FP = false positive
FN = false negative

Evaluating classifiers: descriptive statistics

Confusion Matrix: e.g.,

		Actual: Spam	Actual: Trusted
Predicted	Spam	50	10
	Trusted	15	100

Commonly-used statistical descriptions:

- How many ***actual spam*** results are there? - 65
- How many ***actual trusted*** results are there? - 110
- How many ***correctly classified instances***? - 150/175 ~ 86%
- How many ***incorrectly classified instances***? - 25/175 ~ 14%

- What is the ***precision***? $$\frac{TP}{TP + FP}$$
  - 50/(50+10) ~ 83%

- What is the ***recall***? $$\frac{TP}{TP + FN}$$
  - 50/(50+15) ~ 77%

Today’s Topics

- Classification applications
- Introduction to Probability
- Decision Tree Model (Discriminative Model)
- Naïve Bayes Model (Generative Model)
- Classification Evaluation Basics
- Lab

References Used for Today's Material

- http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/06_trees.pdf
- http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/tutorial3_CrossVal-DTs.pdf
- http://www.cs.cmu.edu/~epxing/Class/10701/slides/classification15.pdf