

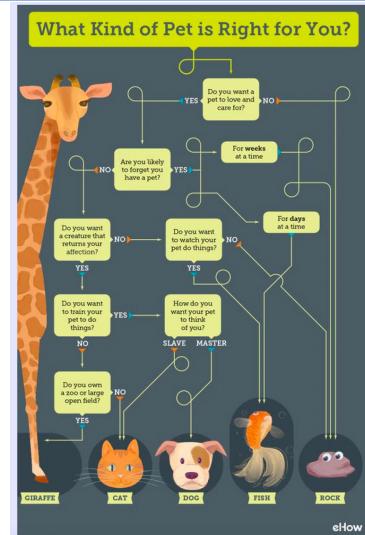
# DECISION TREE AND RANDOM FOREST

1

## Decision Tree Algorithm

2

- Similar to how humans make many different decisions
- Decision trees look at one feature/variable at a time



2

## Decision Tree Algorithm

3

- Root node
- Parent, child/sub nodes
- Branch, splitting
- Leaf nodes

Is a Person Fit?

```

graph TD
    Root[Is a Person Fit?] --> Age30{Age < 30 ?}
    Age30 -- Yes --> Pizzas{Eats a lot of pizzas?}
    Pizzas -- Yes --> Unfit1[Unfit!]
    Pizzas -- No --> Fit1[Fit]
    Age30 -- No --> Exercises{Exercises in the morning?}
    Exercises -- Yes --> Fit2[Fit]
    Exercises -- No --> Unfit2[Unfit!]
  
```

3

## Decision Tree Algorithm

4

- Training dataset

| Day | Outlook  | Temp | Humidity | Wind   | Tennis? |
|-----|----------|------|----------|--------|---------|
| 1   | Sunny    | Hot  | High     | Weak   | No      |
| 2   | Sunny    | Hot  | High     | Strong | No      |
| 3   | Overcast | Hot  | High     | Weak   | Yes     |
| 4   | Rain     | Mild | High     | Weak   | Yes     |
| 5   | Rain     | Cool | Normal   | Weak   | Yes     |
| 6   | Rain     | Cool | Normal   | Strong | No      |
| 7   | Overcast | Cool | Normal   | Strong | Yes     |
| 8   | Sunny    | Mild | High     | Weak   | No      |
| 9   | Sunny    | Cool | Normal   | Weak   | Yes     |
| 10  | Rain     | Mild | Normal   | Weak   | Yes     |
| 11  | Sunny    | Mild | Normal   | Strong | Yes     |
| 12  | Overcast | Mild | High     | Strong | Yes     |
| 13  | Overcast | Hot  | Normal   | Weak   | Yes     |
| 14  | Rain     | Mild | High     | Strong | No      |

4

## Decision Tree Algorithm

- How can we build a decision tree given a data set?

5

## Decision Tree Algorithm

- We will make the **best choice at each step**
- Identify the best feature/attribute for the **each node**

6

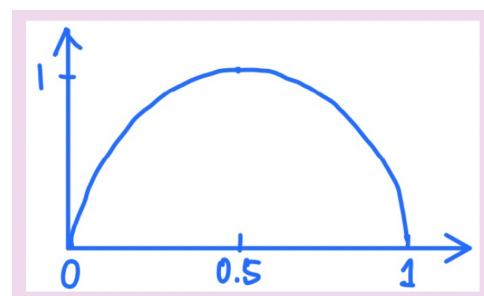
## Decision Tree Algorithm

- 7
- Identify the best feature/attribute for root node
    - Best split: results of each branch should be as **homogeneous** (or **pure**) as possible
    - a feature that reduces **impurity** as much as possible
    - How do we **measure the impurity** in a set of examples
      - **Entropy** from information theory
      - Alternatively, use **Gini Index**

7

## Decision Tree Algorithm

- 8
- Entropy for a distribution over two outcomes



8

## Decision Tree Algorithm

- Quantifying the information content of a feature
  - entropy of the examples before testing the feature minus the entropy of the examples after testing the feature – **Information Gain**

9

## Decision Tree Algorithm

- Quantifying the information content of a feature
  - Information gain or entropy reduction

$$\text{InfoGain} = I_{\text{before}} - I_{\text{after}}$$

10

## Decision Tree Algorithm

11

- Information Gain (entropy reduction)

The first diagram shows a root node labeled "All applicants" containing blue and orange dots representing "hired" and "rejected" status. It splits into two nodes: one for ">5 years" which contains mostly blue dots, and one for "<5 years" which contains mostly orange dots. The second diagram shows a root node labeled "All applicants" containing blue and orange dots. It splits into two nodes: one for "certified" which contains mostly blue dots, and one for "not certified" which contains mostly orange dots. Both diagrams include a legend at the bottom indicating that blue dots represent "hired" and orange dots represent "rejected".

11

## Decision Tree Algorithm

12

- Entropy of the examples before we select a feature for the root node

$$H_{\text{before}} = - \left( \frac{9}{14} \log_2 \left( \frac{9}{14} \right) + \frac{5}{14} \log_2 \left( \frac{5}{14} \right) \right) \approx 0.94$$

| Day | Outlook  | Temp | Humidity | Wind   | Tennis? |
|-----|----------|------|----------|--------|---------|
| 1   | Sunny    | Hot  | High     | Weak   | No      |
| 2   | Sunny    | Hot  | High     | Strong | No      |
| 3   | Overcast | Hot  | High     | Weak   | Yes     |
| 4   | Rain     | Mild | High     | Weak   | Yes     |
| 5   | Rain     | Cool | Normal   | Weak   | Yes     |
| 6   | Rain     | Cool | Normal   | Strong | No      |
| 7   | Overcast | Cool | Normal   | Strong | Yes     |
| 8   | Sunny    | Mild | High     | Weak   | No      |
| 9   | Sunny    | Cool | Normal   | Weak   | Yes     |
| 10  | Rain     | Mild | Normal   | Weak   | Yes     |
| 11  | Sunny    | Mild | Normal   | Strong | Yes     |
| 12  | Overcast | Mild | High     | Strong | Yes     |
| 13  | Overcast | Hot  | Normal   | Weak   | Yes     |
| 14  | Rain     | Mild | High     | Strong | No      |

12

## Decision Tree Algorithm

13

□ Information gain if we select Outlook for the root node

$$\text{Outlook} = \begin{cases} \text{Sunny} & 2+ \quad 3- \quad 5 \text{ total} \\ \text{Overcast} & 4+ \quad 0- \quad 4 \text{ total} \\ \text{Rain} & 3+ \quad 2- \quad 5 \text{ total} \end{cases}$$

$$\text{Gain(Outlook)} = 0.94 - \left( \frac{5}{14} \cdot I\left(\frac{2}{5}, \frac{3}{5}\right) + \frac{4}{14} \cdot I\left(\frac{4}{4}, \frac{0}{4}\right) + \frac{5}{14} \cdot I\left(\frac{3}{5}, \frac{2}{5}\right) \right)$$

$$= 0.247$$

| Day | Outlook  | Temp | Humidity | Wind   | Tennis? |
|-----|----------|------|----------|--------|---------|
| 1   | Sunny    | Hot  | High     | Weak   | No      |
| 2   | Sunny    | Hot  | High     | Strong | No      |
| 3   | Overcast | Hot  | High     | Weak   | Yes     |
| 4   | Rain     | Mild | High     | Weak   | Yes     |
| 5   | Rain     | Cool | Normal   | Weak   | Yes     |
| 6   | Rain     | Cool | Normal   | Strong | No      |
| 7   | Overcast | Cool | Normal   | Strong | Yes     |
| 8   | Sunny    | Mild | High     | Weak   | No      |
| 9   | Sunny    | Cool | Normal   | Weak   | Yes     |
| 10  | Rain     | Mild | Normal   | Weak   | Yes     |
| 11  | Sunny    | Mild | Normal   | Strong | Yes     |
| 12  | Overcast | Mild | High     | Strong | Yes     |
| 13  | Overcast | Hot  | Normal   | Weak   | Yes     |
| 14  | Rain     | Mild | High     | Strong | No      |

13

## Decision Tree Algorithm

14

□ Information gain if we select Humidity for the root node

$$\text{Humidity} = \begin{cases} \text{Normal} & 6+ \quad 1- \quad 7 \text{ total} \\ \text{High} & 3+ \quad 4- \quad 7 \text{ total} \end{cases}$$

$$\text{Gain(Humidity)} = 0.94 - \left( \frac{7}{14} \cdot I\left(\frac{6}{7}, \frac{1}{7}\right) + \frac{7}{14} \cdot I\left(\frac{3}{7}, \frac{4}{7}\right) \right)$$

$$= 0.151$$

| Day | Outlook  | Temp | Humidity | Wind   | Tennis? |
|-----|----------|------|----------|--------|---------|
| 1   | Sunny    | Hot  | High     | Weak   | No      |
| 2   | Sunny    | Hot  | High     | Strong | No      |
| 3   | Overcast | Hot  | High     | Weak   | Yes     |
| 4   | Rain     | Mild | High     | Weak   | Yes     |
| 5   | Rain     | Cool | Normal   | Weak   | Yes     |
| 6   | Rain     | Cool | Normal   | Strong | No      |
| 7   | Overcast | Cool | Normal   | Strong | Yes     |
| 8   | Sunny    | Mild | High     | Weak   | No      |
| 9   | Sunny    | Cool | Normal   | Weak   | Yes     |
| 10  | Rain     | Mild | Normal   | Weak   | Yes     |
| 11  | Sunny    | Mild | Normal   | Strong | Yes     |
| 12  | Overcast | Mild | High     | Strong | Yes     |
| 13  | Overcast | Hot  | Normal   | Weak   | Yes     |
| 14  | Rain     | Mild | High     | Strong | No      |

14

## Decision Tree Algorithm

15

□ Outlook has the greatest information gain

|                       |                        |
|-----------------------|------------------------|
| Gain(Outlook) = 0.247 | Gain(Humidity) = 0.151 |
| Gain(Temp) = 0.029    | Gain(Wind) = 0.048     |

| Day | Outlook  | Temp | Humidity | Wind   | Tennis? |
|-----|----------|------|----------|--------|---------|
| 1   | Sunny    | Hot  | High     | Weak   | No      |
| 2   | Sunny    | Hot  | High     | Strong | No      |
| 3   | Overcast | Hot  | High     | Weak   | Yes     |
| 4   | Rain     | Mild | High     | Weak   | Yes     |
| 5   | Rain     | Cool | Normal   | Weak   | Yes     |
| 6   | Rain     | Cool | Normal   | Strong | No      |
| 7   | Overcast | Cool | Normal   | Strong | Yes     |
| 8   | Sunny    | Mild | High     | Weak   | No      |
| 9   | Sunny    | Cool | Normal   | Weak   | Yes     |
| 10  | Rain     | Mild | Normal   | Weak   | Yes     |
| 11  | Sunny    | Mild | Normal   | Strong | Yes     |
| 12  | Overcast | Mild | High     | Strong | Yes     |
| 13  | Overcast | Hot  | Normal   | Weak   | Yes     |
| 14  | Rain     | Mild | High     | Strong | No      |

15

## Decision Tree Algorithm

16

□ Outlook has the greatest information gain

```

graph TD
    Outlook((Outlook)) -- Sunny --> Sunny1["+ : 9, 11  
- : 1, 2, 8"]
    Outlook -- Overcast --> Overcast["+ : 3, 7, 12, 13  
- : -"]
    Outlook -- Rain --> Rain["+ : 4, 5, 10  
- : 6, 14"]
    Overcast --> Yes1[Yes]
    Rain -- Wind --> WindWeak["+ : 4, 5, 10  
- : -"]
    WindWeak --> Yes2[Yes]
    WindWeak --> No1[No]
    WindWeak --> Strong["+ : 6, 14  
- : -"]
    Strong --> No2[No]
    Sunny1 -- Humidity --> HumidityHigh["+ : -  
- : 1, 2, 8"]
    Sunny1 -- Humidity --> HumidityNormal["+ : 9, 11  
- : -"]
    HumidityHigh --> No3[No]
    HumidityNormal --> Yes3[Yes]
  
```

16

## Gini Impurity to Build Decision Trees

age income student credit\_rate default

|    |            |        |     |           |     |
|----|------------|--------|-----|-----------|-----|
| 0  | youth      | high   | no  | fair      | no  |
| 1  | youth      | high   | no  | excellent | no  |
| 2  | middle_age | high   | no  | fair      | yes |
| 3  | senior     | medium | no  | fair      | yes |
| 4  | senior     | low    | yes | fair      | yes |
| 5  | senior     | low    | yes | excellent | no  |
| 6  | middle_age | low    | yes | excellent | yes |
| 7  | youth      | medium | no  | fair      | no  |
| 8  | youth      | low    | yes | fair      | yes |
| 9  | senior     | medium | yes | fair      | yes |
| 10 | youth      | medium | yes | excellent | yes |
| 11 | middle_age | medium | no  | excellent | yes |
| 12 | middle_age | high   | yes | fair      | yes |
| 13 | senior     | medium | no  | excellent | no  |

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

$$Gini_A(D) = \frac{n_1}{n} Gini(D_1) + \frac{n_2}{n} Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

Credit Rating

```

graph TD
    CR[Credit Rating] --> E[Excellent]
    CR --> F[Fair]
    E --> Y1[Yes: 3]
    E --> N1[No: 3]
    E --> G1[Gini: 0.5]
    F --> Y2[Yes: 2]
    F --> N2[No: 6]
    F --> G2[Gini: 0.37]
  
```

Gini Impurity for Credit Rating is 0.429

17

## Decision Tree for Regression

```

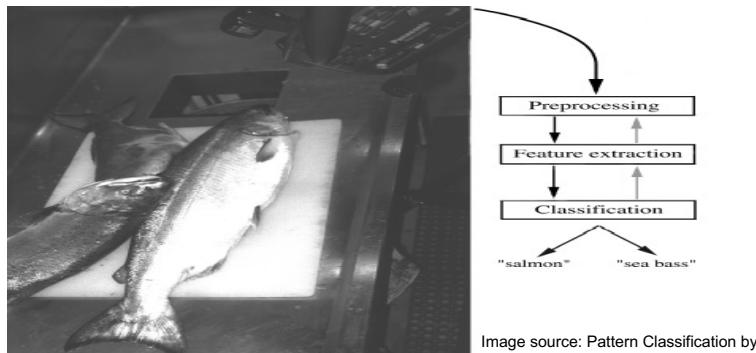
graph TD
    Root["cgpa <= 8.845  
mse = 0.021  
samples = 320  
value = 0.727"] --> S1["cgpa <= 8.035  
mse = 0.012  
samples = 210  
value = 0.651"]
    Root --> S2["cgpa <= 9.195  
mse = 0.005  
samples = 110  
value = 0.872"]
    S1 --> L1["mse = 0.009  
samples = 60  
value = 0.533"]
    S1 --> L2["mse = 0.006  
samples = 150  
value = 0.698"]
    S2 --> L3["mse = 0.003  
samples = 55  
value = 0.816"]
    S2 --> L4["mse = 0.001  
samples = 55  
value = 0.928"]
  
```

18

## An example: A Practical Problem

19

- A fish-packing plant wants to automate the process of sorting incoming fish according to species
- Problem: Identifying species of a fish on a conveyor belt
  - Species: Sea bass and salmon

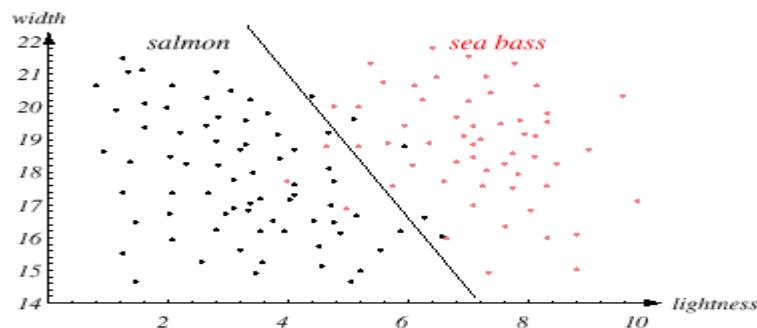


19

## Feature Space

20

- Two features for classification



Can we improve the performance further? If yes, how?

Image source: Pattern Classification by Duda, Hart and Stork

20

## Feature Space

21

- Two features for classification

Nonlinear Decision Boundary

width

salmon

sea bass

lightness

Perfect Classification! Is there a catch?

Image source: Pattern Classification by Duda, Hart and Stork

21

## Generalization

22

- Classification Goal: Make **accurate predictions** for **new/unseen data** - **Good Generalization**
- The model should NOT be tuned to the specific characteristics of the training data – **Overfitting**
- In practice, training data is likely to contain some noise

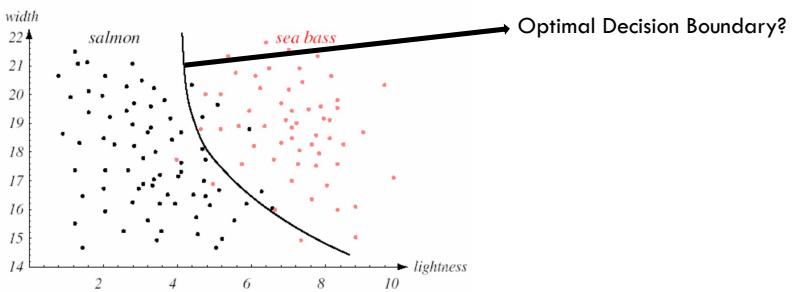
We are better off with a slightly poorer performance on the training examples if this means that our classifier will have better performance on unseen patterns.

22

## Generalization

23

- Classification Goal: Make accurate predictions for new/unseen data - Good Generalization



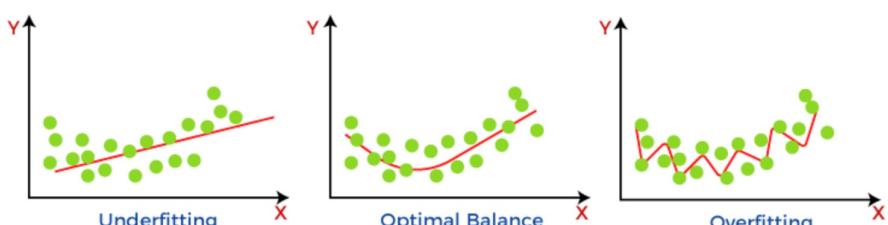
A scatter plot with 'width' on the y-axis (ranging from 14 to 22) and 'lightness' on the x-axis (ranging from 2 to 10). Black dots represent 'salmon' and red dots represent 'sea bass'. A black curve labeled 'Optimal Decision Boundary?' separates the two classes. The curve starts at approximately (3.5, 21) and ends at (9, 15).

- A decision boundary that provides an optimal tradeoff between accuracy on the training set and unseen data

23

## Avoid Overfitting and Achieve Optimal Tradeoff

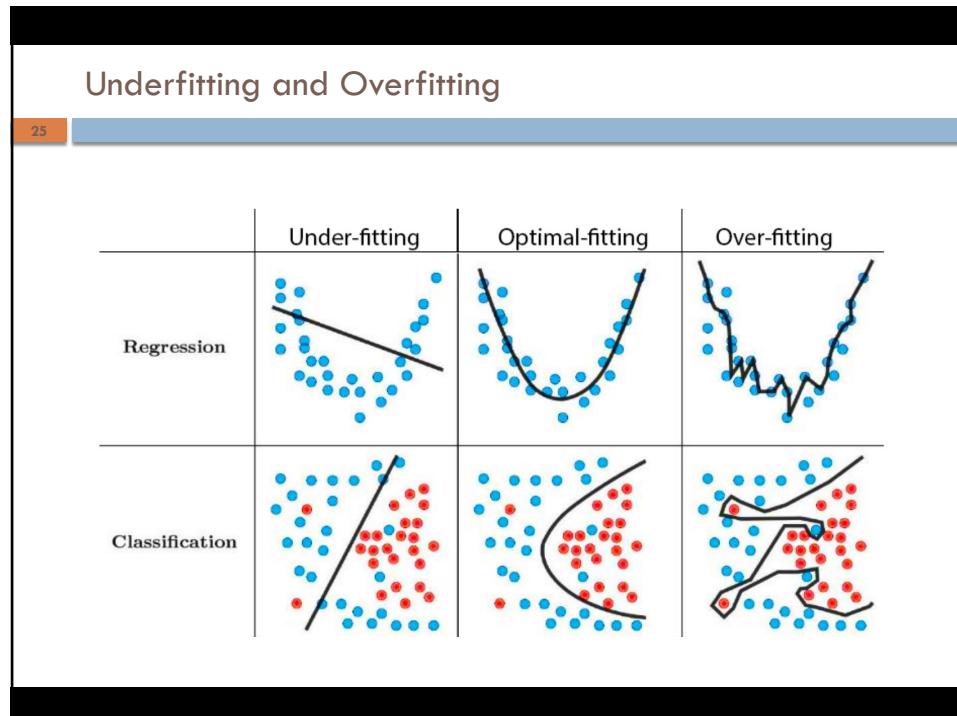
24



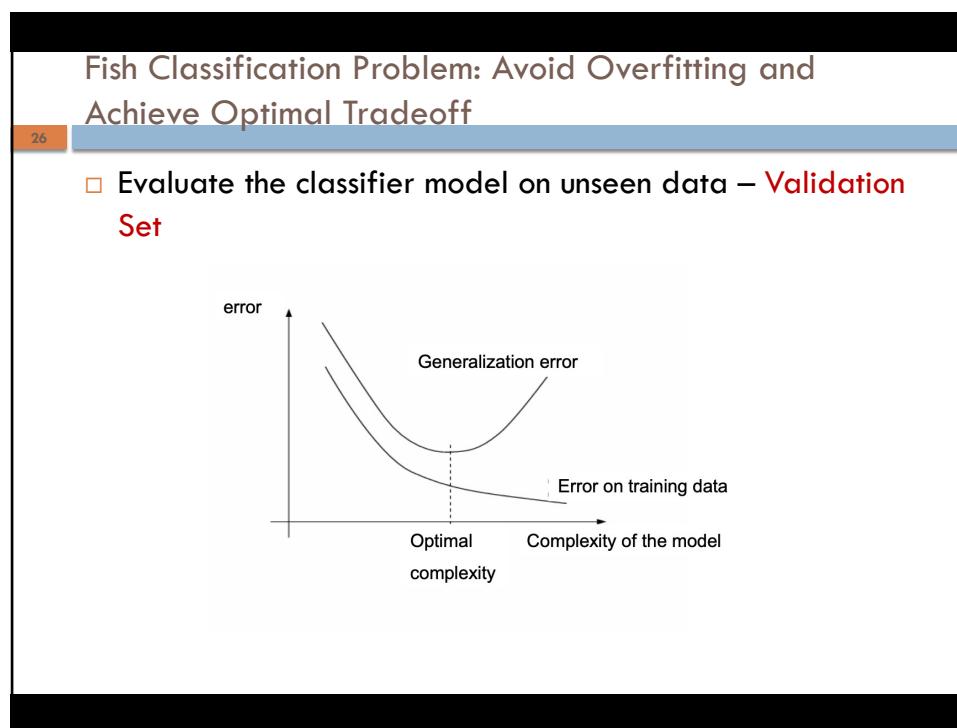
Three side-by-side plots showing data points (green circles) and fitted curves (red lines) on an X-Y coordinate system.

- The first plot, labeled "Underfitting", shows a straight line that fails to capture the underlying trend of the data points.
- The second plot, labeled "Optimal Balance", shows a curve that fits the general trend of the data points well.
- The third plot, labeled "Overfitting", shows a highly oscillatory curve that passes through every single data point but fails to generalize to new data.

24



25



26

## Bias and Variance in Machine Learning

27

- Bias: The model **makes strong assumptions** about the training data to **simplify the learning process**
- Examples: linear regression algorithms or shallow decision trees, which assume simple relationships even when the data patterns are more complex
- Variance: The **model's sensitivity** to **fluctuations** in the training data (the model's prediction changes as it is trained on different subsets of the training data)

27

## Bias and Variance in Machine Learning

28

- Models with high bias have low variance, and models with low bias have high variance (inverse relationship)

The graph shows two curves: a green curve labeled "bias" and a purple curve labeled "variance". The green curve starts high on the y-axis and decreases as the x-axis (labeled "complexity") increases. The purple curve starts low on the y-axis and increases as the x-axis increases. The two curves intersect at a single point, representing the optimal level of complexity where the total error (sum of bias and variance) is minimized.

- Bias-variance trade-off: Minimizing errors caused by oversimplification and excessive complication

28

## Feature Space

29

□ Decision Boundaries in Decision Tree

29

## Avoid Overfitting and Achieve Optimal Tradeoff

30

□ Decision Tree versus Random Forest

30

## Avoid Overfitting and Achieve Optimal Tradeoff

31

- Decision Tree versus Random Forest

31

## Random Forest

32

- Ensemble learning is a machine learning technique that aggregates two or more learners to produce better predictions
  - committee-based learning

32

## Random Forest

33

- Base learner, base model, base estimator - refers to the individual models in ensemble algorithms
- consolidating base learner predictions
  - Majority Voting, Averaging

33

## Random Forest

34

- Random forest uses **bagging** to construct ensembles of **randomized decision trees**
  - Bagging - **bootstrap sampling** and **aggregation**
  - Bootstrap sampling to derive multiple new datasets from one initial training dataset to train multiple base learners

34

## Bootstrap Sampling

35

- Random sampling with replacement

□ Each bootstrap sample only contains approximately **63.2%** of the unique samples from the original dataset

35

## Random Forest

36

- Random forest uses **bagging** to construct ensembles of **randomized decision trees**

36

## Random Forest

37

- Random forest uses **bagging** to construct ensembles of **randomized decision trees**
  - considers **random subsets** of features when splitting a node
  - **max\_features** parameter
- The **greater diversity** among combined models, the **more accurate** the resulting ensemble model

37

## Estimating generalization Performance: Out-of-bag (OOB) error/score

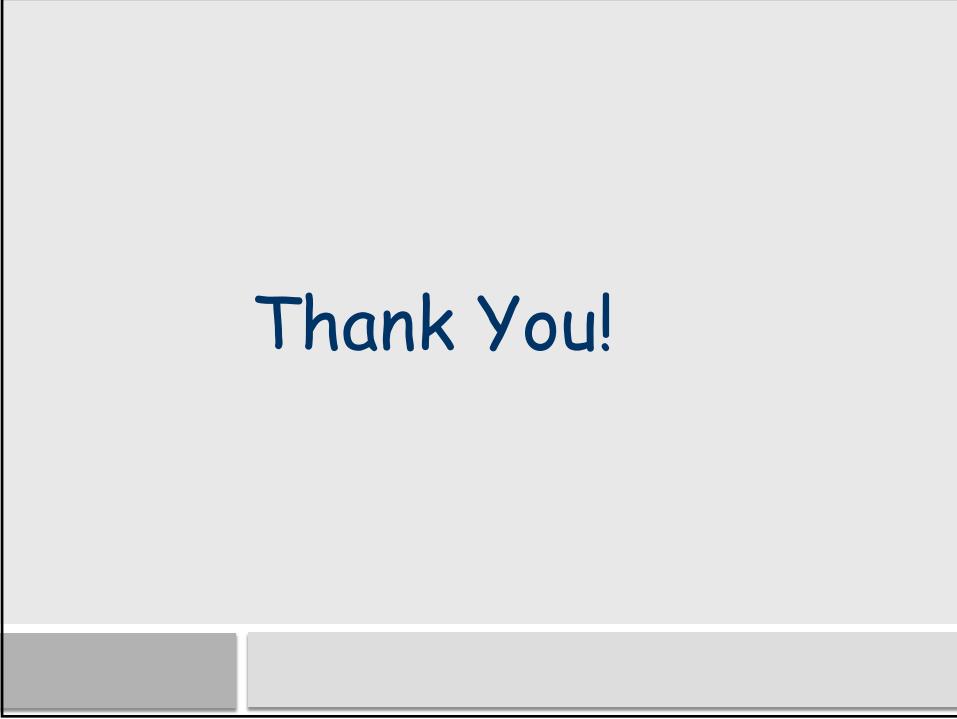
38

- **Out-of-bag** samples as test sets for evaluation
  - Out-of-bag samples are the unique sets of datapoints that are not used for model fitting

|                  |          |       |       |          |       |       |       |       |       |          |  |
|------------------|----------|-------|-------|----------|-------|-------|-------|-------|-------|----------|--|
| Original Dataset | $x_1$    | $x_2$ | $x_3$ | $x_4$    | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ |  |
| Bootstrap 1      | $x_1$    | $x_6$ | $x_2$ | $x_3$    | $x_5$ | $x_9$ | $x_7$ | $x_4$ | $x_8$ | $x_2$    |  |
| Bootstrap 2      | $x_{10}$ | $x_1$ | $x_3$ | $x_5$    | $x_1$ | $x_7$ | $x_4$ | $x_2$ | $x_1$ | $x_8$    |  |
| Bootstrap 3      | $x_1$    | $x_5$ | $x_4$ | $x_1$    | $x_2$ | $x_4$ | $x_2$ | $x_6$ | $x_9$ | $x_2$    |  |
| Training Sets    |          |       |       |          |       |       |       |       |       |          |  |
| Test Sets        | $x_3$    | $x_7$ | $x_8$ | $x_{10}$ |       |       |       |       |       |          |  |

- Each bootstrap sample only contains approximately **63.2%** of the unique samples from the original dataset

38



Thank You!