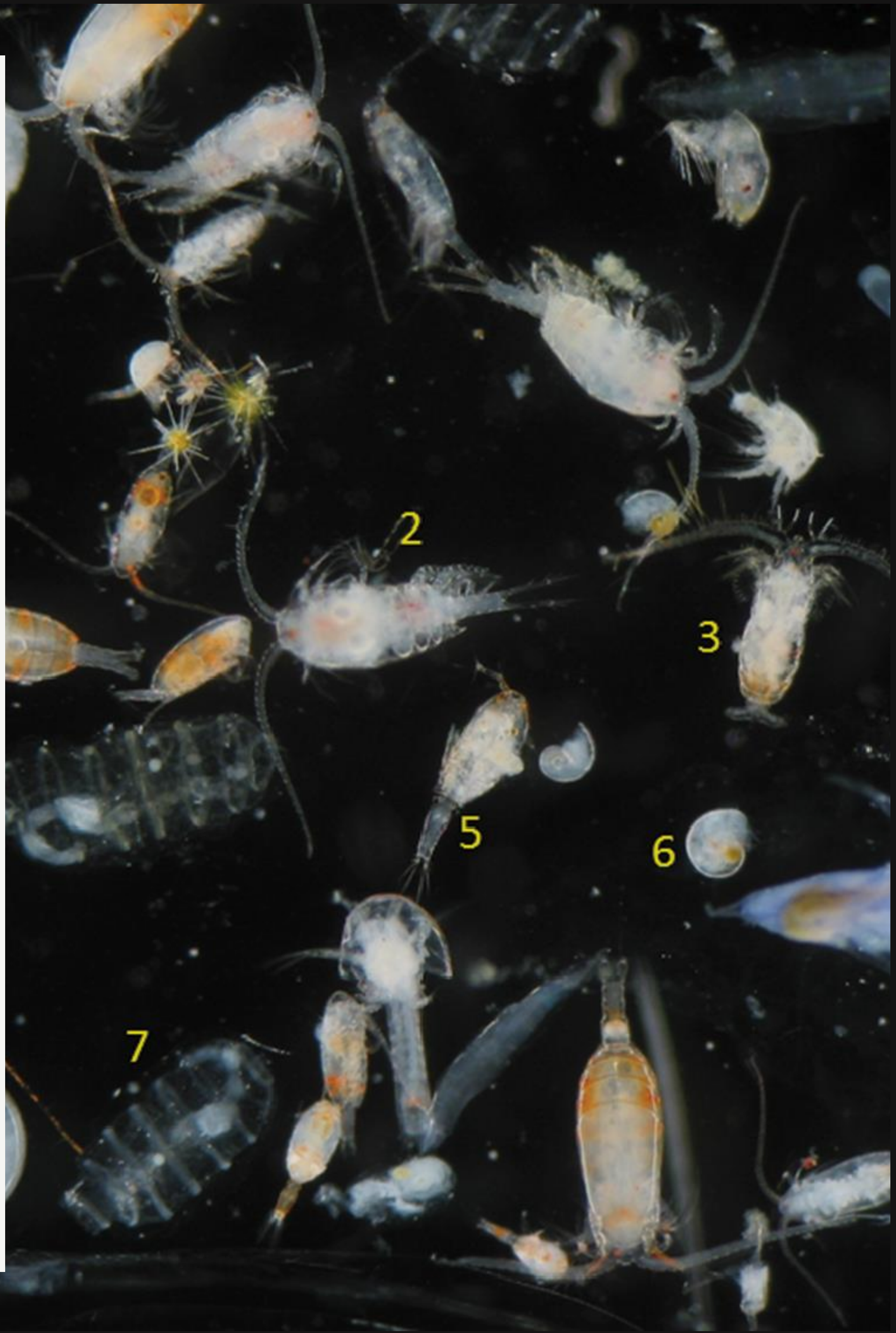
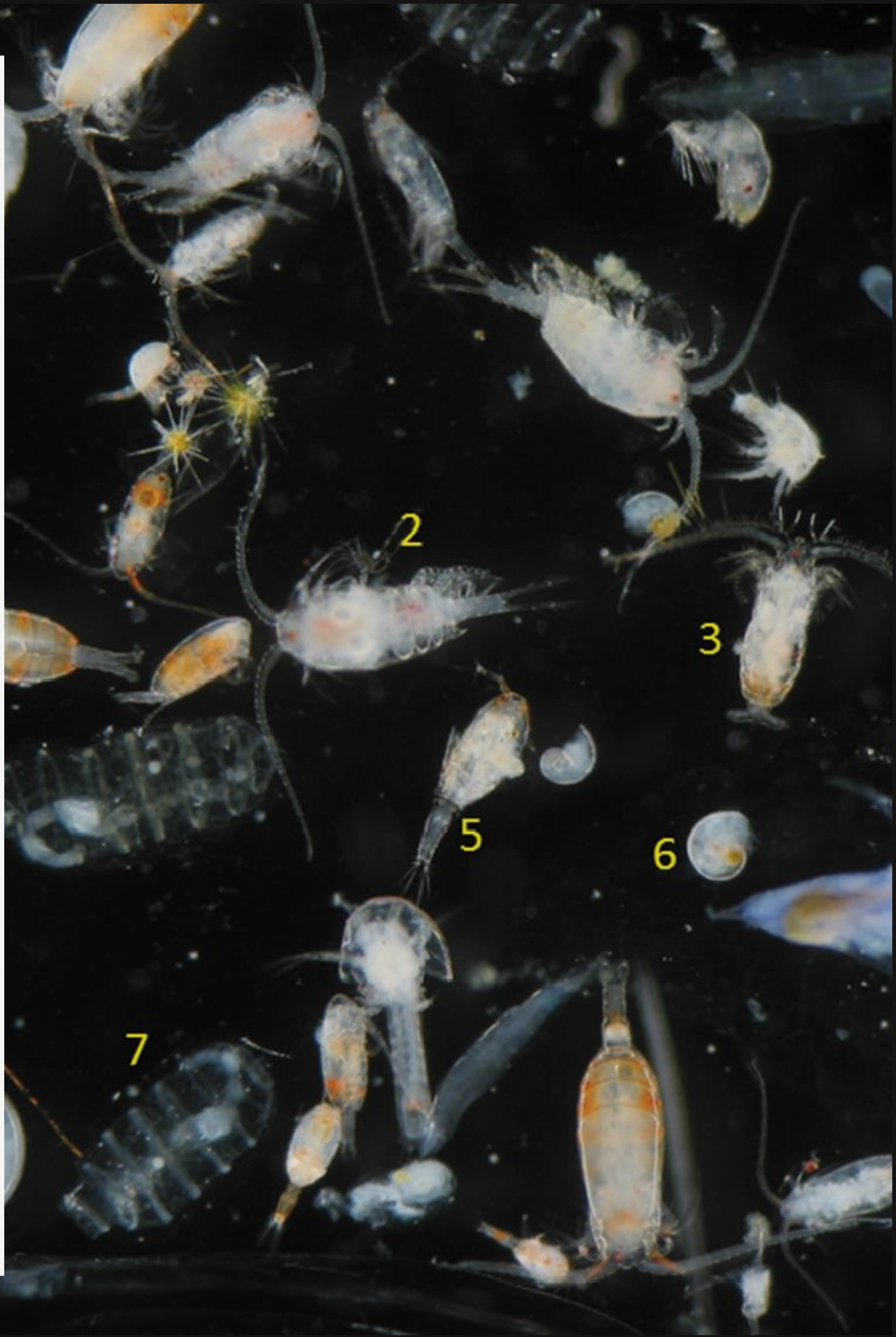
FW 599 Special Topics: Multivariate Analysis of Ecological Data in R

Lecture 12: Classification and Regression Trees

Tuesday, November 12, 2024



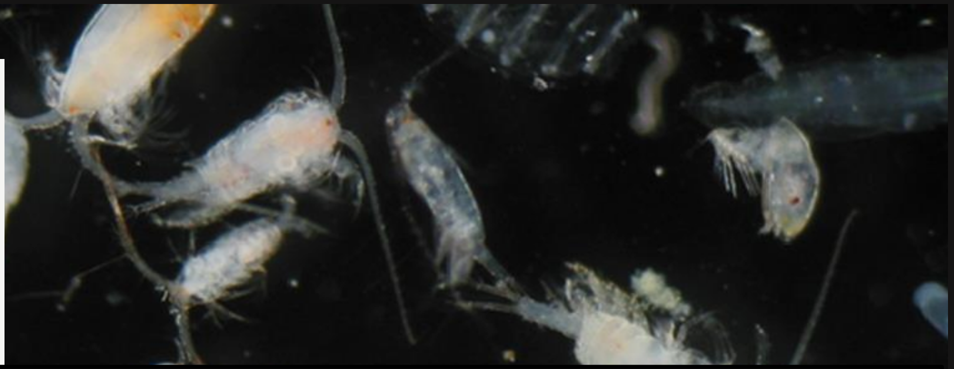
Lecture 12: Classification and Regression Trees

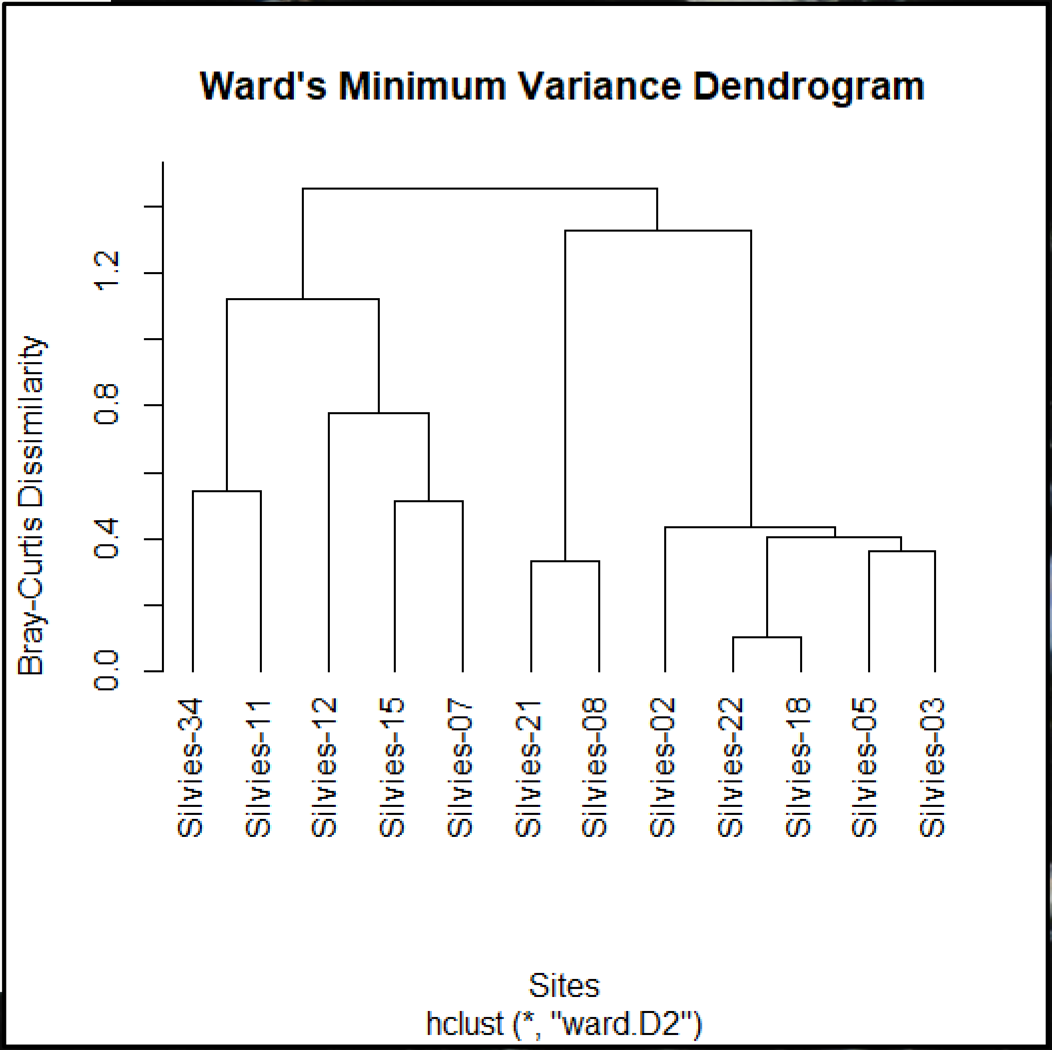


Recap: Cluster Analysis

Hierarchical cluster analysis is used to classify objects, such as species, habitats, or environmental variables, into clusters based on their similarities or dissimilarities.

This technique helps ecologists to identify natural groupings and patterns within ecological data.

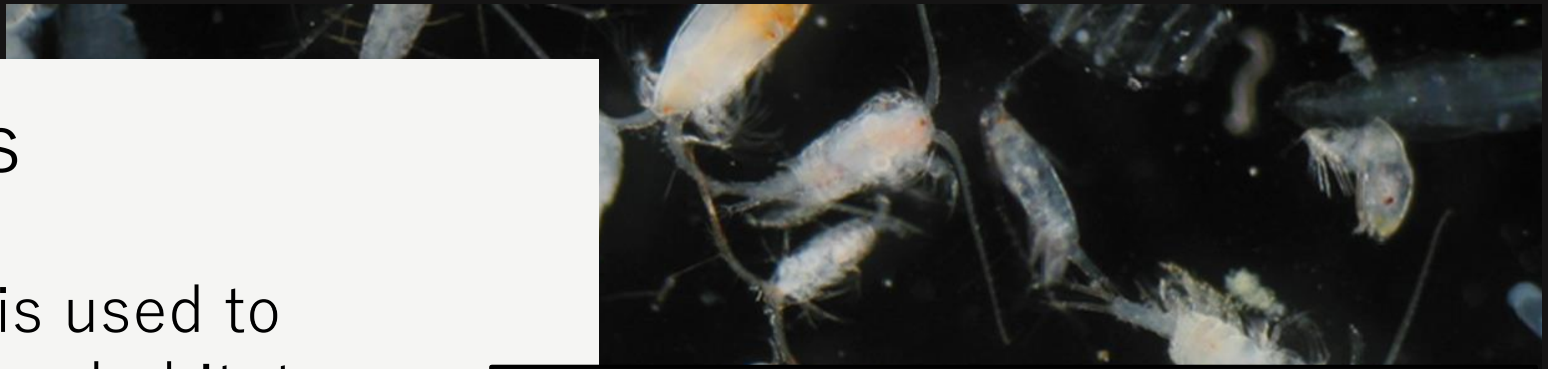


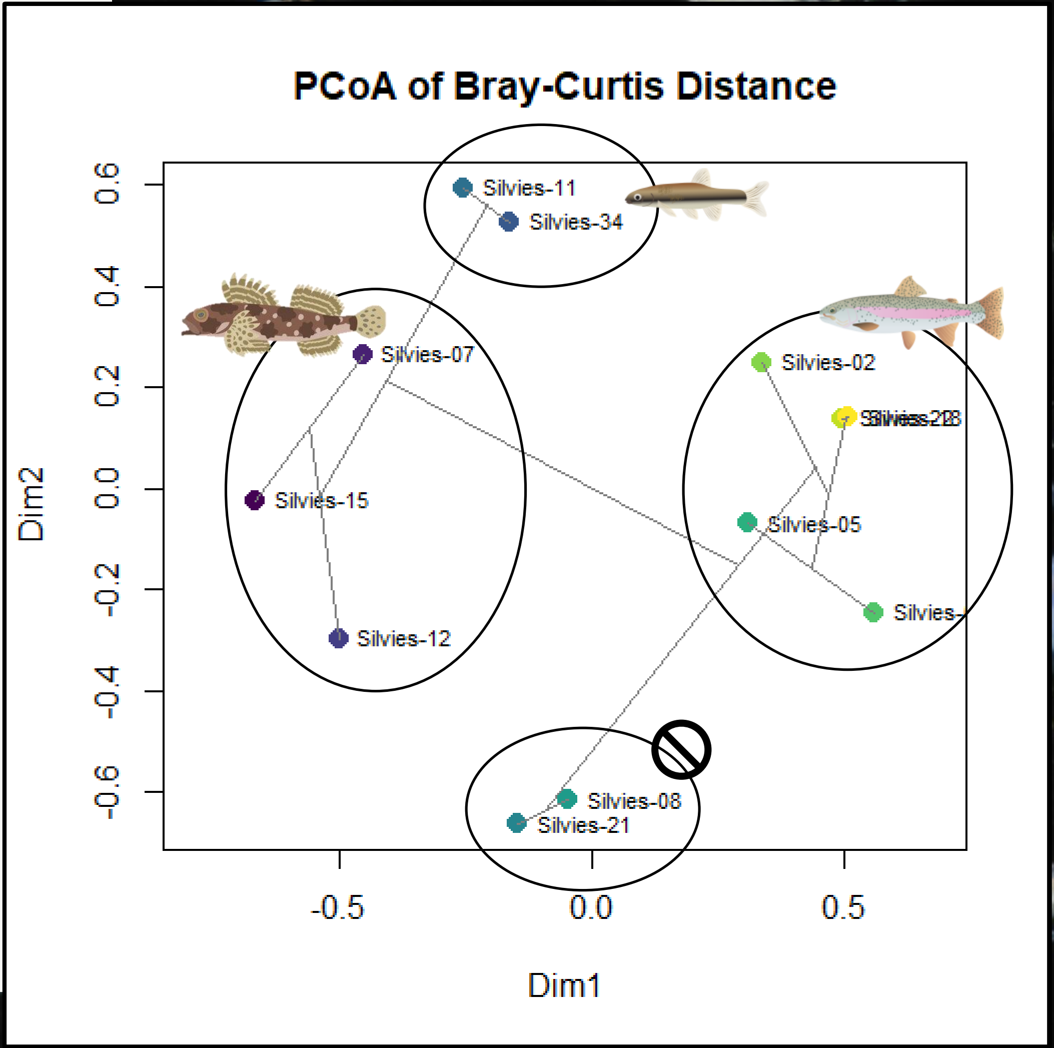


Recap: Cluster Analysis

Hierarchical cluster analysis is used to classify objects, such as species, habitats, or environmental variables, into clusters based on their similarities or dissimilarities.

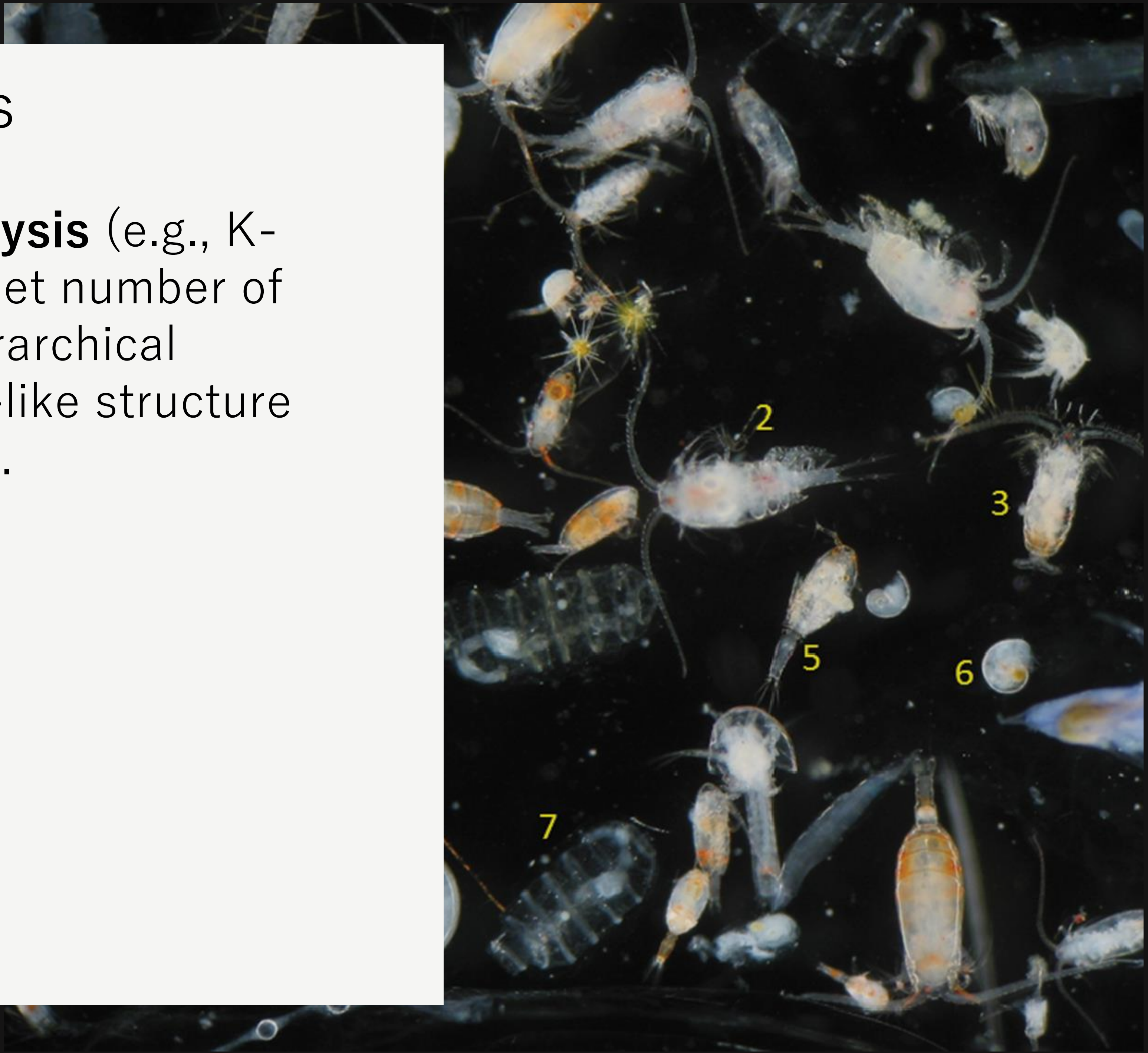
This technique helps ecologists to identify natural groupings and patterns within ecological data.





Recap: Cluster Analysis

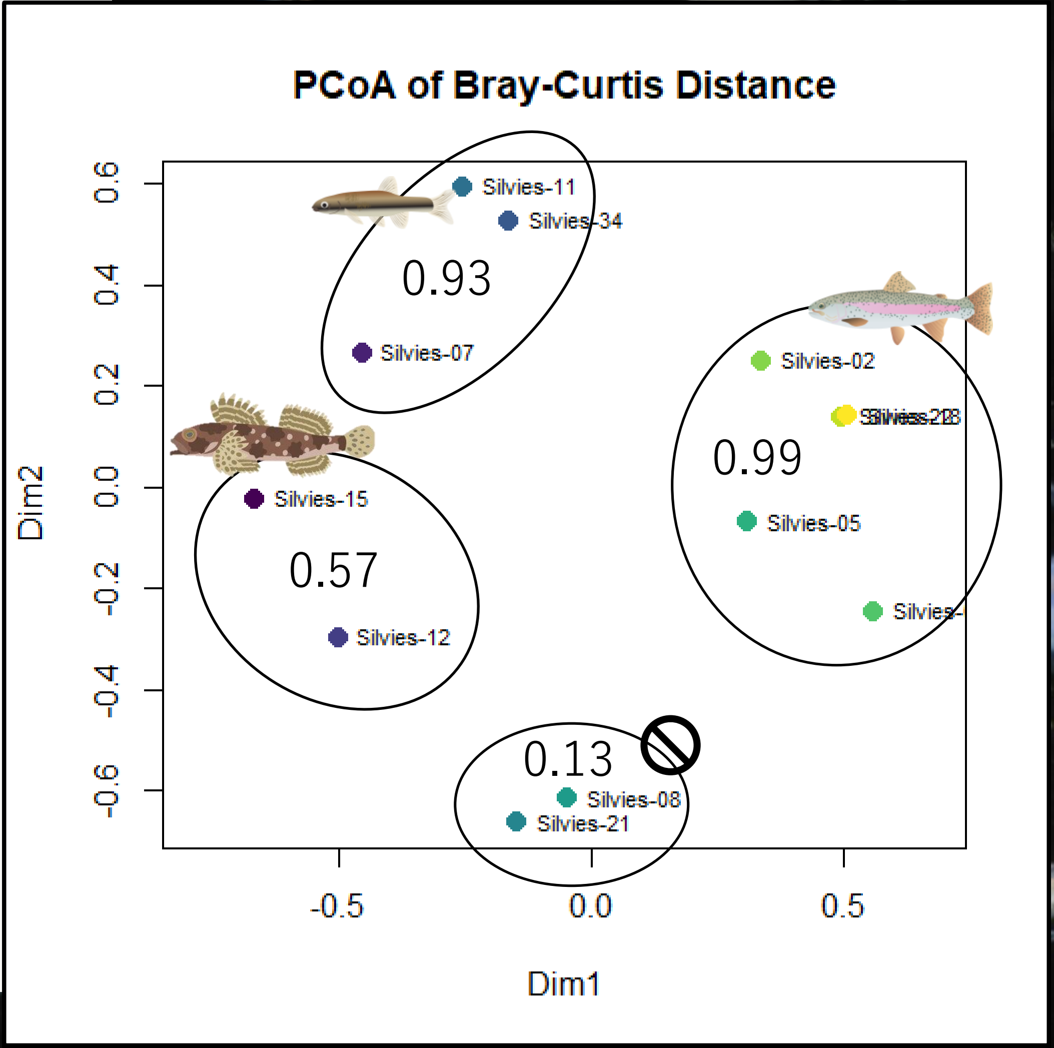
Non-hierarchical cluster analysis (e.g., K-means) groups objects into a set number of clusters. It's different from hierarchical clustering, which builds a tree-like structure to represent data relationships.



Recap: Cluster Analysis

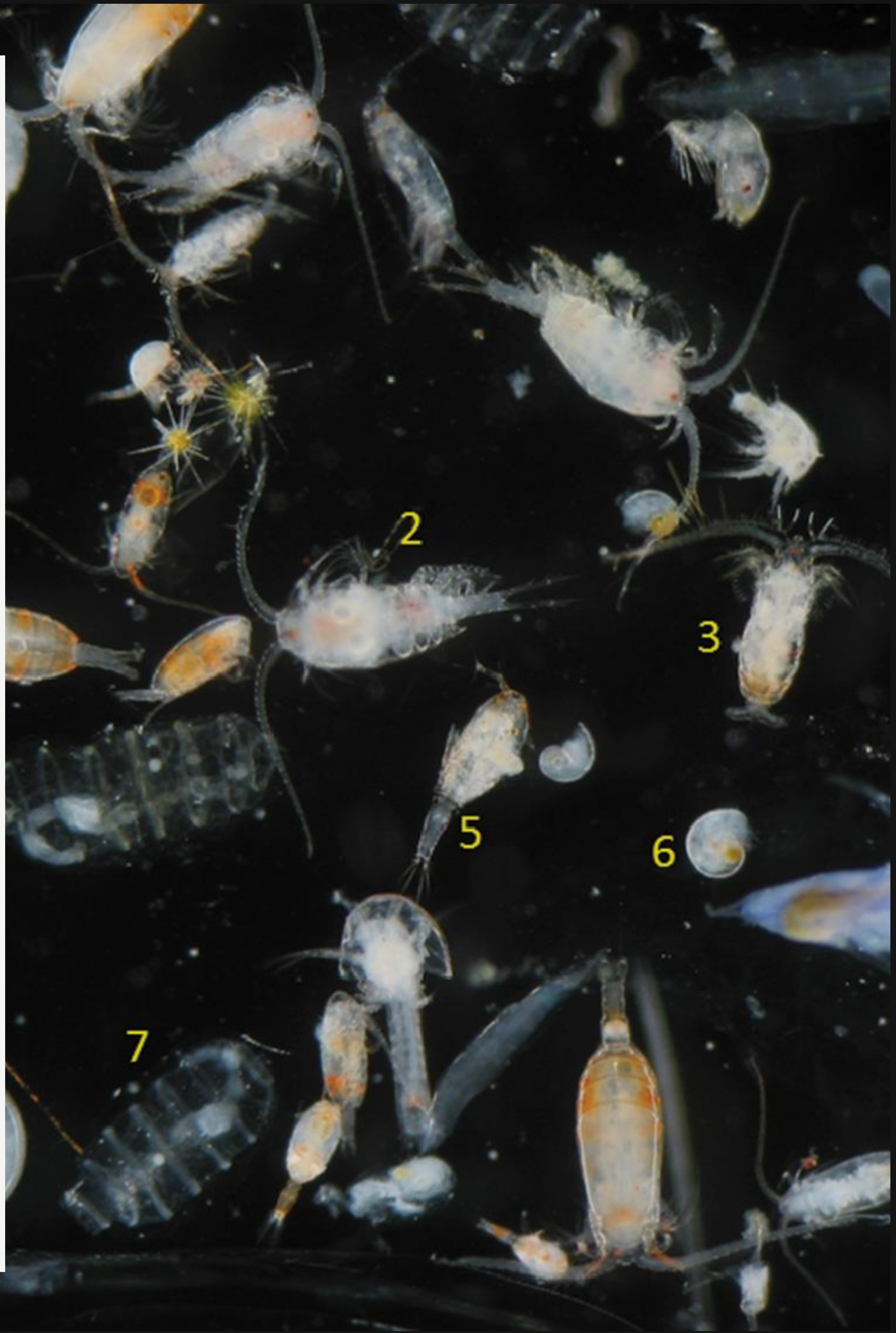
Non-hierarchical cluster analysis (e.g., K-means) groups objects into a set number of clusters. It's different from hierarchical clustering, which builds a tree-like structure to represent data relationships.

K-means partitioning or clustering is a method used to partition data into *k* clusters by minimizing within-cluster sum of squares error.

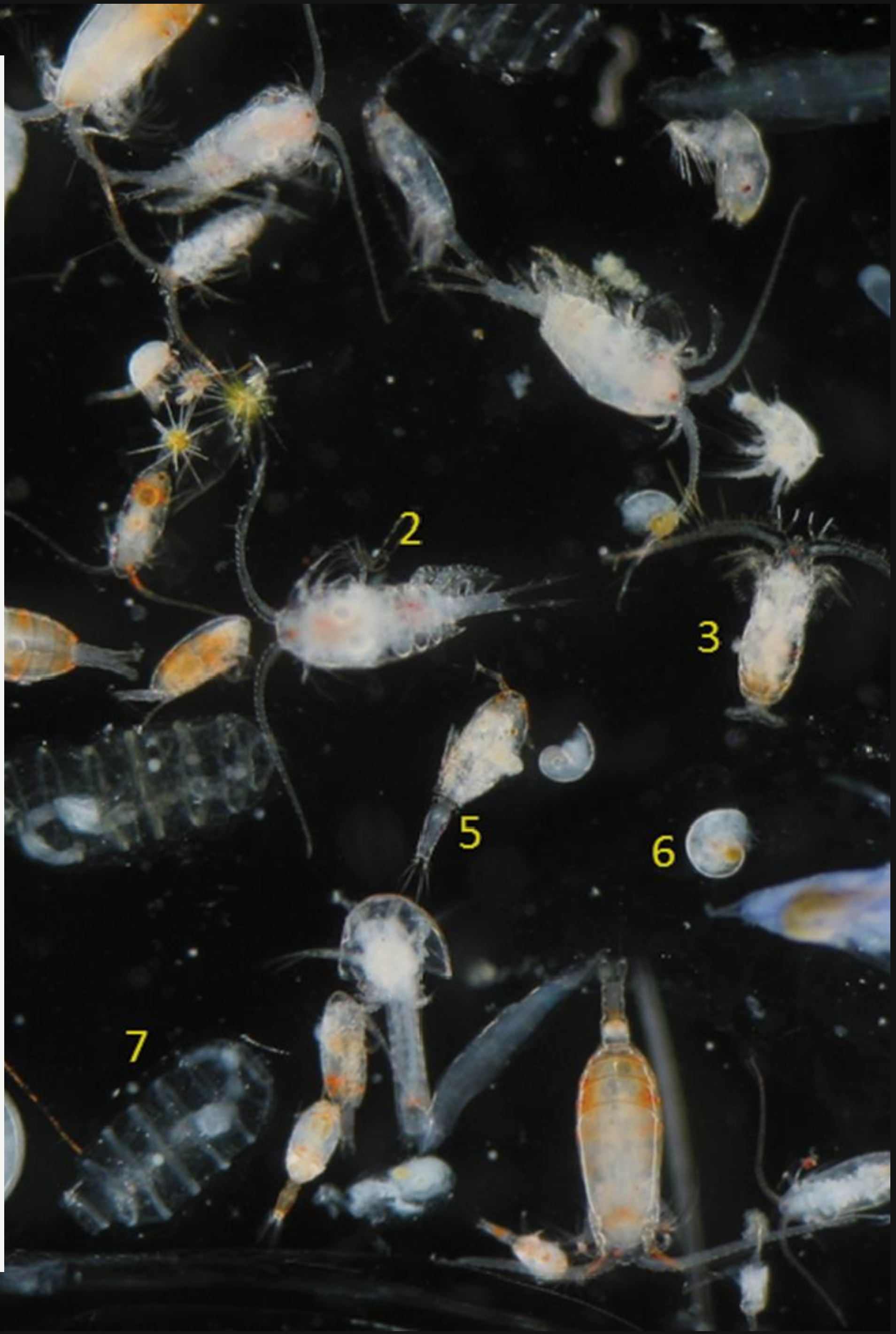


Recap: Cluster Analysis

Cluster analysis is a form of **unsupervised classification**.

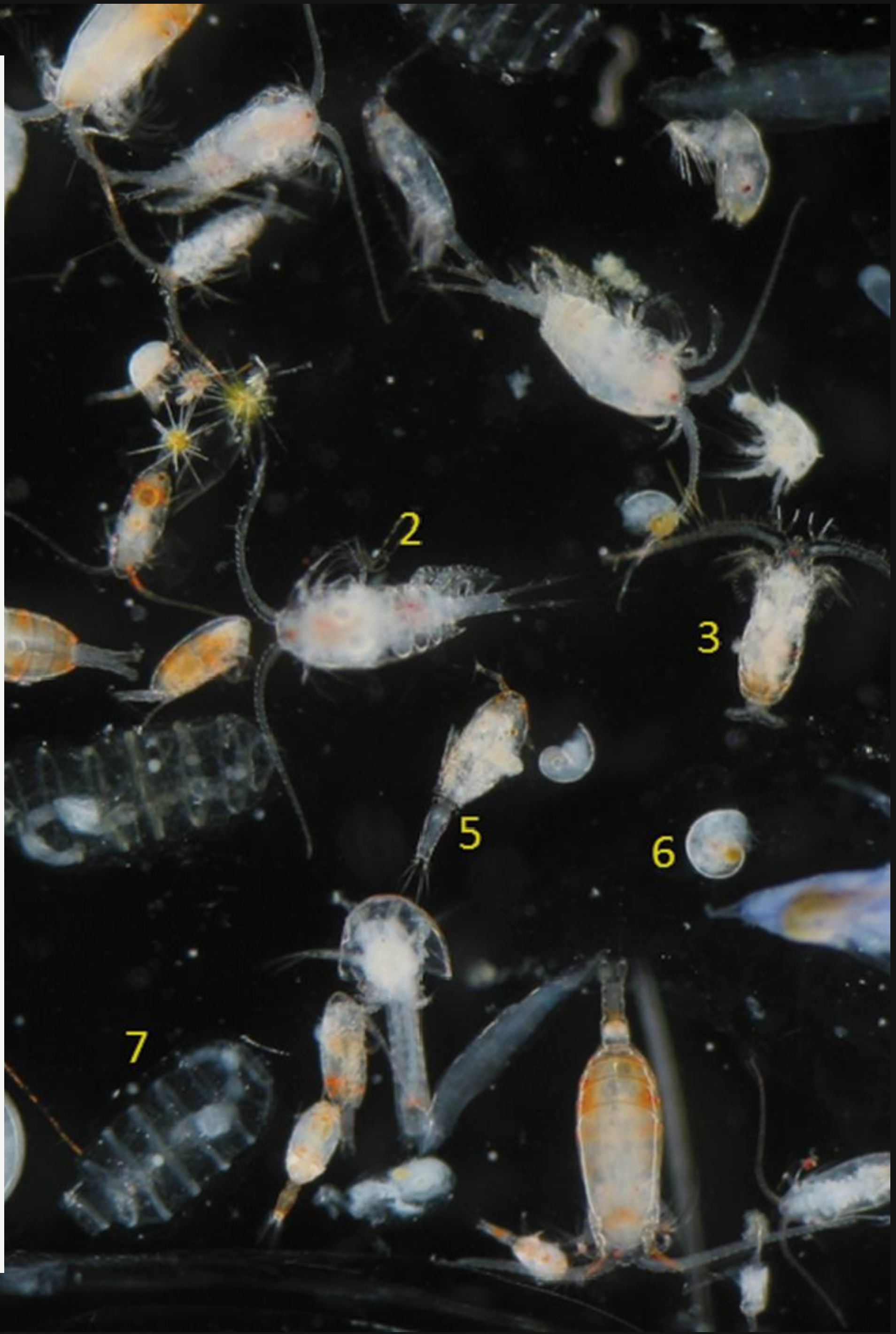


Supervised and Unsupervised Classification



Supervised and Unsupervised Classification

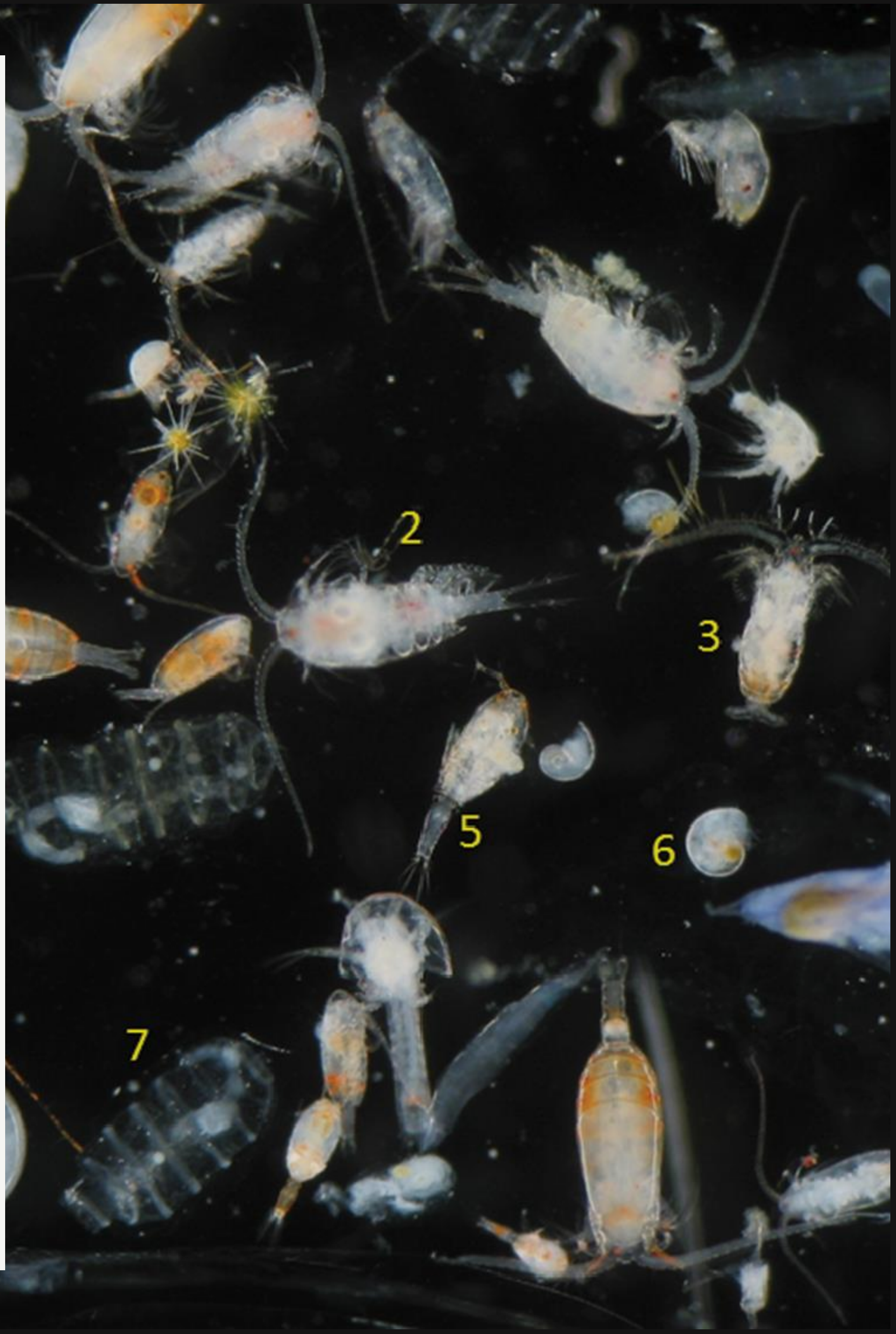
In **unsupervised classification**, the algorithm is trained on data that <u>does not</u> have pre-determined groupings. The goal is to find patterns, structures, or groups in the data based on similarities among data points.



Supervised and Unsupervised Classification

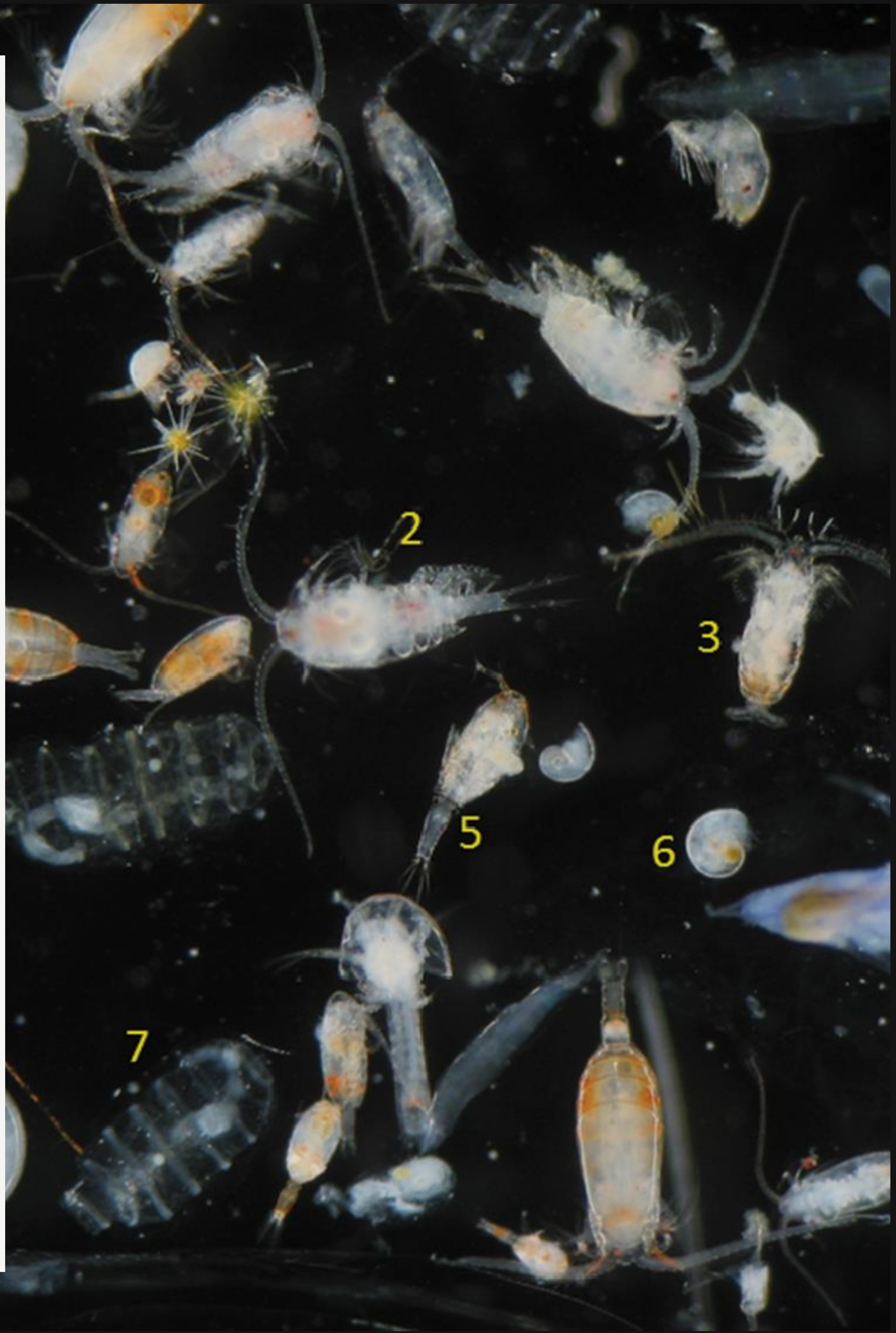
In **unsupervised classification**, the algorithm is trained on data that <u>does not</u> have pre-determined groupings. The goal is to find patterns, structures, or groups in the data based on similarities among data points.

- **Hierarchical clustering**
- **K-means clustering**
- Gaussian mixture models



Supervised and Unsupervised Classification

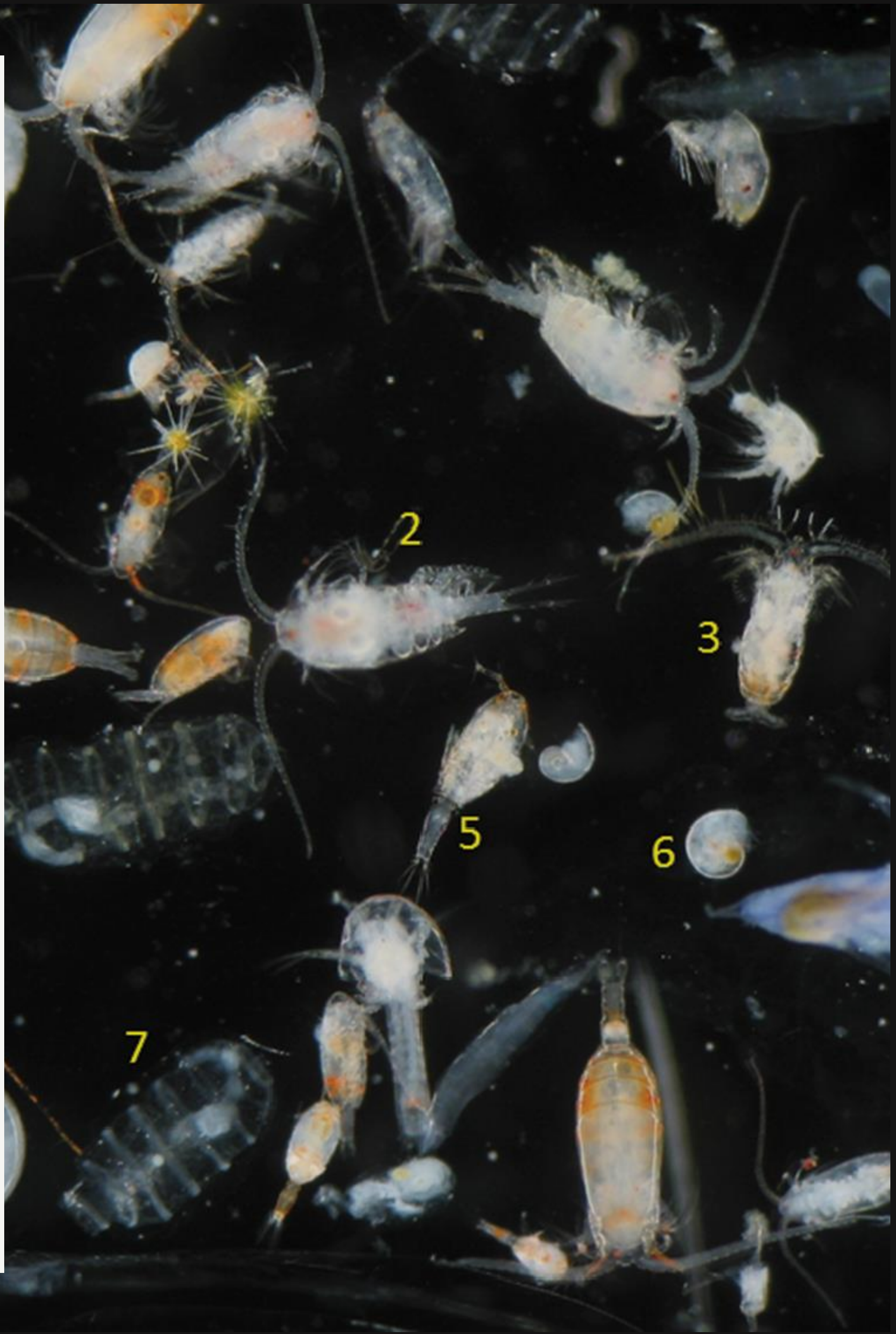
In **supervised classification**, the algorithm is <u>trained on a labeled dataset</u>, which means that each input data point is associated with a corresponding output classification. The goal is to classify new, unseen data based on this learning.



Supervised and Unsupervised Classification

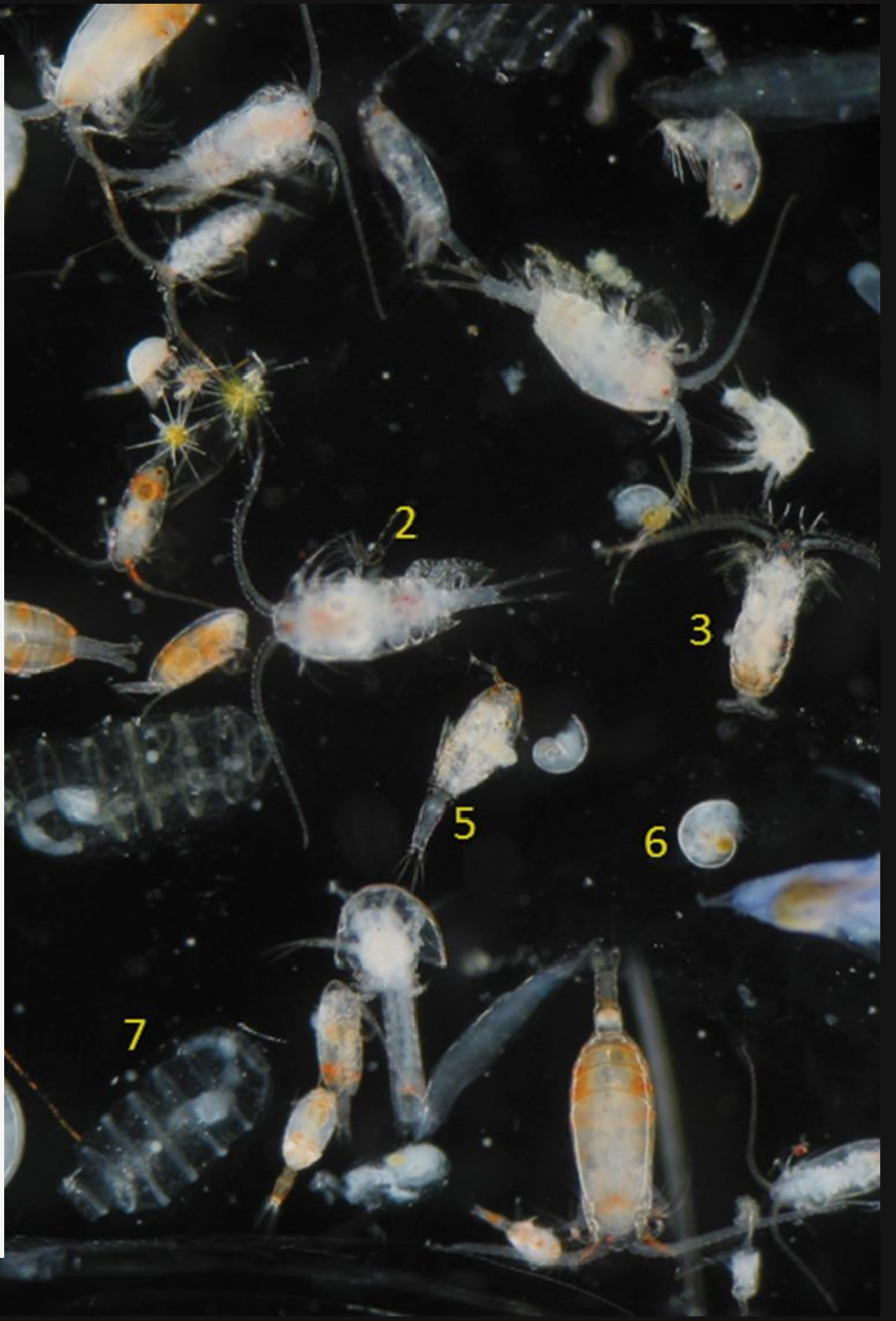
In **supervised classification**, the algorithm is trained on a labeled dataset, which means that each input data point is associated with a corresponding output classification. The goal is to classify new, unseen data based on this learning.

- **Decision Trees**
- **Random Forests**
- Neural Networks

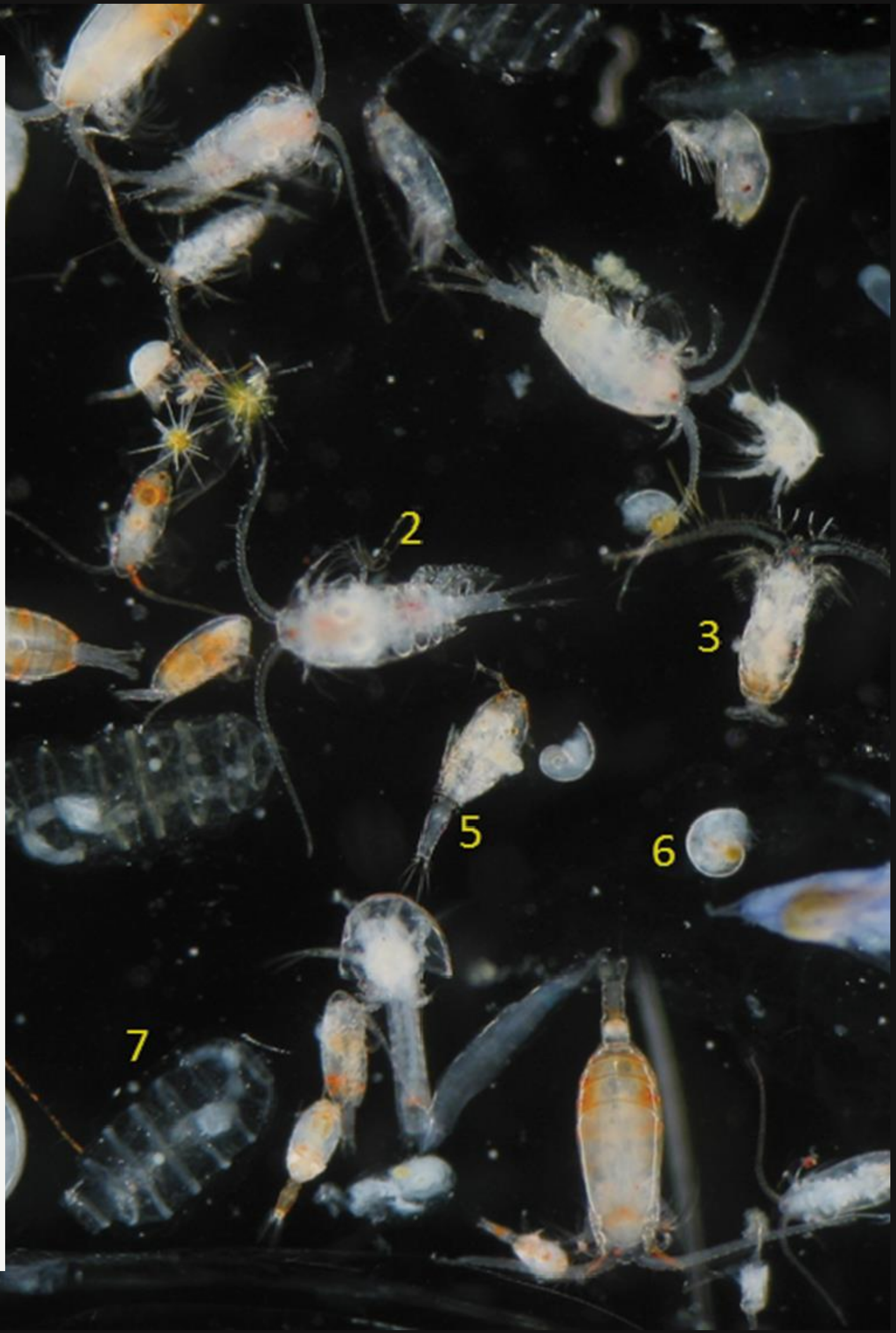


Supervised and Unsupervised Classification

Aspect	Supervised Classification	Unsupervised Classification
Data Labels	Labeled data (input-output pairs)	Unlabeled data (only input data, no labels)
Goal	Learn a mapping from input to output labels	Discover structure, patterns, or clusters in the data
Use Case	Predict or classify new data based on learned relationships	Group or cluster similar data points without prior knowledge
Output	Discrete classes or continuous values	Clusters or groups of similar data points
Algorithms	**Decision Trees**, **Random Forests**, Neural Networks, etc.	**K-Means**, **Hierarchical Clustering**, etc.
Training Complexity	Often more computationally intensive (depends on labeled data)	Computationally less intensive but can be harder to interpret

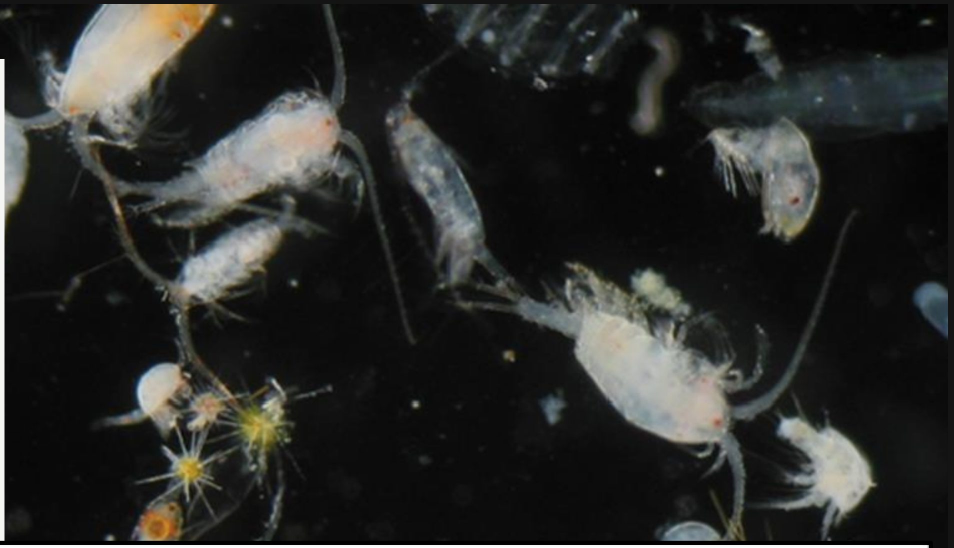


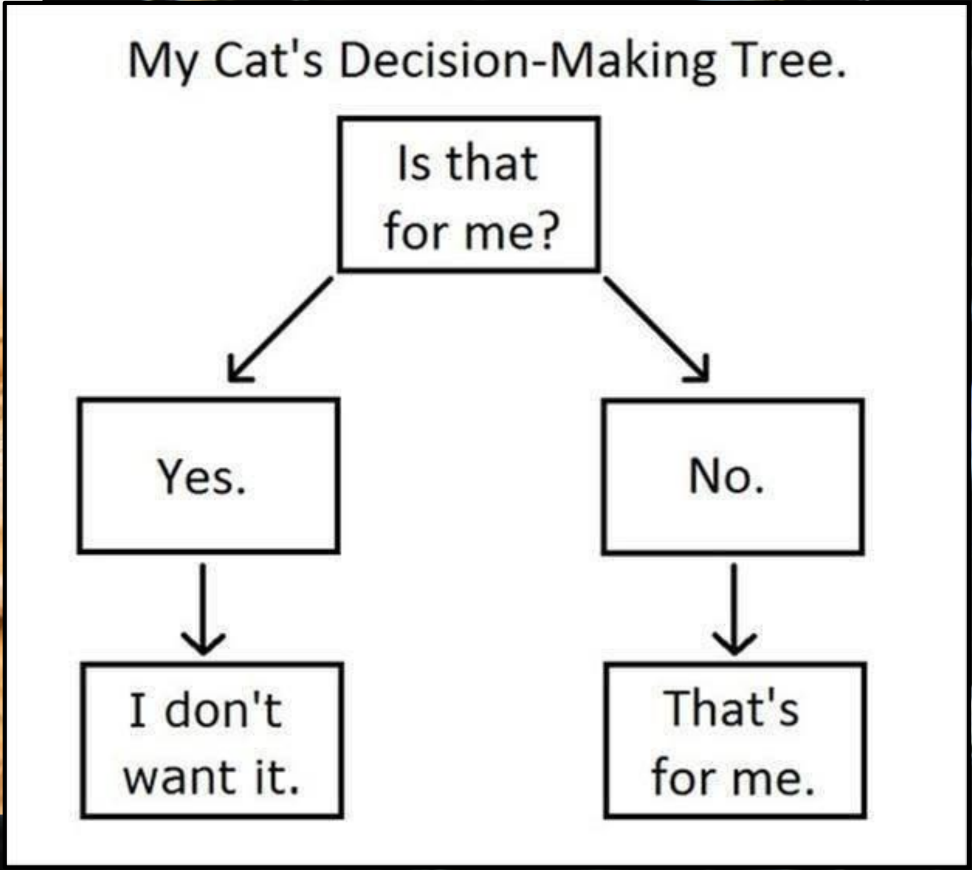
Classification and Regression Trees



Classification and Regression Trees

Classification and Regression Trees (CART) are decision tree algorithms used for classification (discrete outcomes) and regression (continuous outcomes).

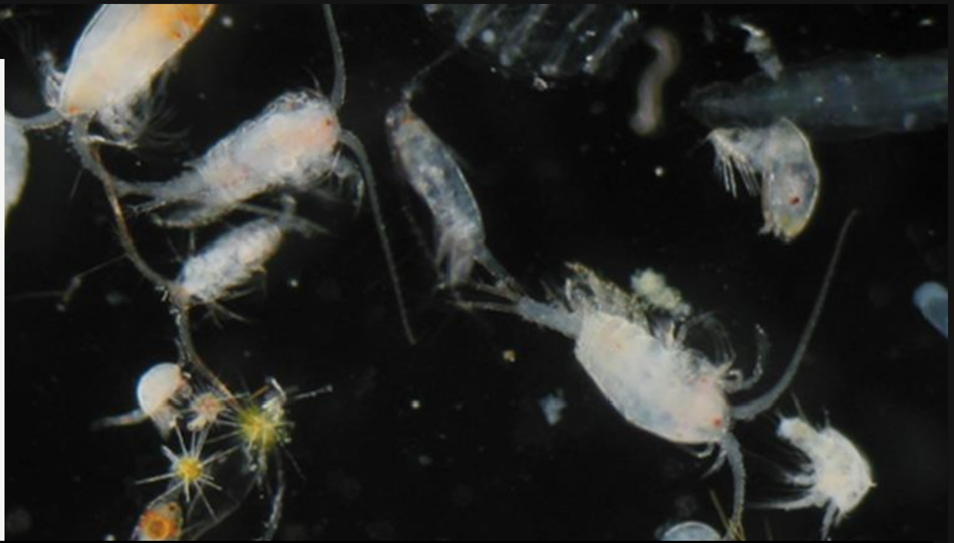


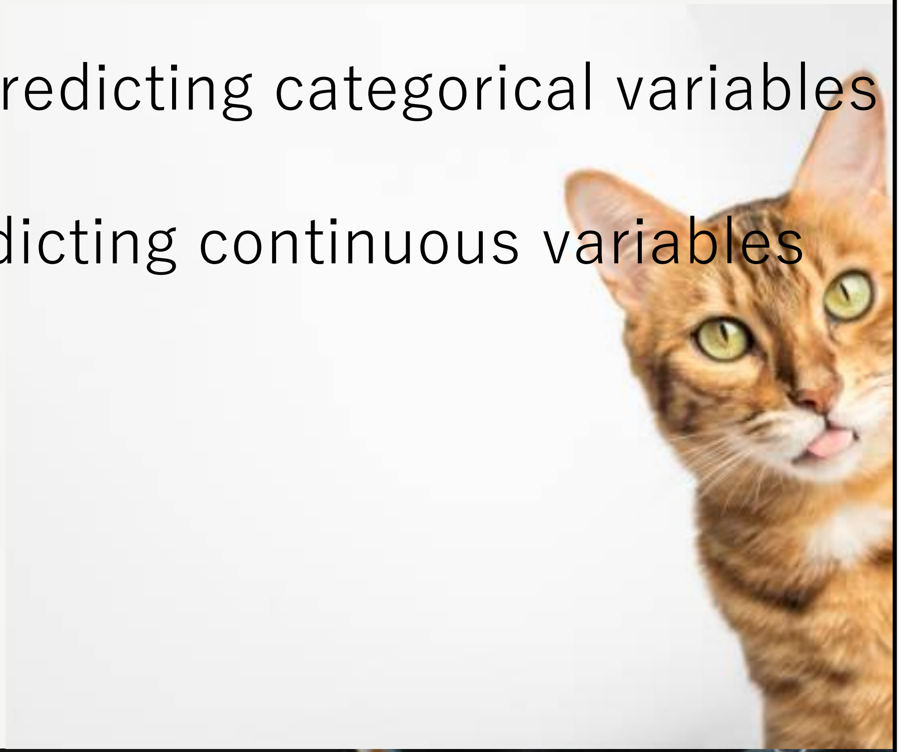


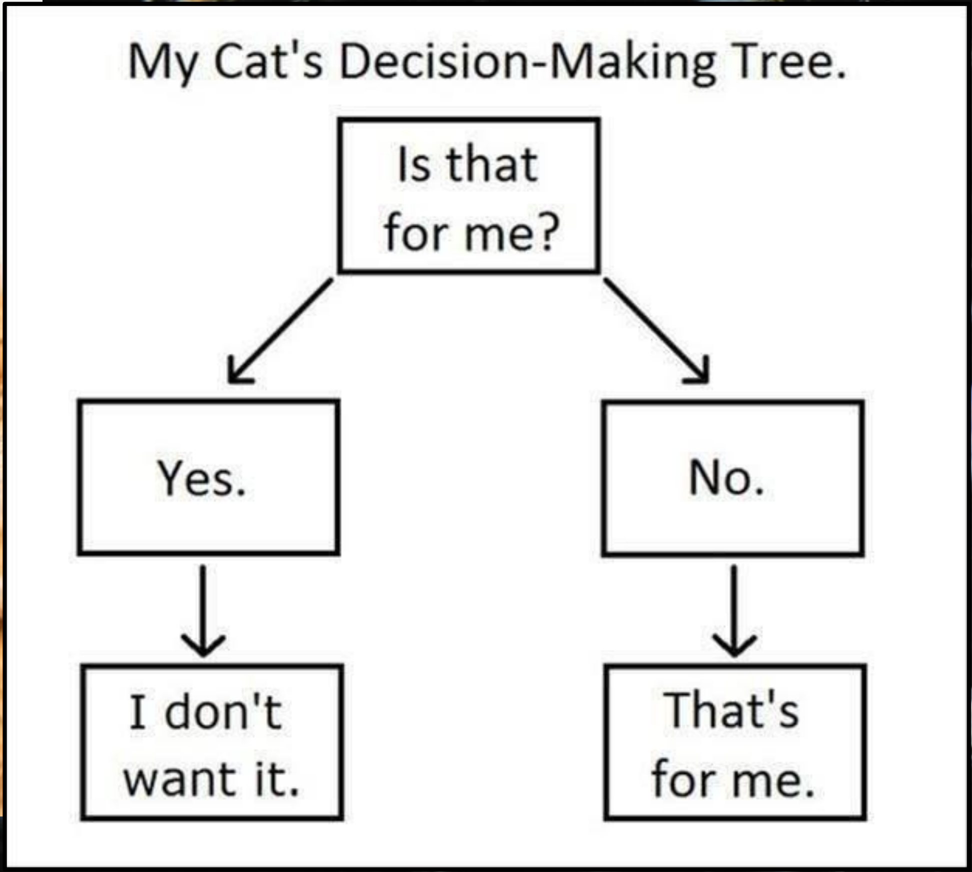
Classification and Regression Trees

Classification and Regression Trees (CART) are decision tree algorithms used for classification (discrete outcomes) and regression (continuous outcomes).

- **Classification**: predicting categorical variables
- **Regression:** predicting continuous variables

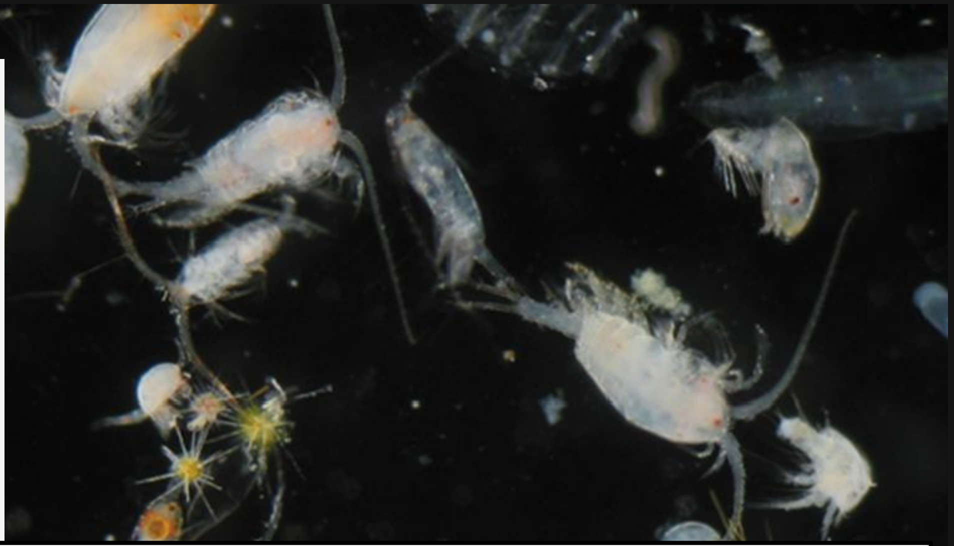


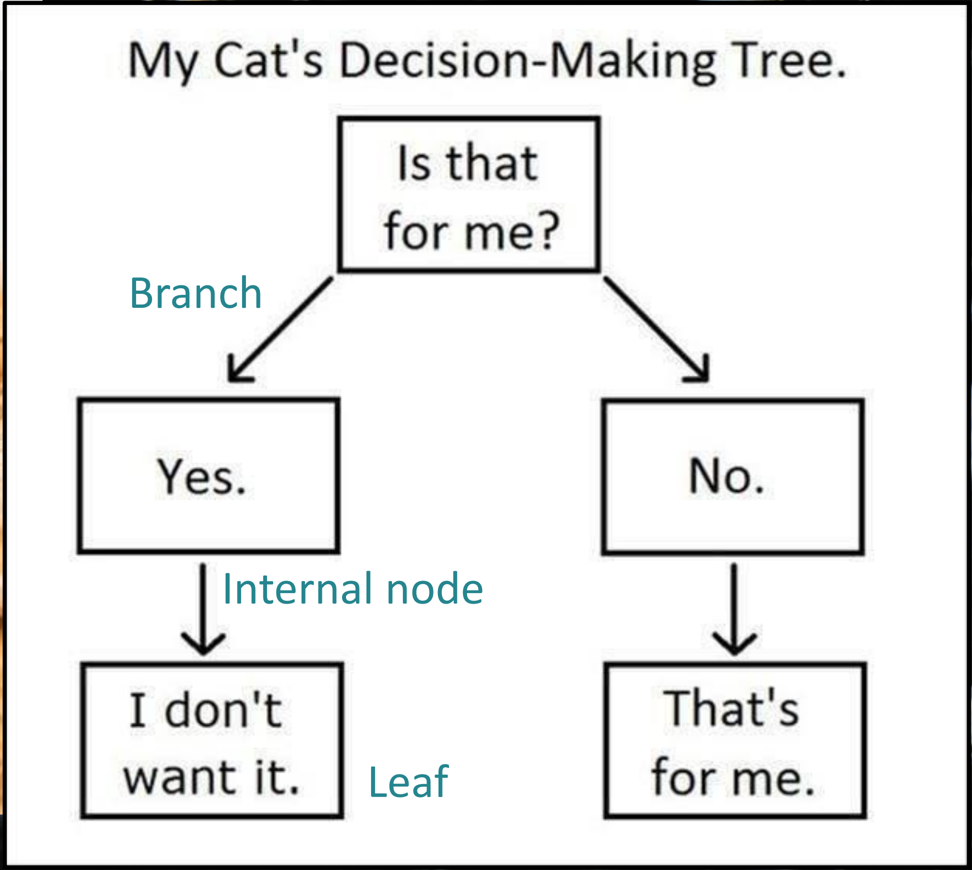




Classification and Regression Trees

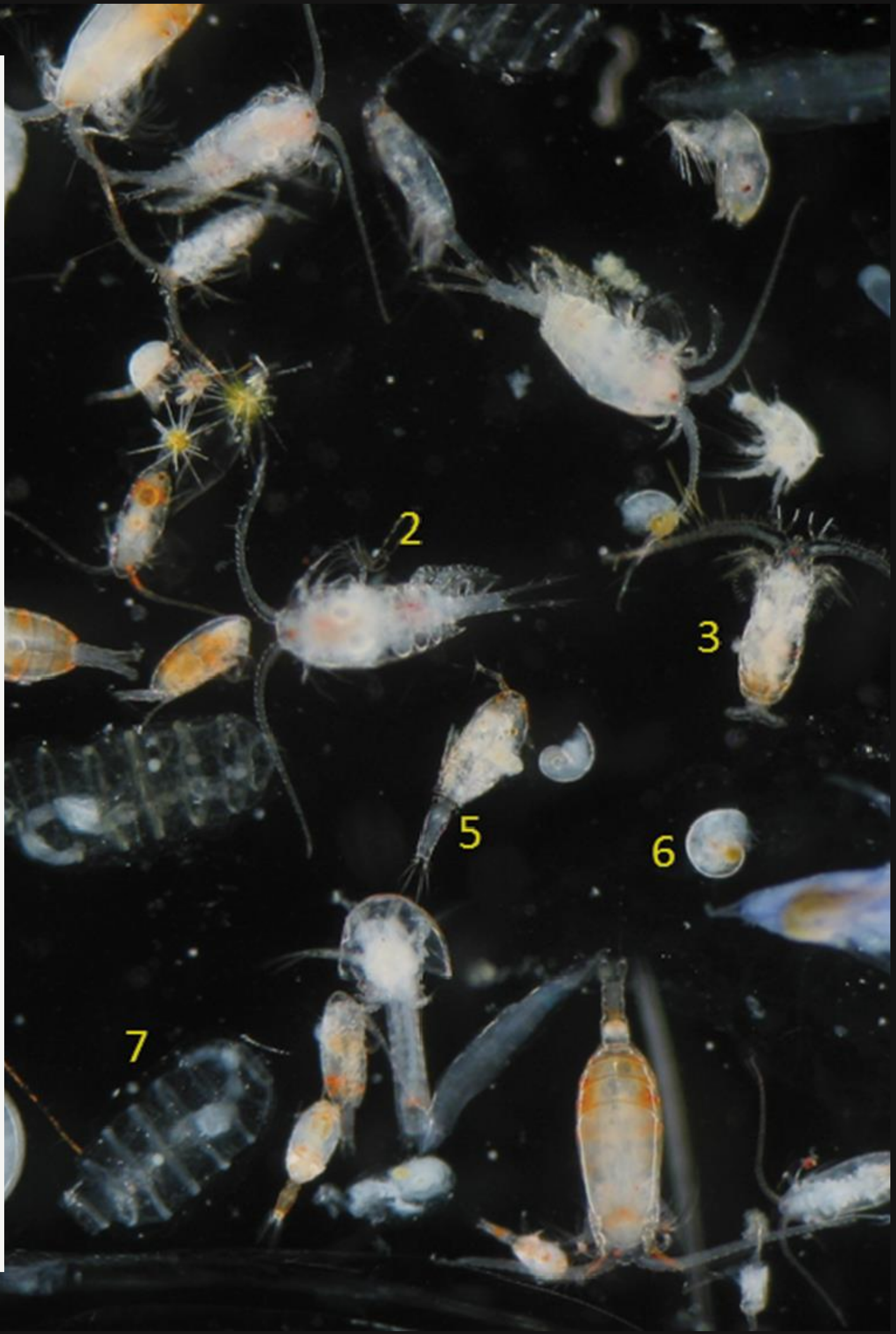
Classification and Regression Trees (CART) are decision tree algorithms used for classification (discrete outcomes) and regression (continuous outcomes).





Classification and Regression Trees: The Splitting Process

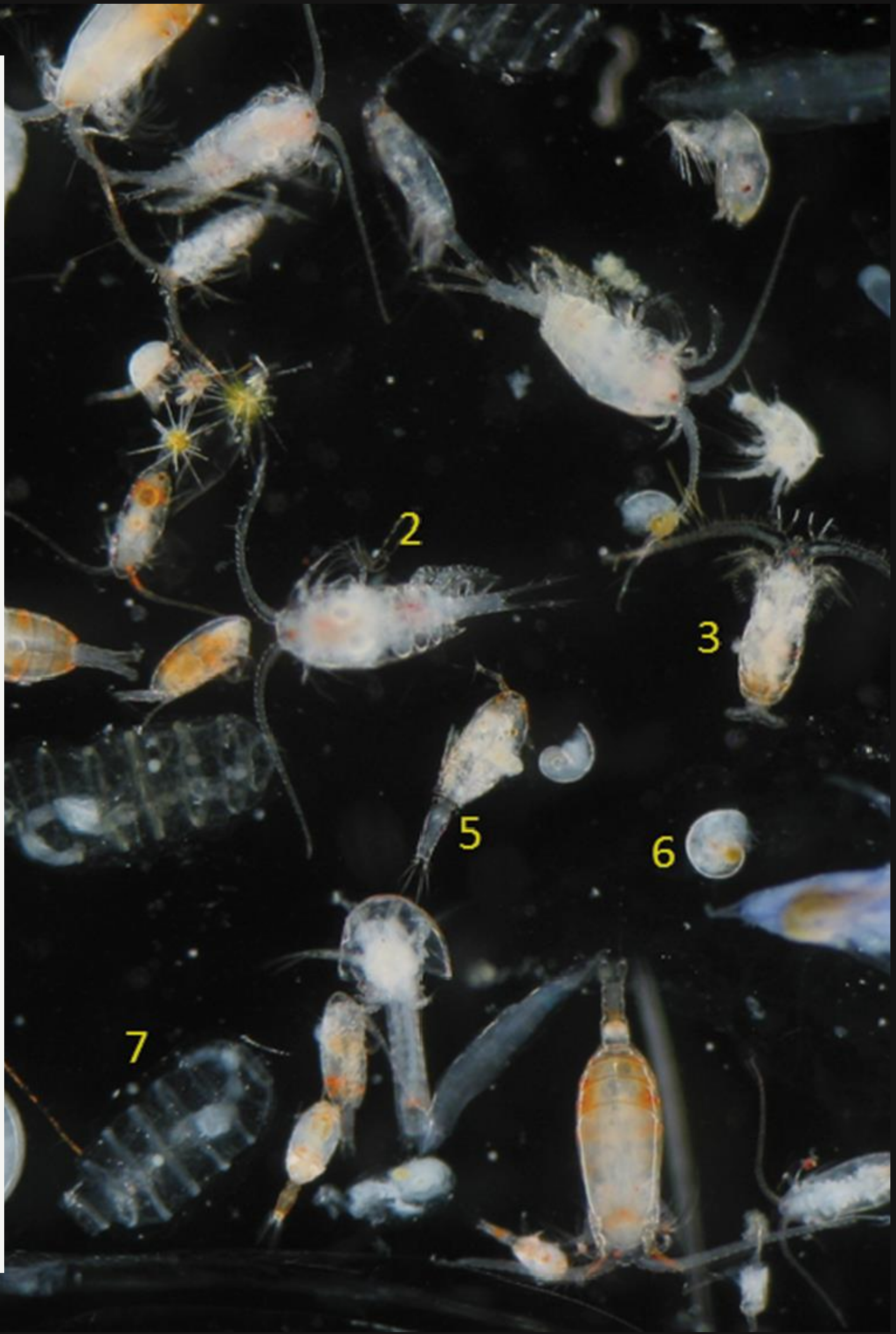
The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.



Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

For **classification**, the split is based on minimizing **impurity**.

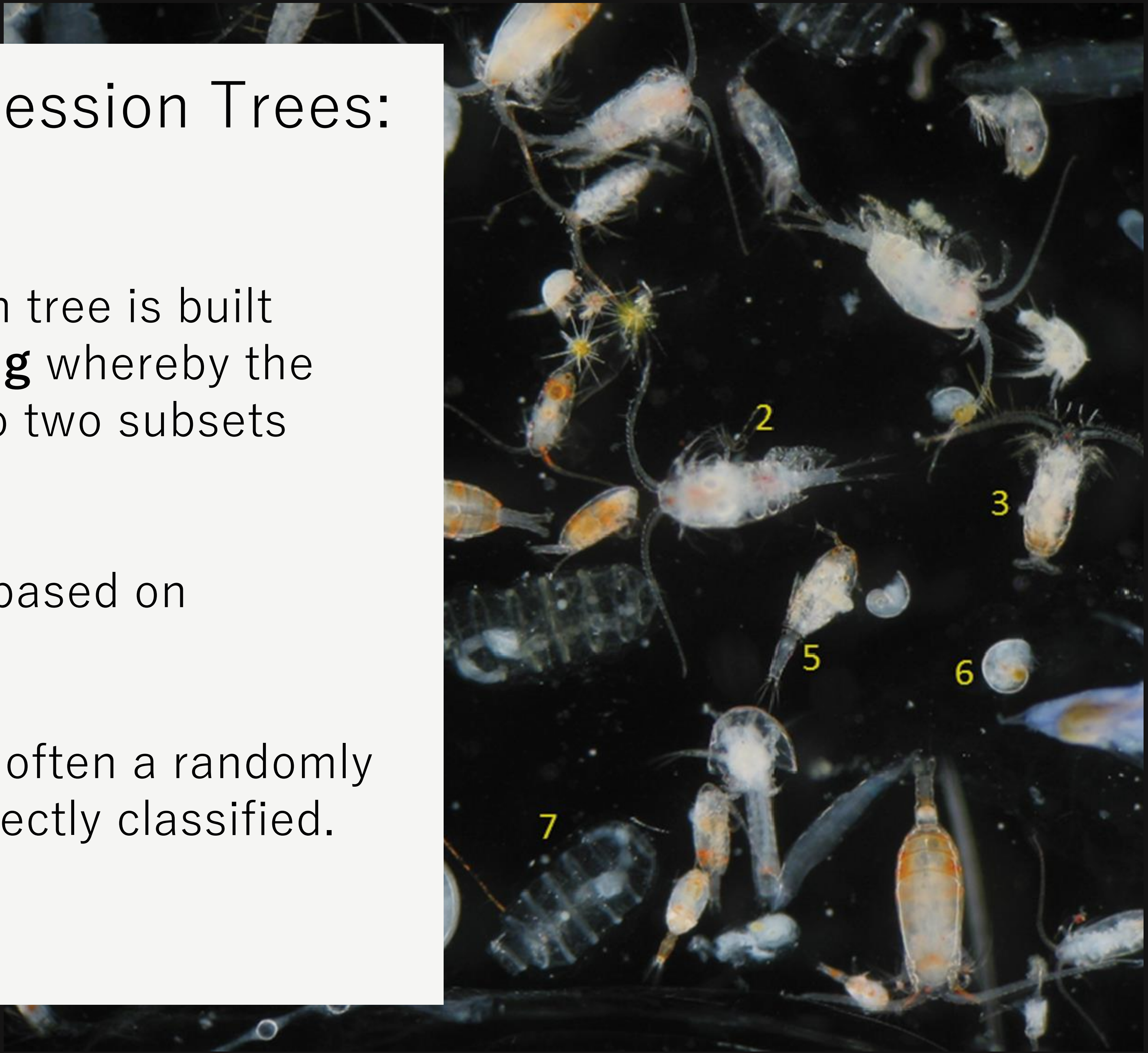


Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

For **classification**, the split is based on minimizing **impurity**.

The **Gini Index** measures how often a randomly chose element would be incorrectly classified.

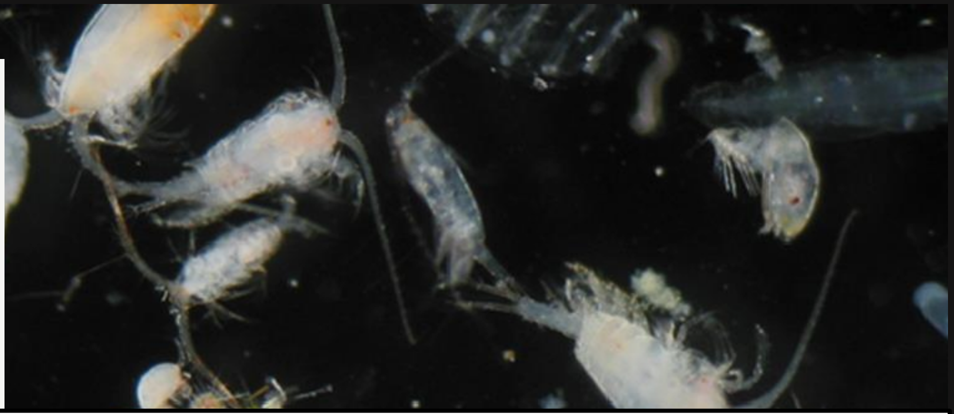


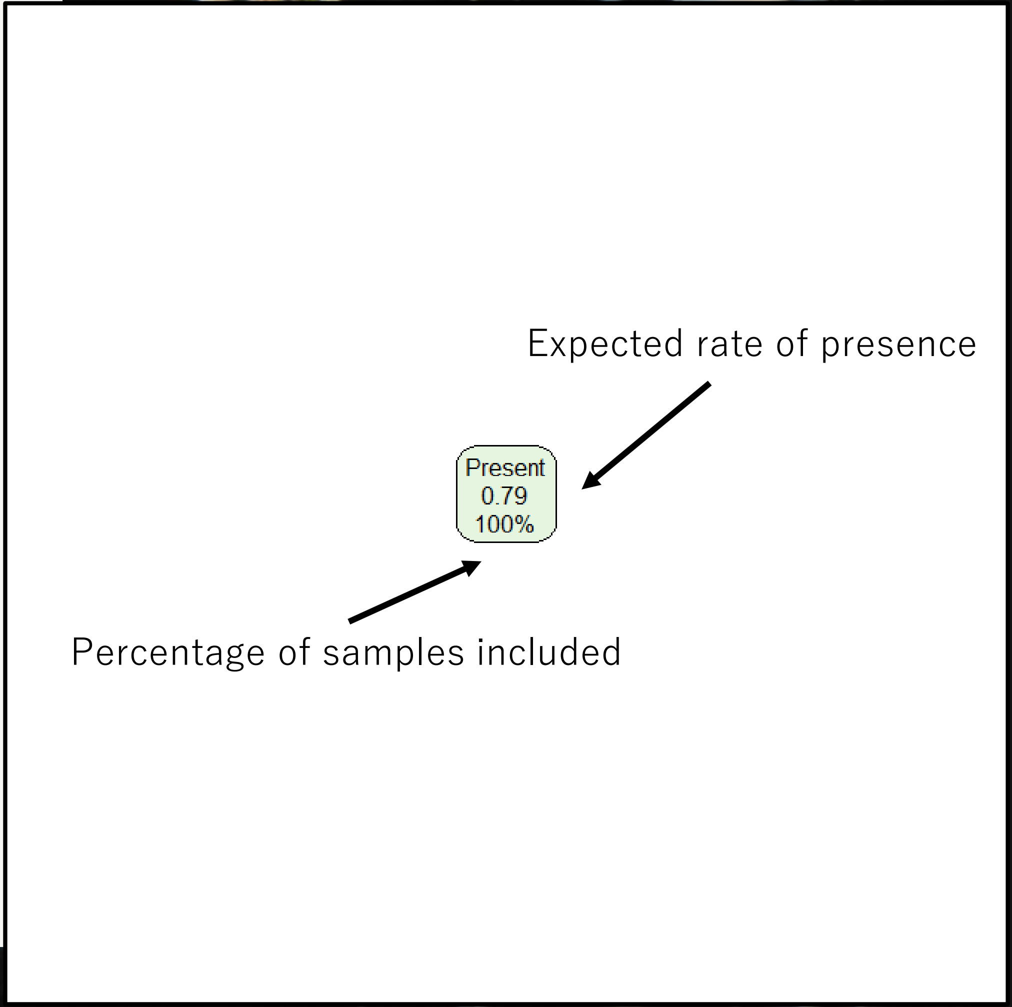
Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

For **classification**, the split is based on minimizing **impurity**.

The **Gini Index** measures how often a randomly chose element would be incorrectly classified.



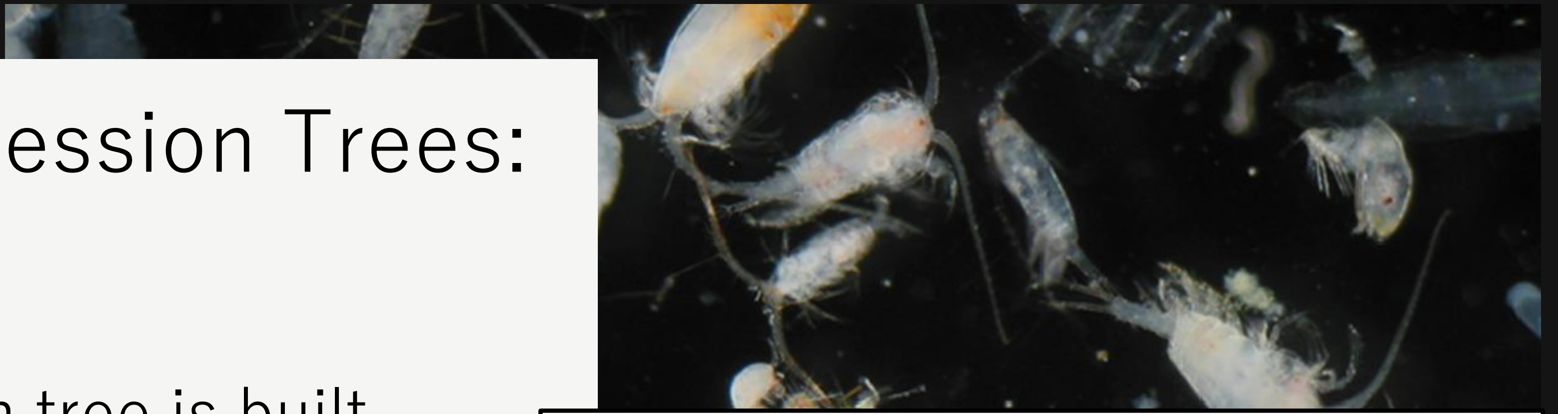


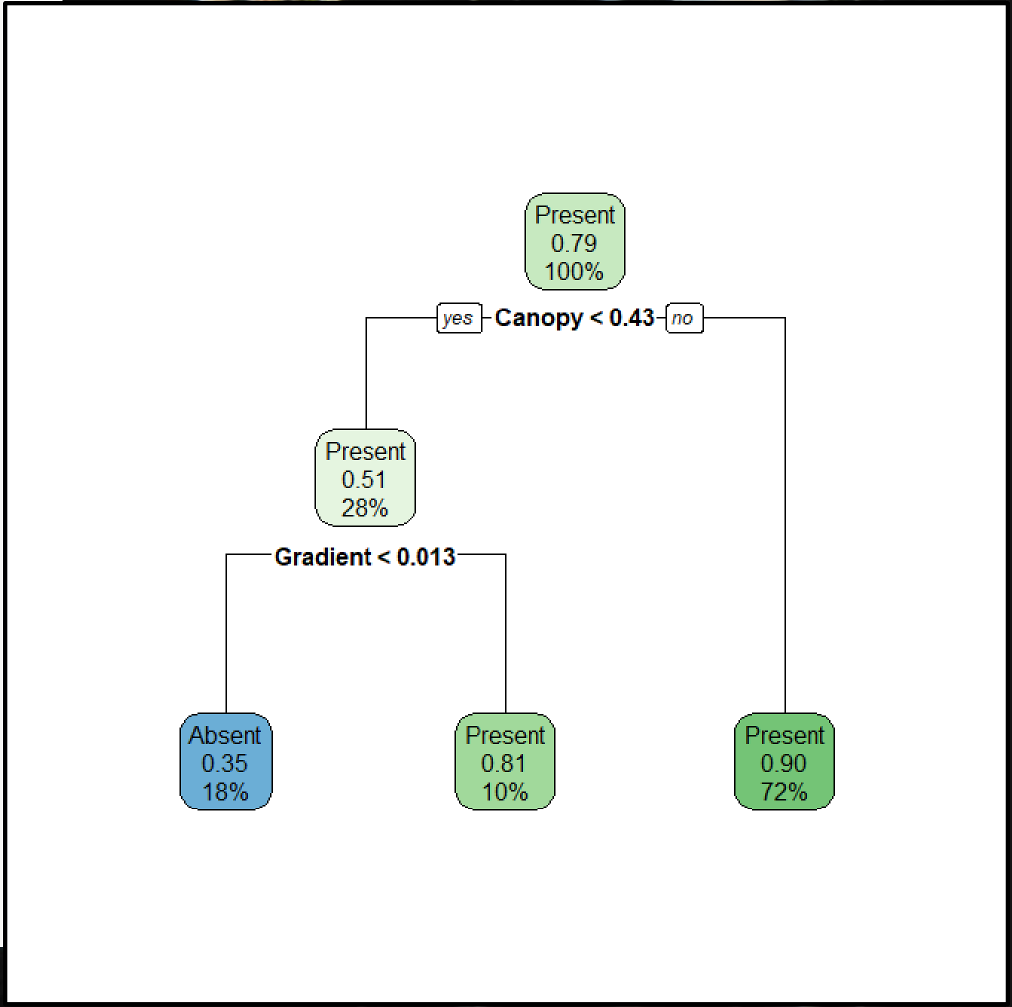
Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

For **classification**, the split is based on minimizing **impurity**.

The **Gini Index** measures how often a randomly chose element would be incorrectly classified.



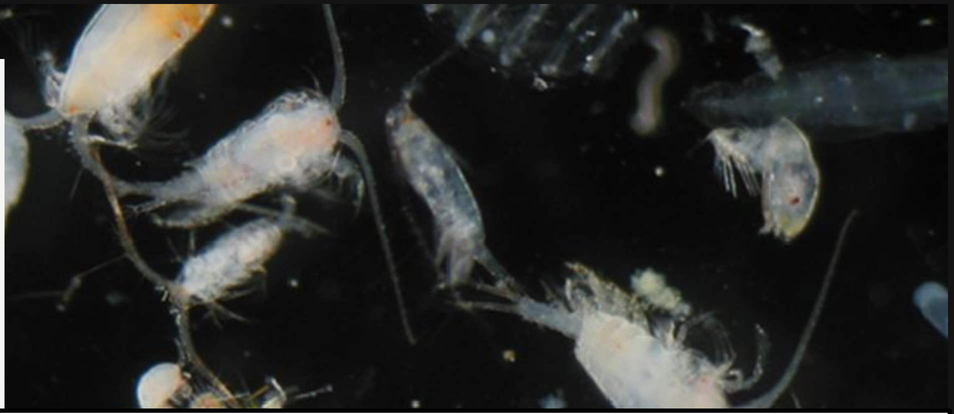


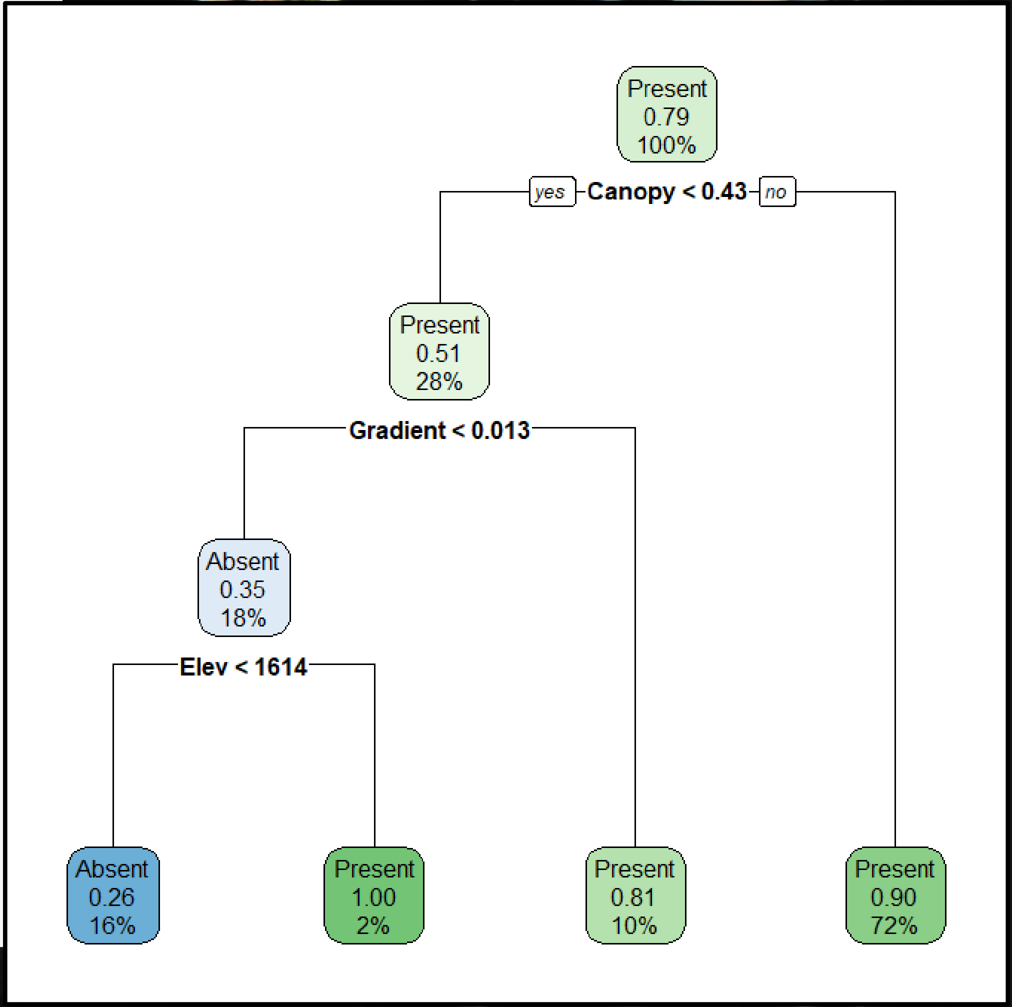
Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

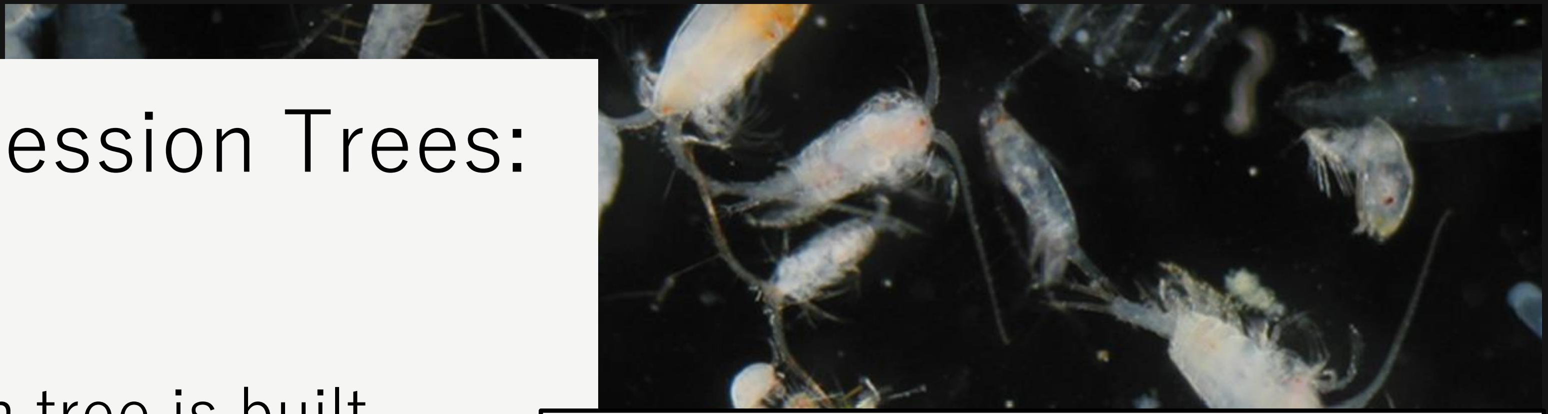
For **classification**, the split is based on minimizing **impurity**.

The **Gini Index** measures how often a randomly chose element would be incorrectly classified.





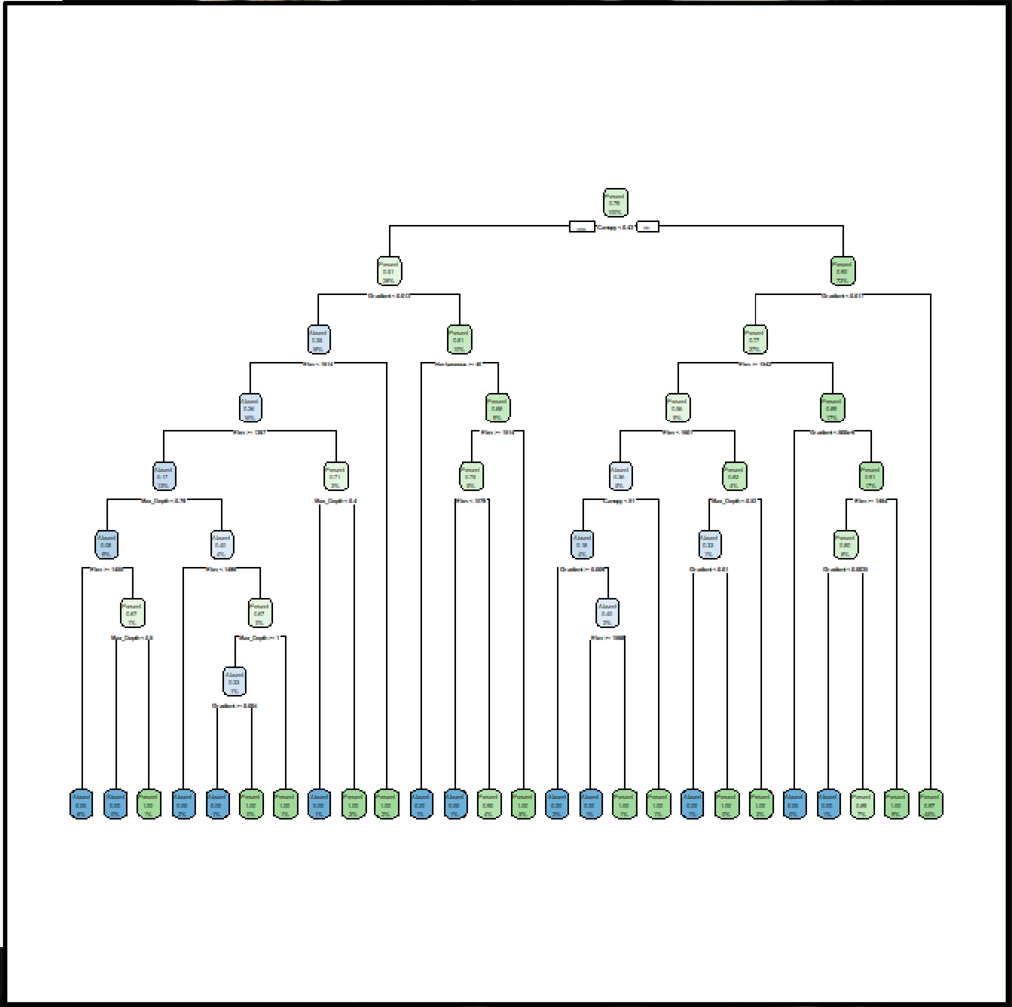
Classification and Regression Trees: The Splitting Process



The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

For **classification**, the split is based on minimizing **impurity**.

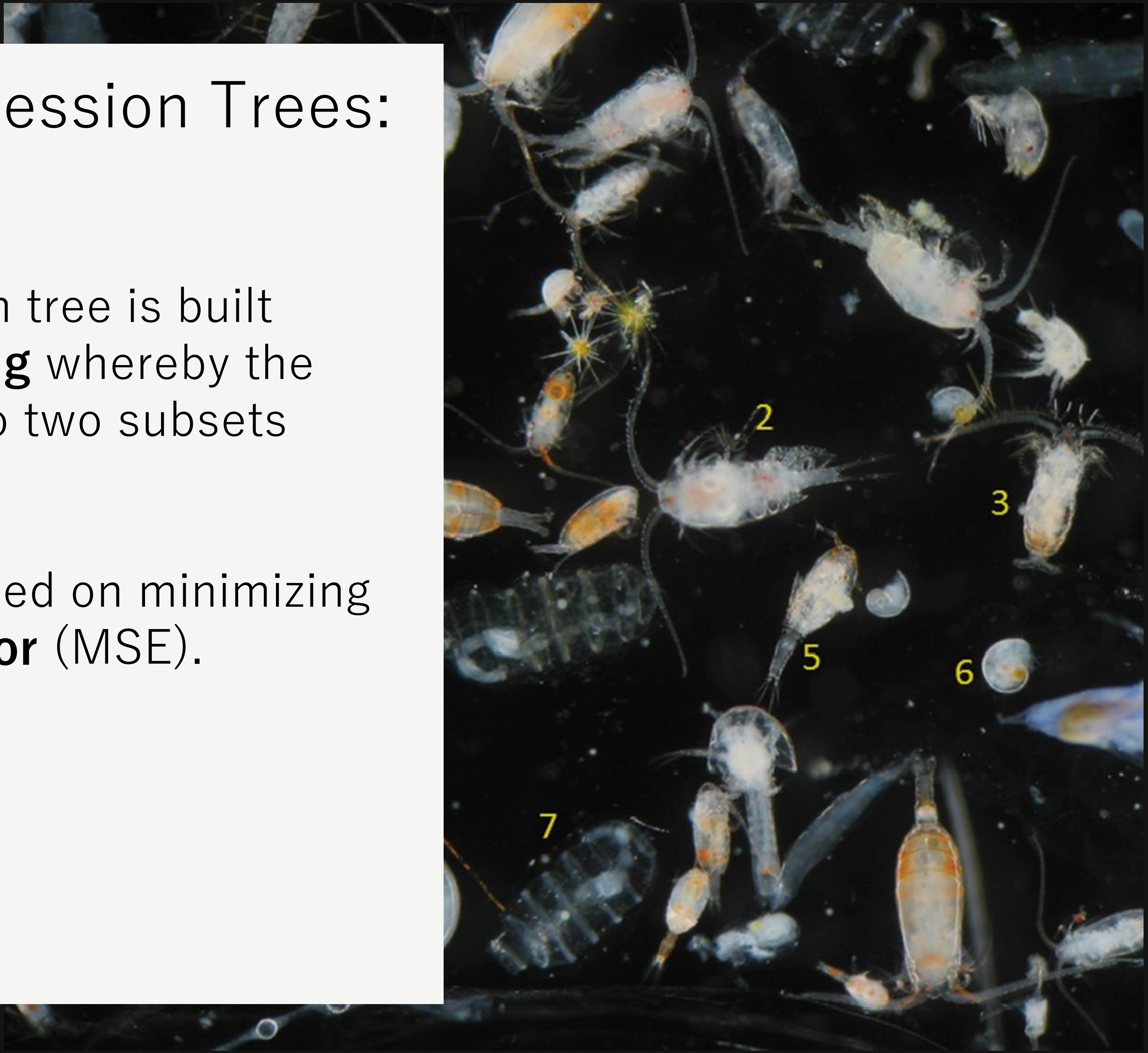
The **Gini Index** measures how often a randomly chose element would be incorrectly classified.



Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

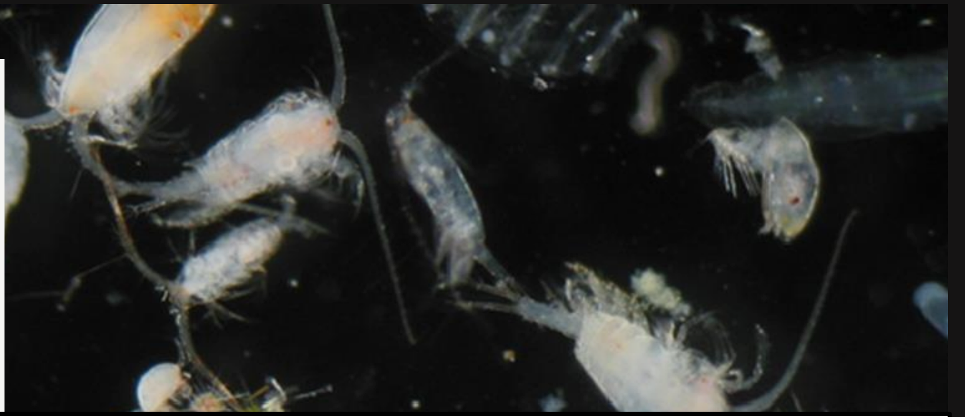
For **regression**, the split is based on minimizing variance or **mean squared error** (MSE).

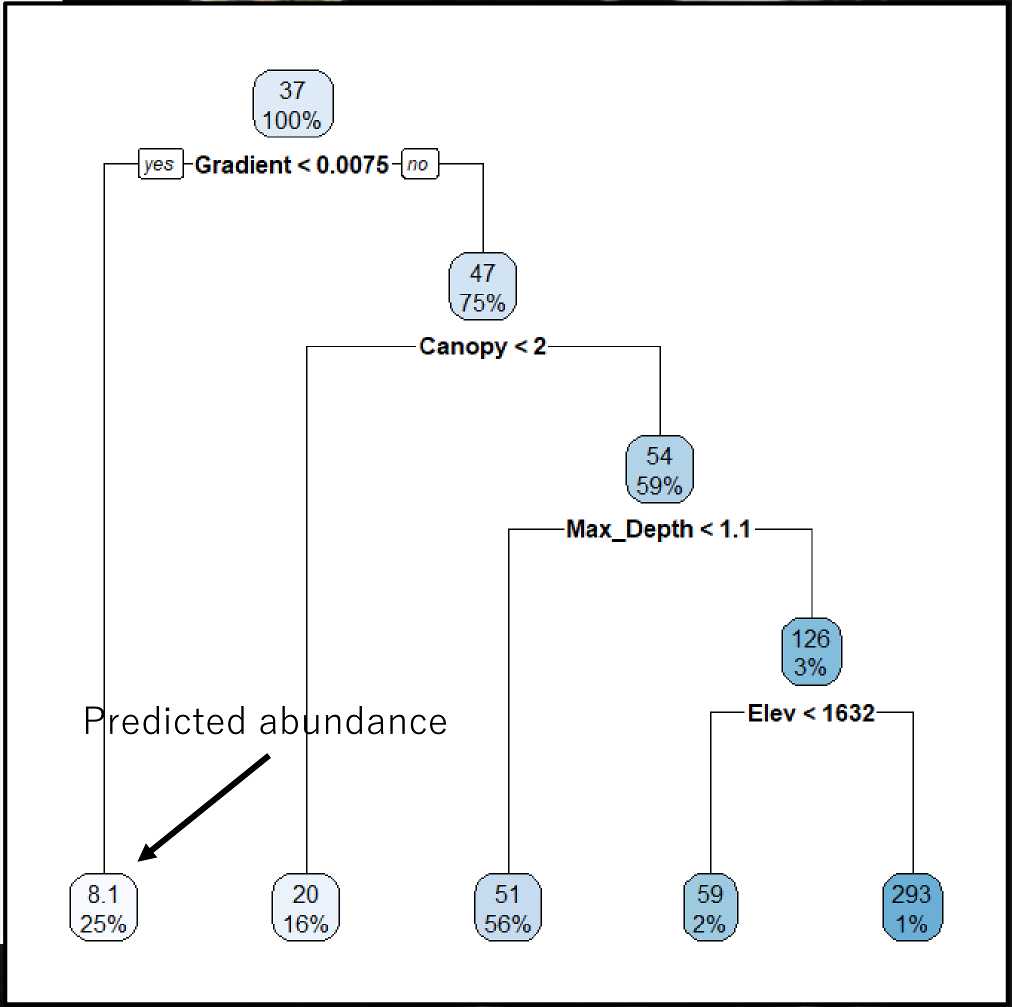


Classification and Regression Trees: The Splitting Process

The classification or regression tree is built using **recursive binary splitting** whereby the data are split at each node into two subsets based on a threshold.

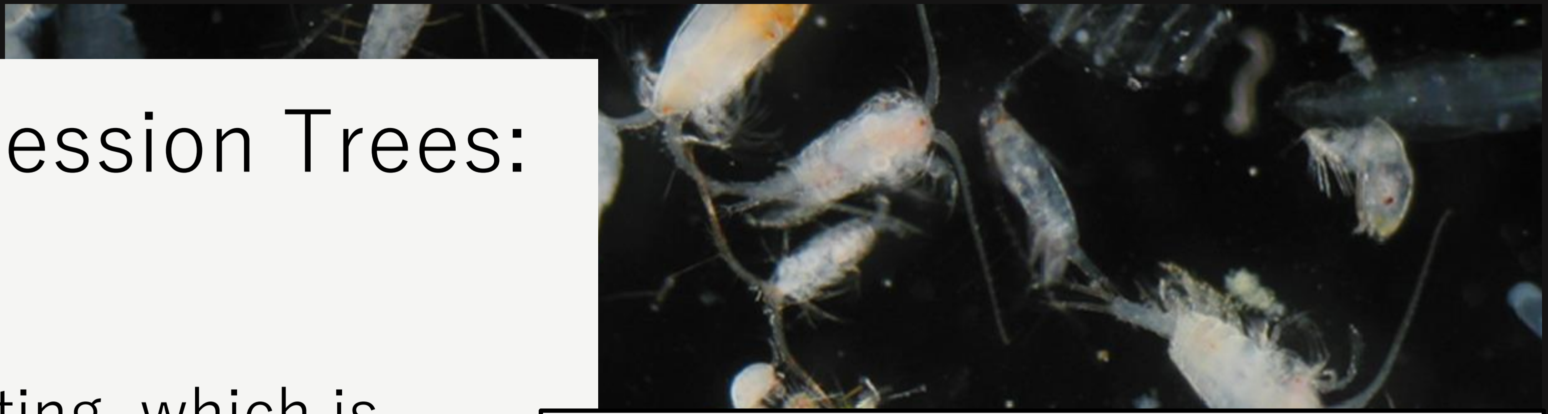
For **regression**, the split is based on minimizing variance or **mean squared error** (MSE).

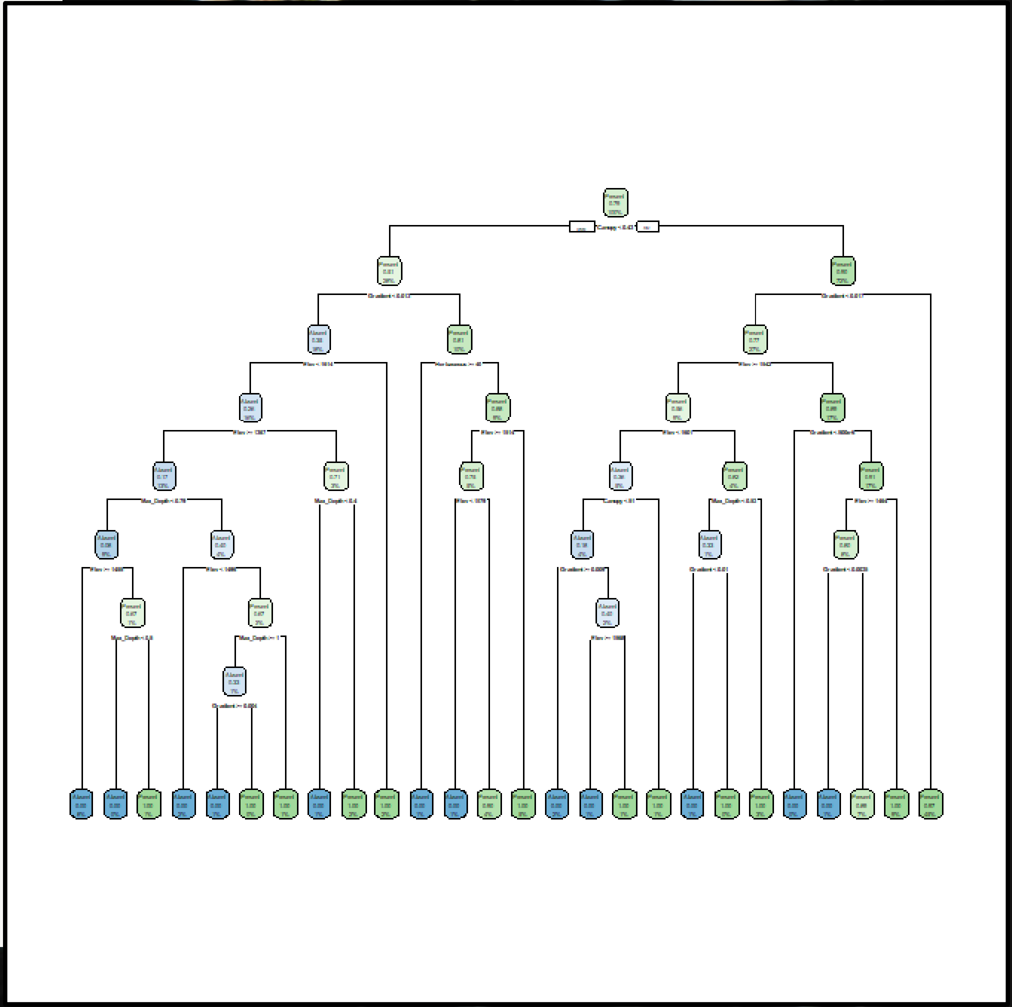




Classification and Regression Trees: Pruning the Tree

CART is *highly* prone to overfitting, which is problematic since this method is commonly used in a predictive capacity!

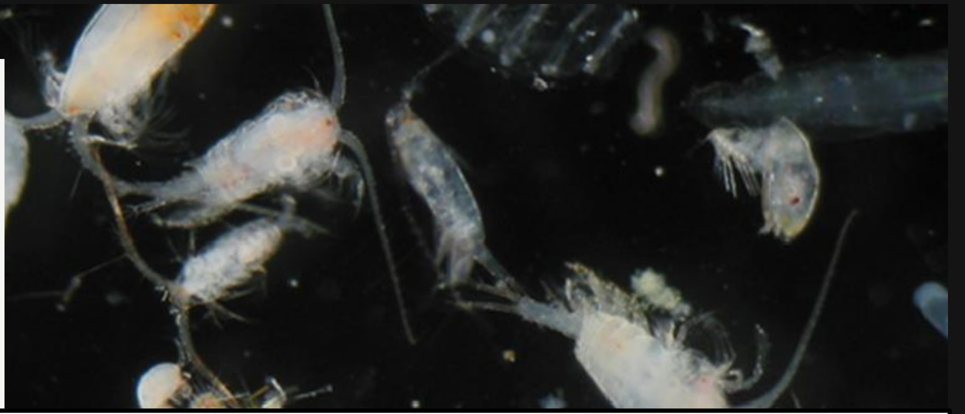


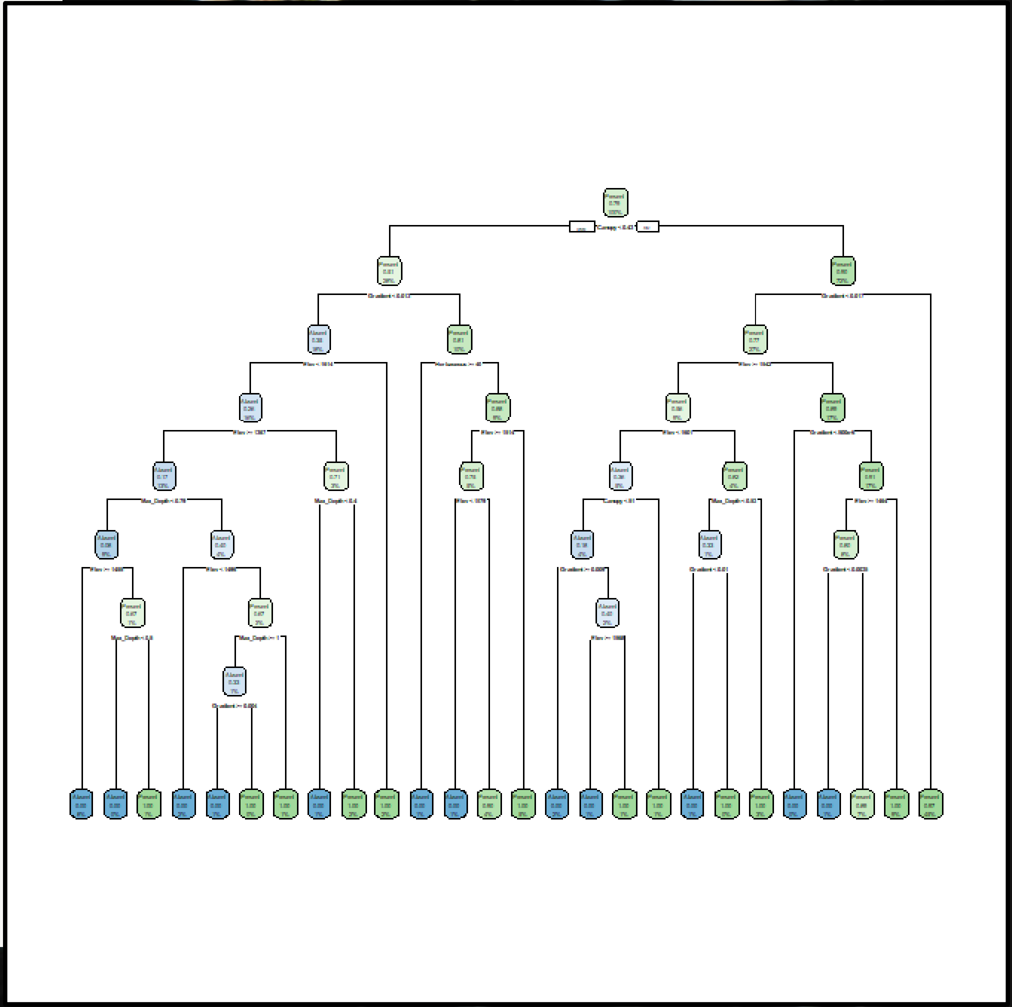


Classification and Regression Trees: Pruning the Tree

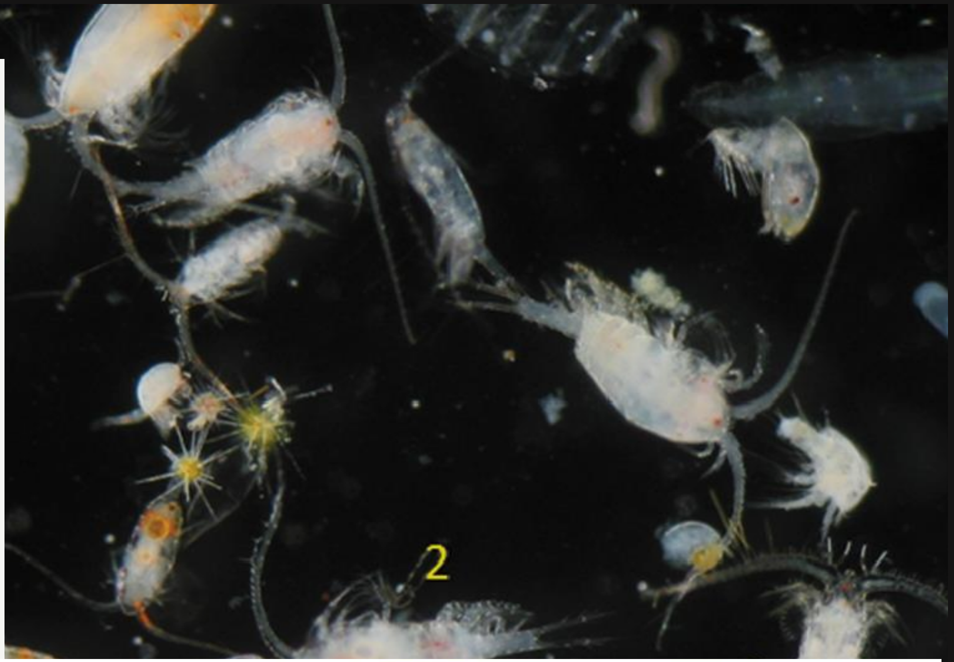
CART is *highly* prone to overfitting, which is problematic since this method is commonly used in a predictive capacity!

Overfit trees are also difficult to interpret.





Classification and Regression Trees: Pruning the Tree



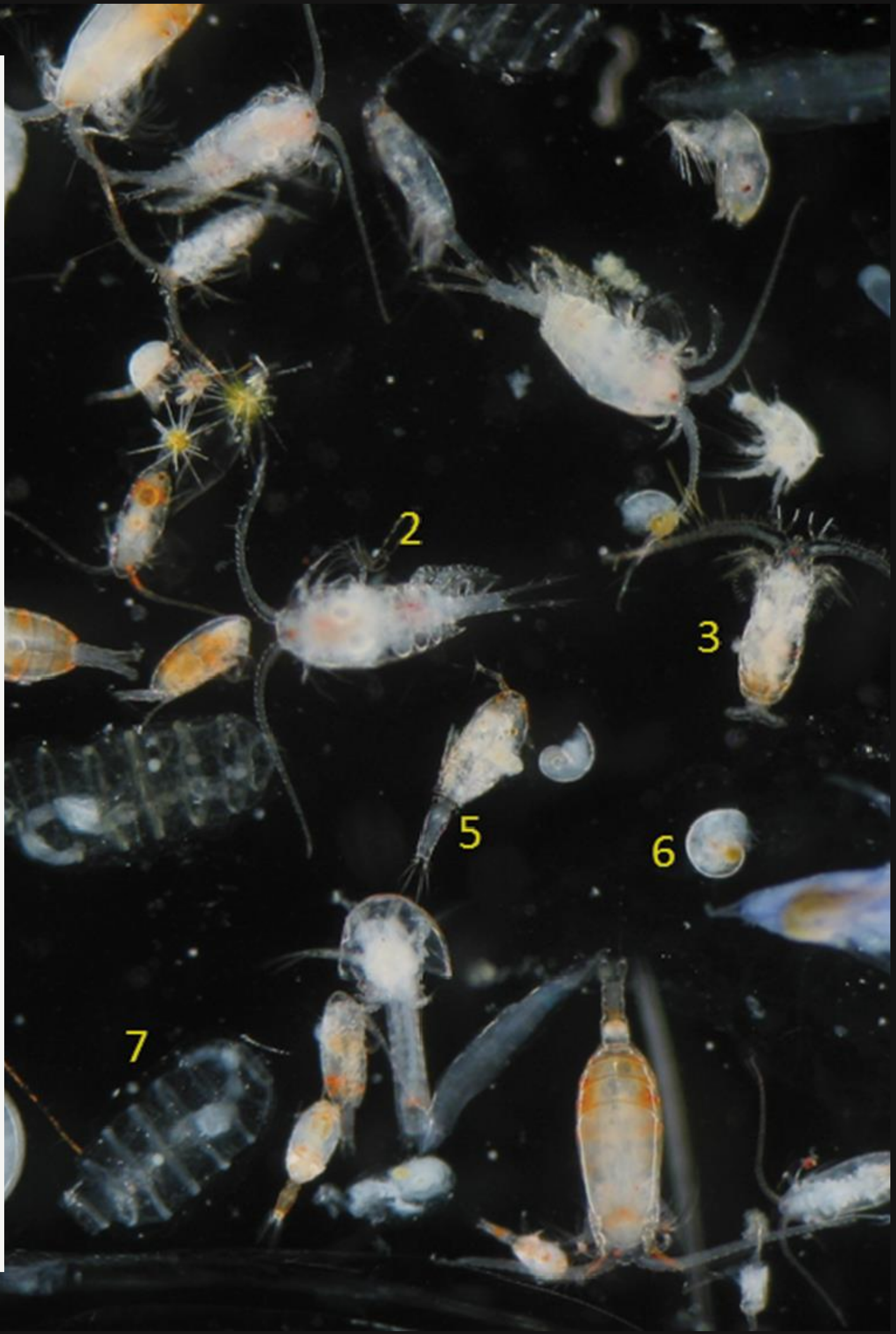
CART is *highly* prone to overfitting, which is problematic since this method is commonly used in a predictive capacity!

Overfit trees are also difficult to interpret.

Enter, PRUNING!

Classification and Regression Trees: Pruning the Tree

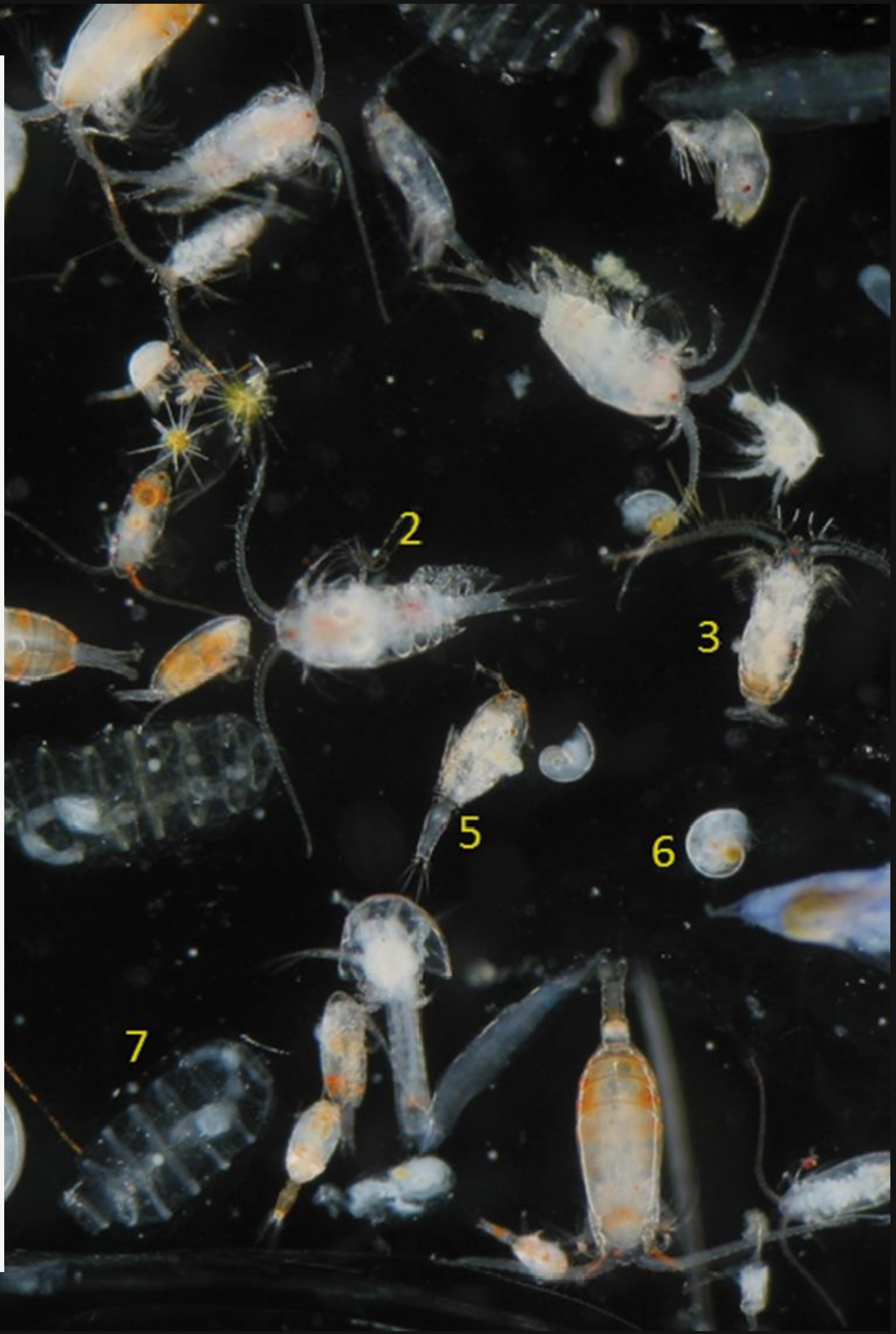
Pruning balances tree size with model accuracy.



Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

Cross-validation is used to determine optimal tree depth and prevent overfitting by **maximizing accuracy** while balancing bias and variance.

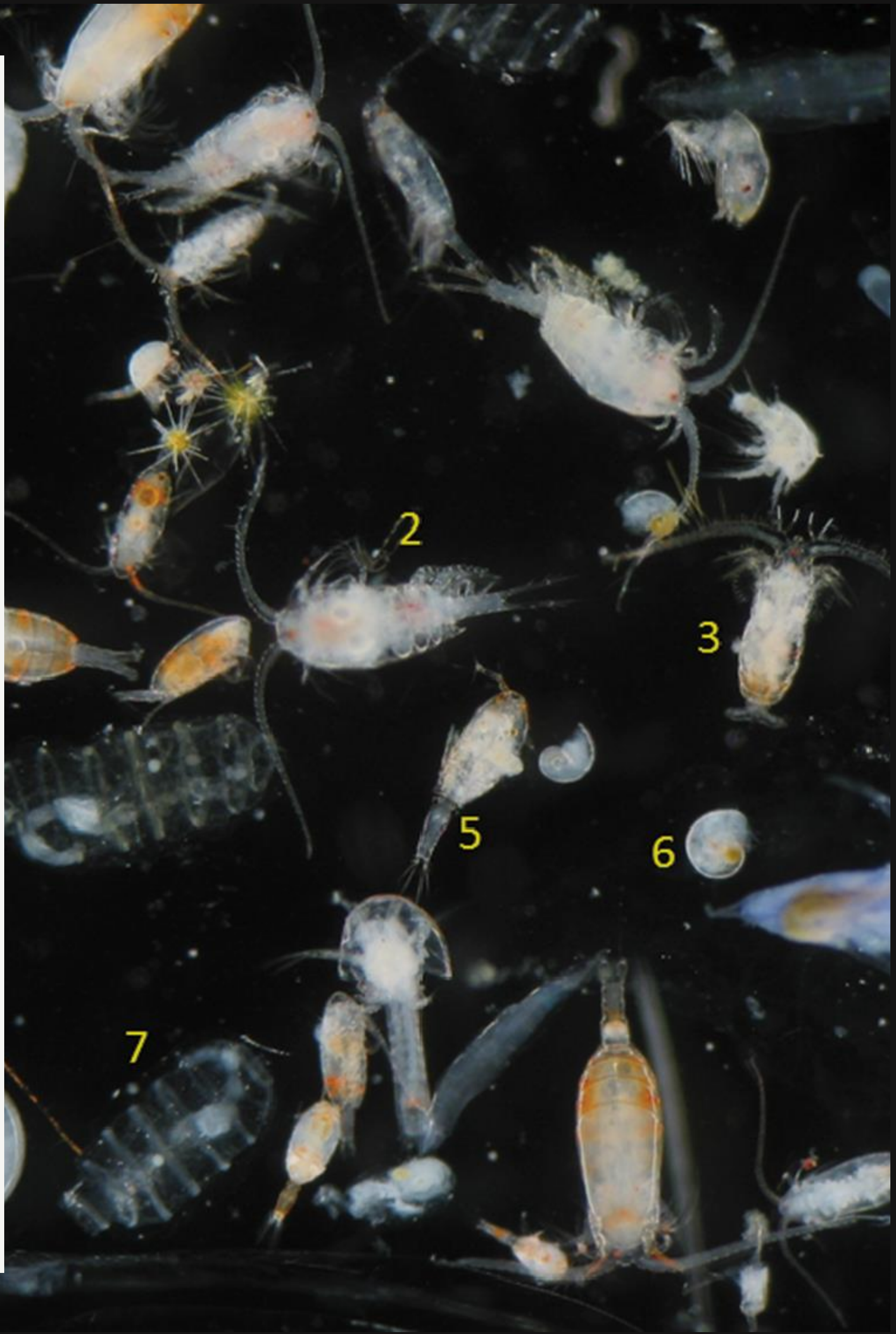


Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

Cross-validation is used to determine optimal tree depth and prevent overfitting by **maximizing accuracy** while balancing bias and variance.

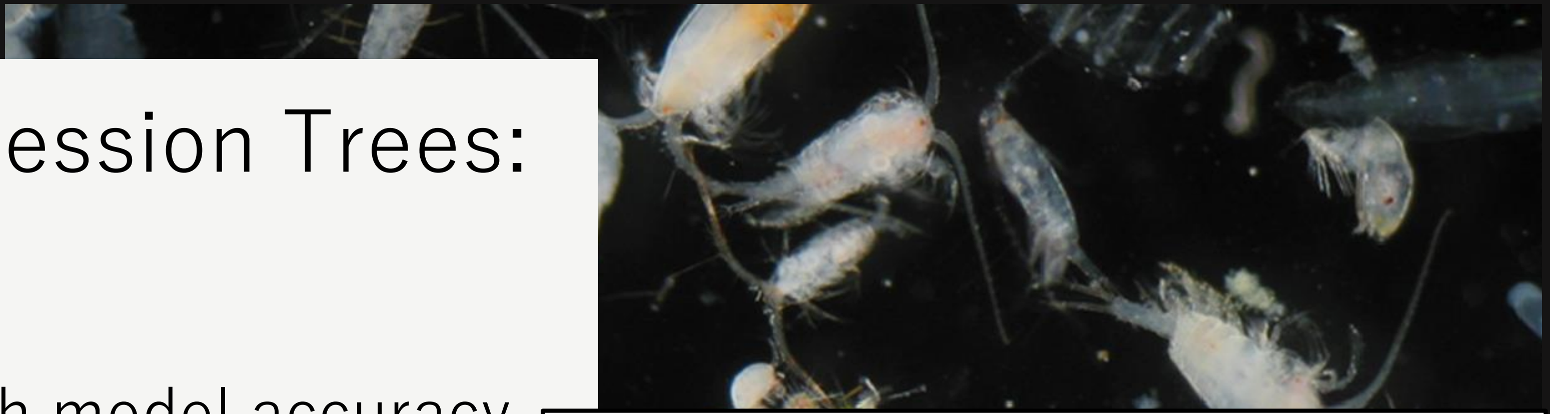
K-fold cross-validation evaluates how well a model generalizes to unseen data. The data are split into k parts; the model is trained on $k-1$ parts and tested on the remaining one.

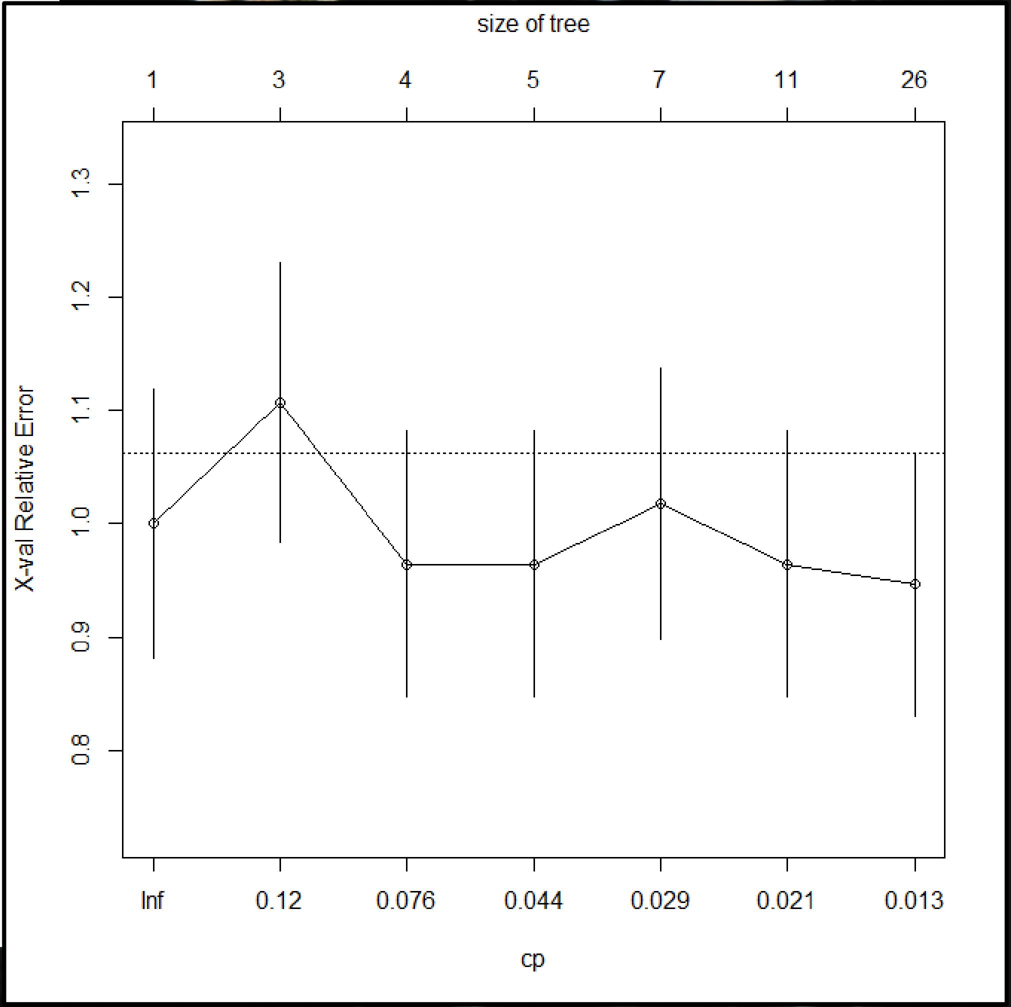


Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

The **complexity parameter (cp)** provides information about how we can best prune the tree. A smaller cp allows the tree to grow more complex, while a larger cp results in a simpler tree that may generalize better to unseen data.



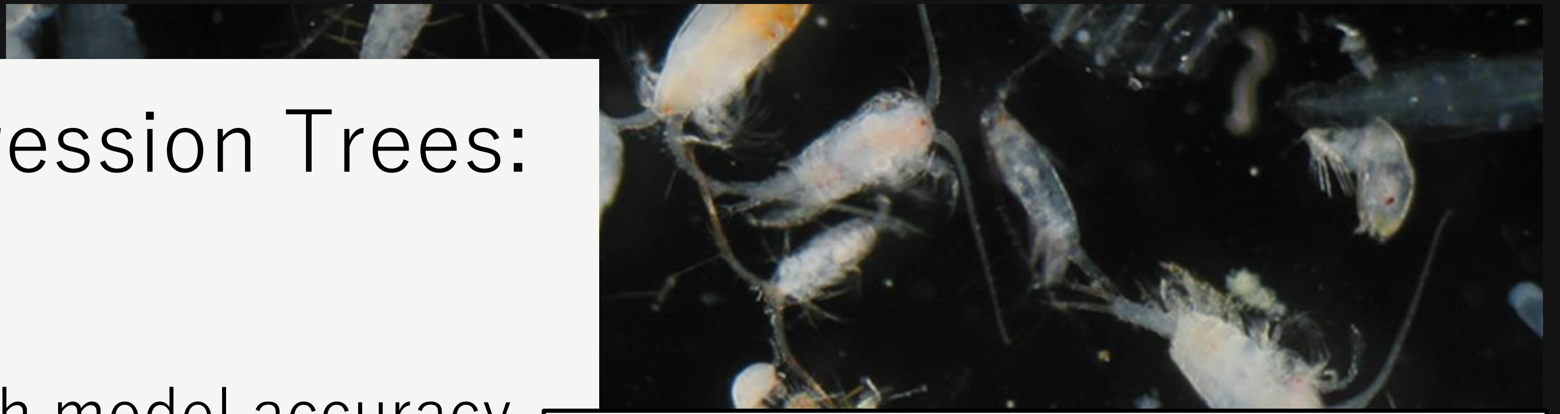


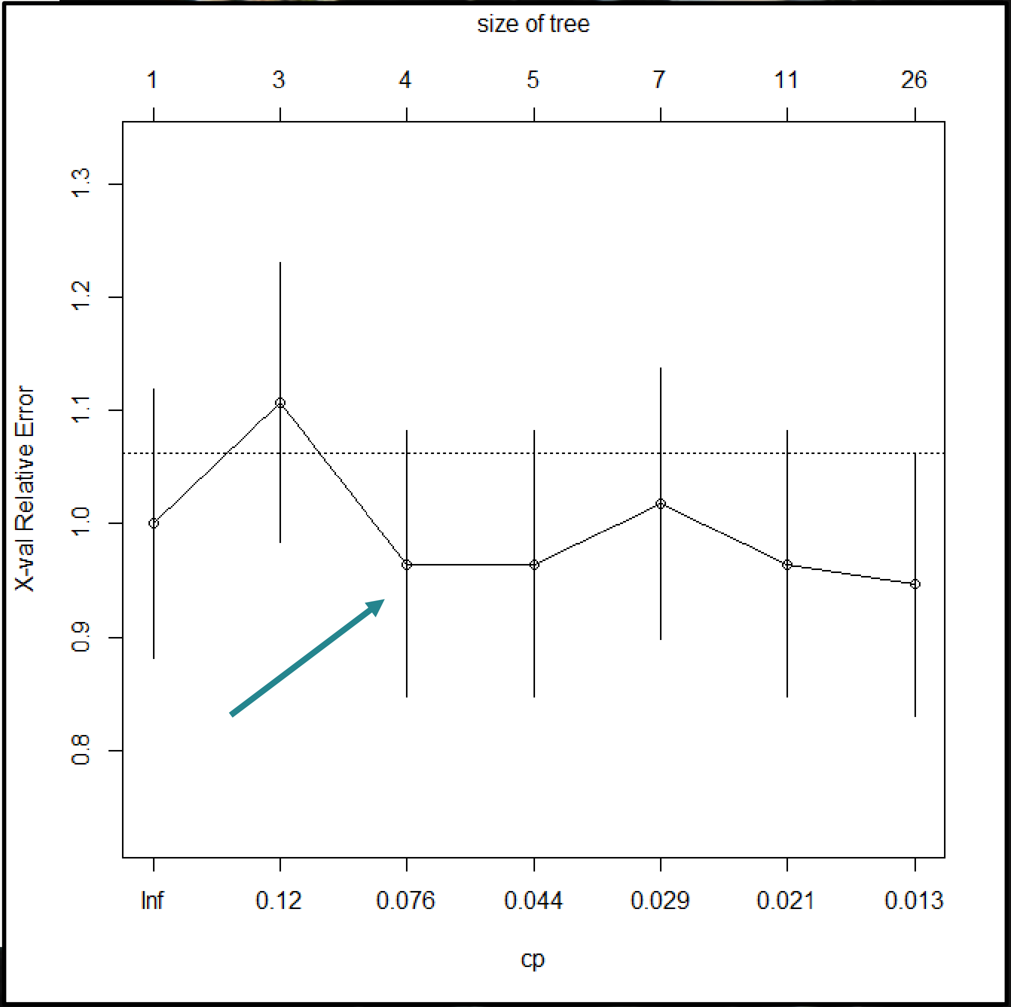
Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

The **complexity parameter (cp)** provides information about how we can best prune the tree. A smaller cp allows the tree to grow more complex, while a larger cp results in a simpler tree that may generalize better to unseen data.

A good choice of cp for pruning is often the leftmost value for which the mean lies significantly below the horizontal line representing the 1 SE of the minimum.



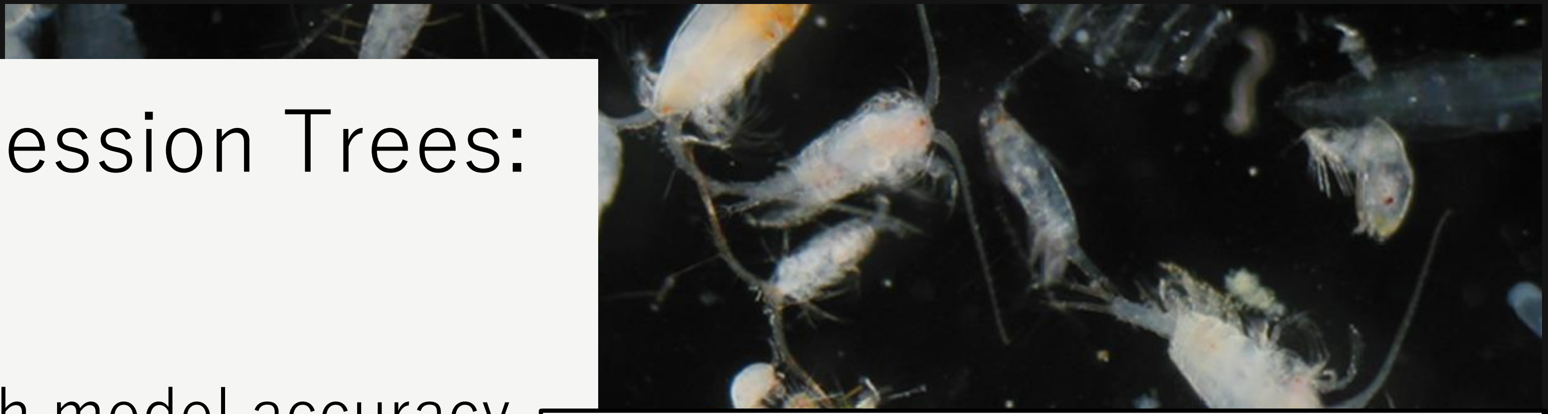


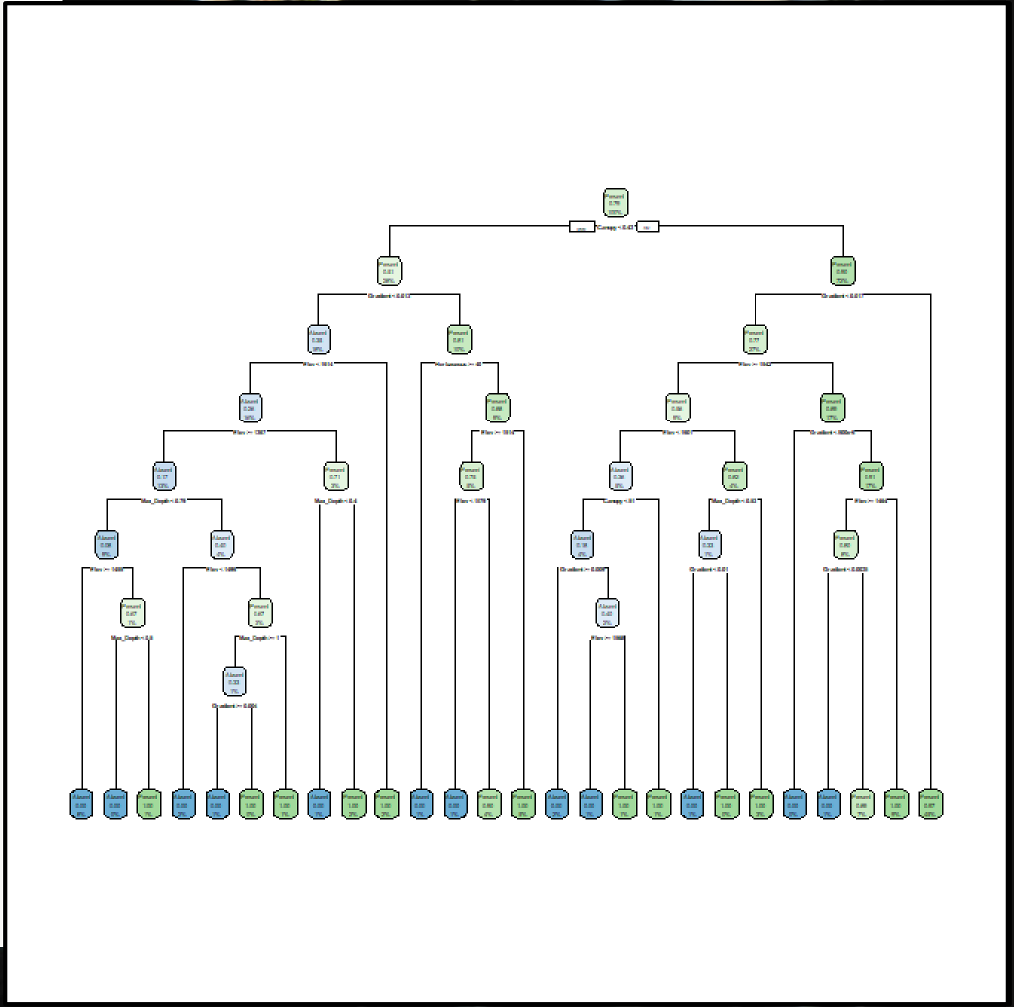
Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

The **complexity parameter (cp)** provides information about how we can best prune the tree. A smaller cp allows the tree to grow more complex, while a larger cp results in a simpler tree that may generalize better to unseen data.

A good choice of cp for pruning is often the leftmost value for which the mean lies significantly below the horizontal line representing the 1 SE of the minimum.



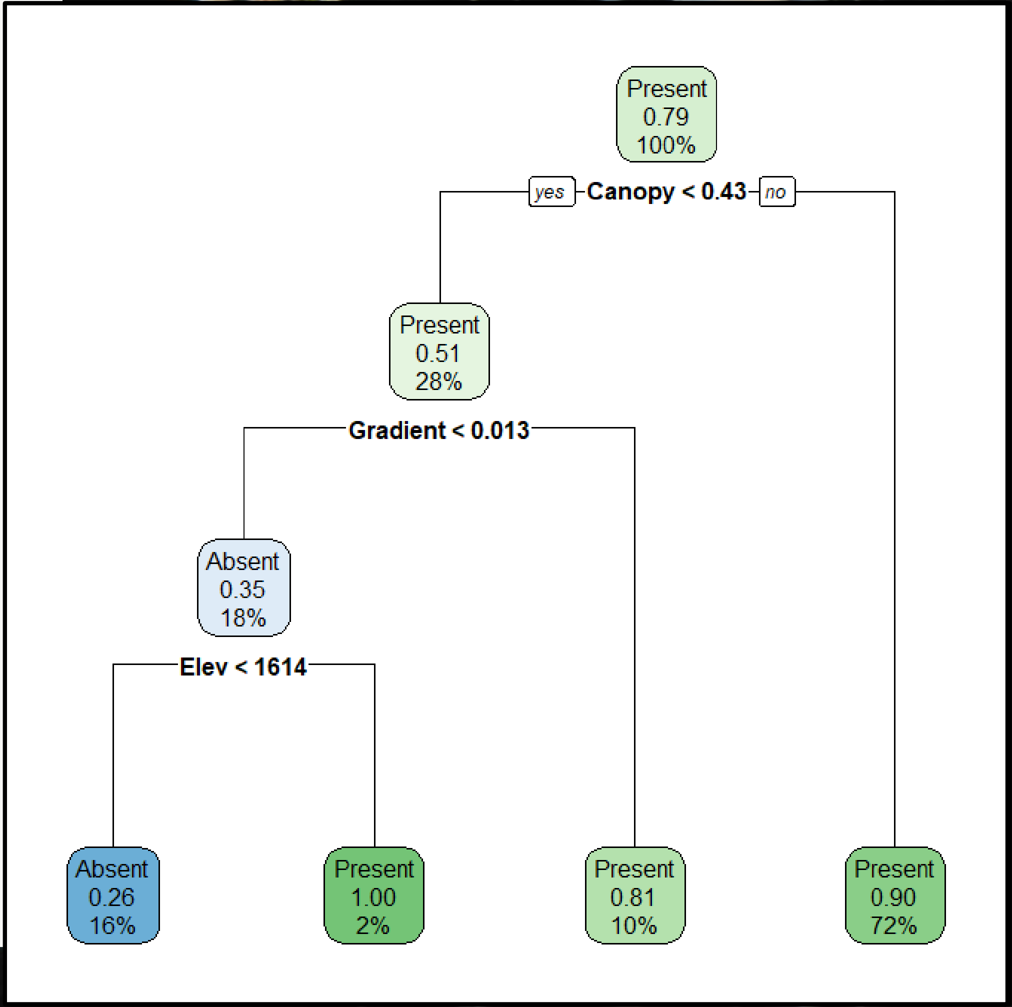


Classification and Regression Trees: Pruning the Tree

Pruning balances tree size with model accuracy.

The **complexity parameter (cp)** provides information about how we can best prune the tree. A smaller cp allows the tree to grow more complex, while a larger cp results in a simpler tree that may generalize better to unseen data.

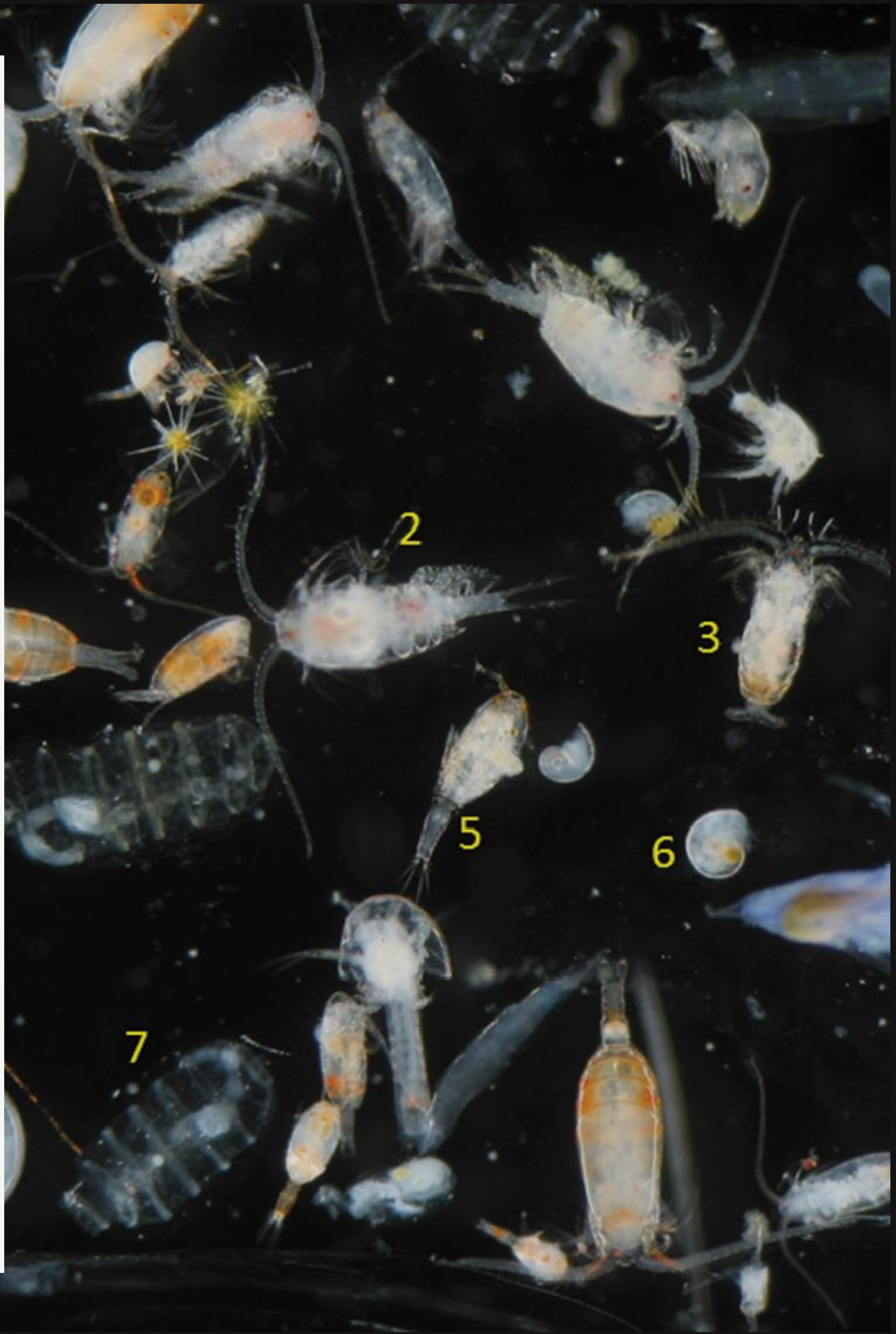
A good choice of cp for pruning is often the leftmost value for which the mean lies significantly below the horizontal line representing the 1 SE of the minimum.



Classification and Regression Trees: Strengths and Weaknesses

Advantages:

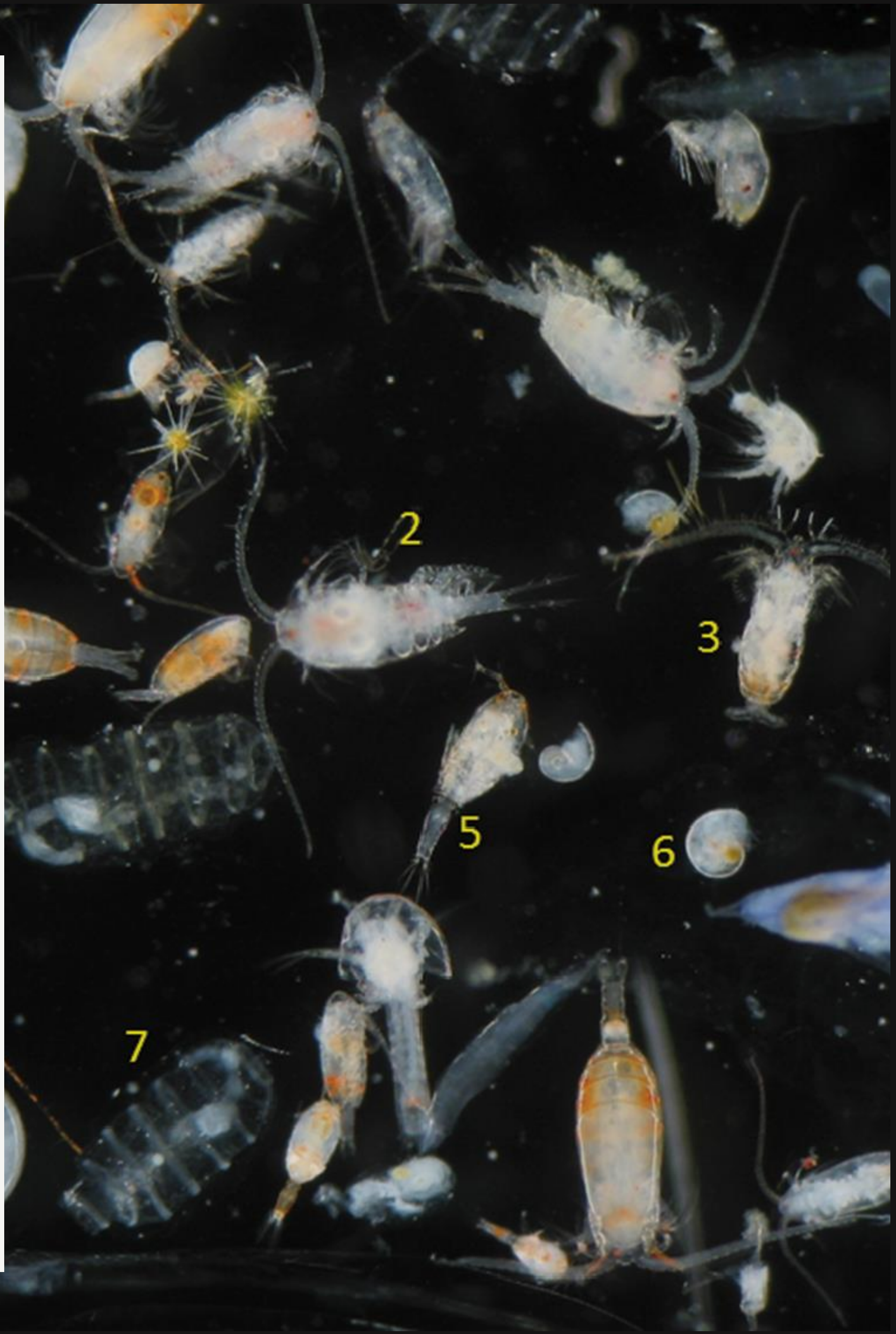
- Easy to visualize and understand
- No assumption of data distribution
- Provides insight into predictor importance
- Can handle non-linearity



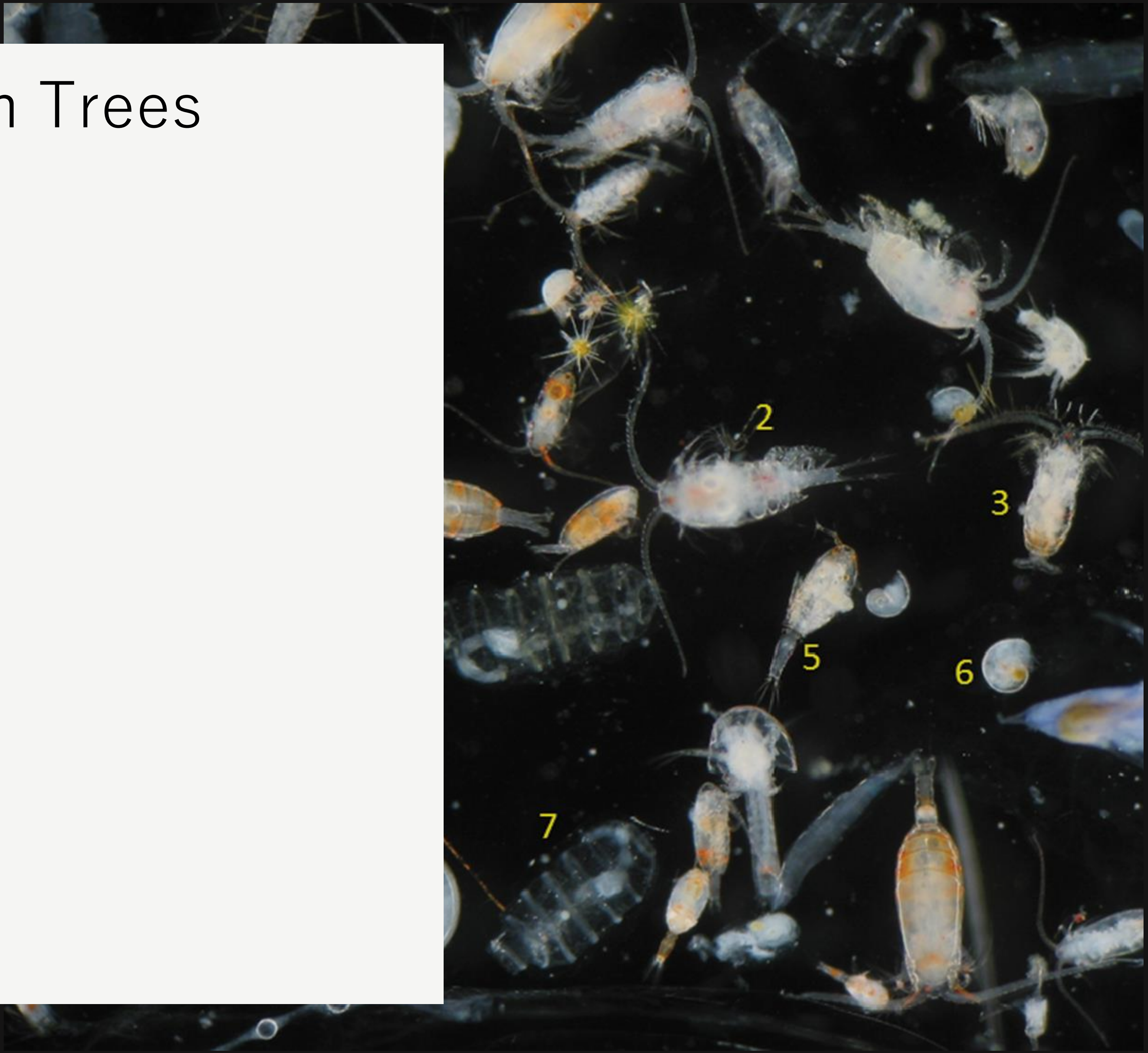
Classification and Regression Trees: Strengths and Weaknesses

Advantages:
- Easy to visualize and understand
- No assumption of data distribution
- Provides insight into predictor importance
- Can handle non-linearity

Limitations:
- Overfitting
- Sensitivity (to noise, data changes)
- Bias toward categorical predictors

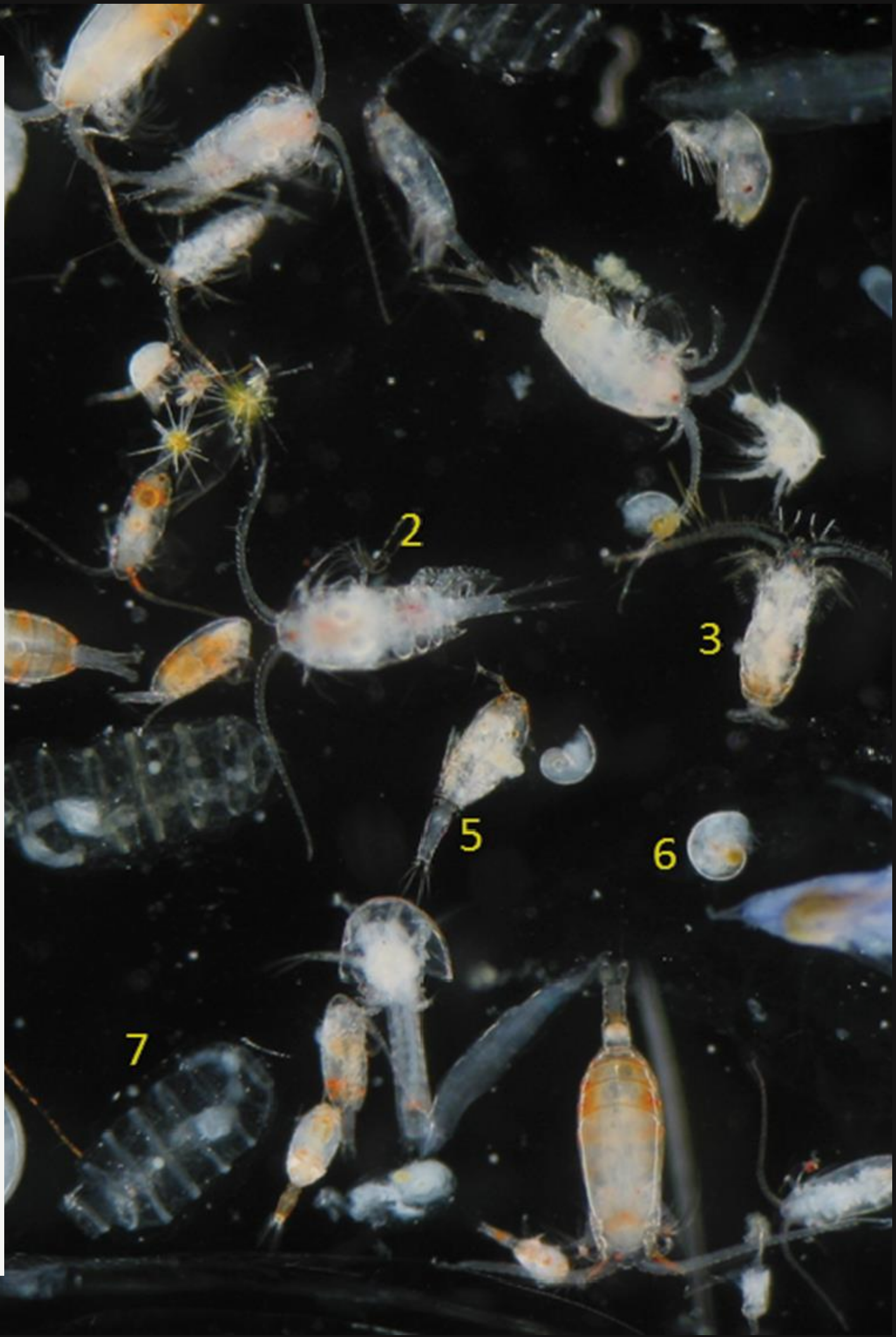


Multivariate Regression Trees



Multivariate Regression Trees

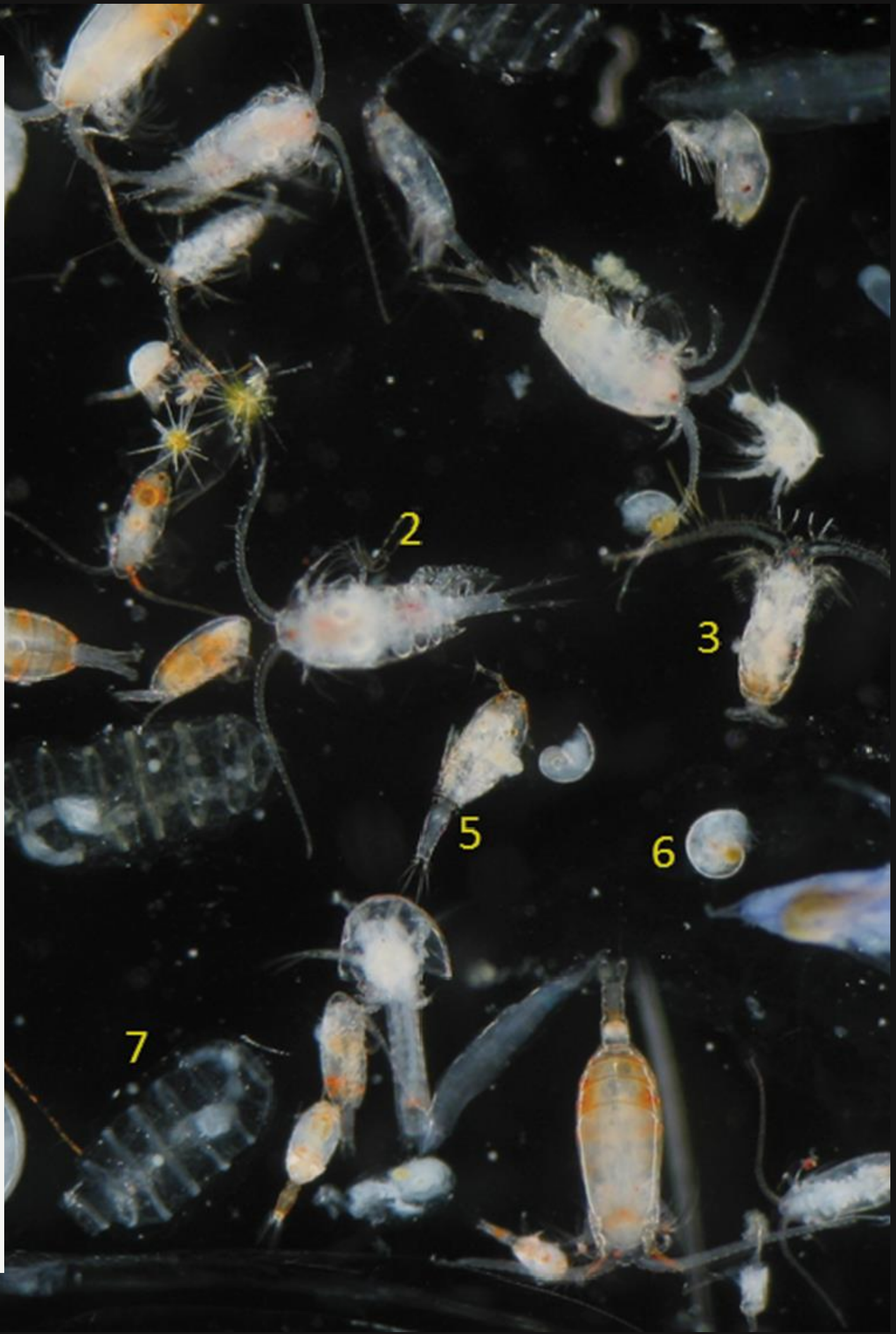
Multivariate Regression Trees (MRT) extend traditional regression trees to handle multiple dependent variables.



Multivariate Regression Trees

Multivariate Regression Trees (MRT) extend traditional regression trees to handle multiple dependent variables.

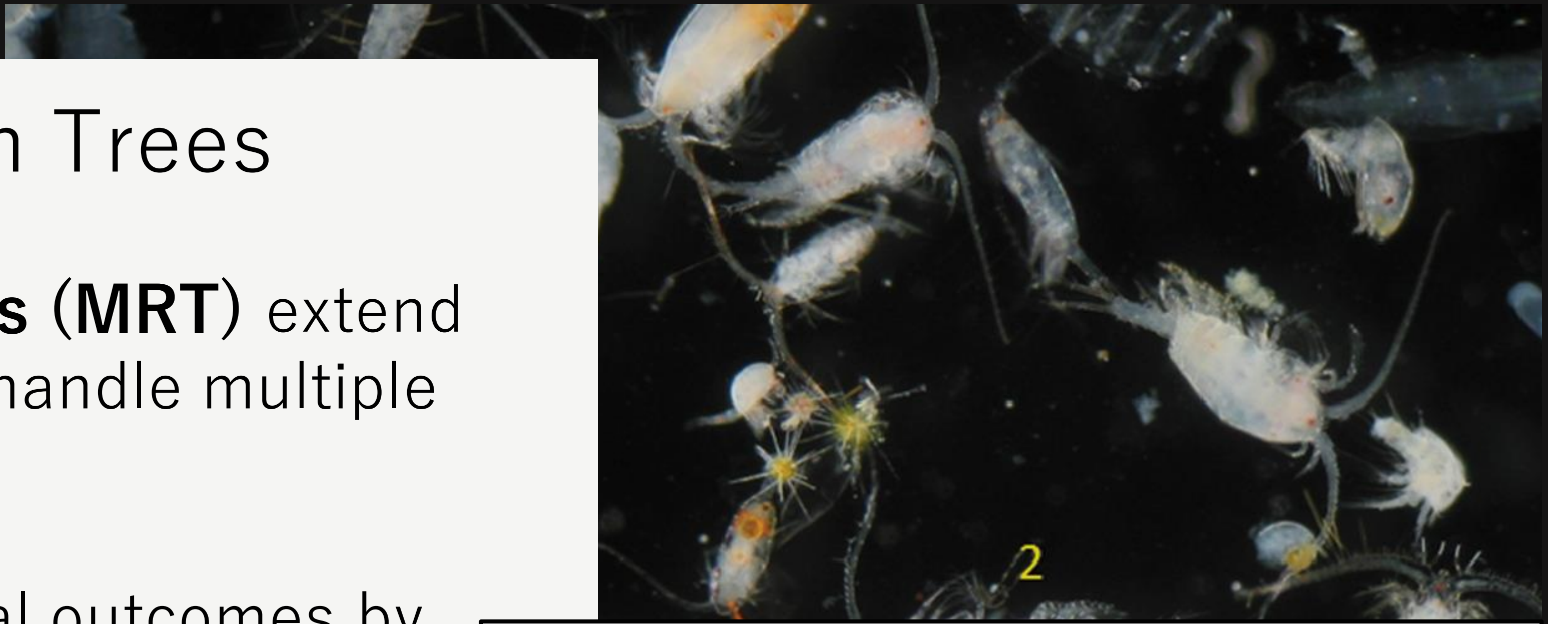
Simultaneously predicts several outcomes by recursively partitioning the predictor space.

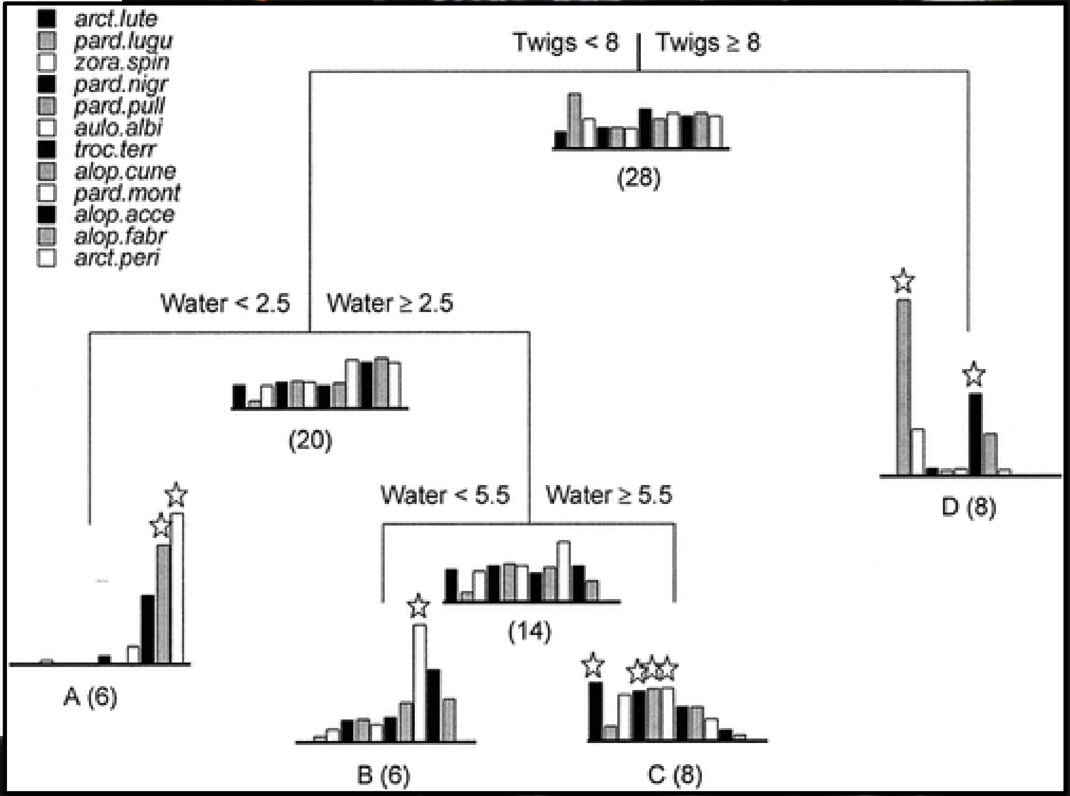


Multivariate Regression Trees

Multivariate Regression Trees (MRT) extend traditional regression trees to handle multiple dependent variables.

Simultaneously predicts several outcomes by recursively partitioning the predictor space.





De'ath 2002

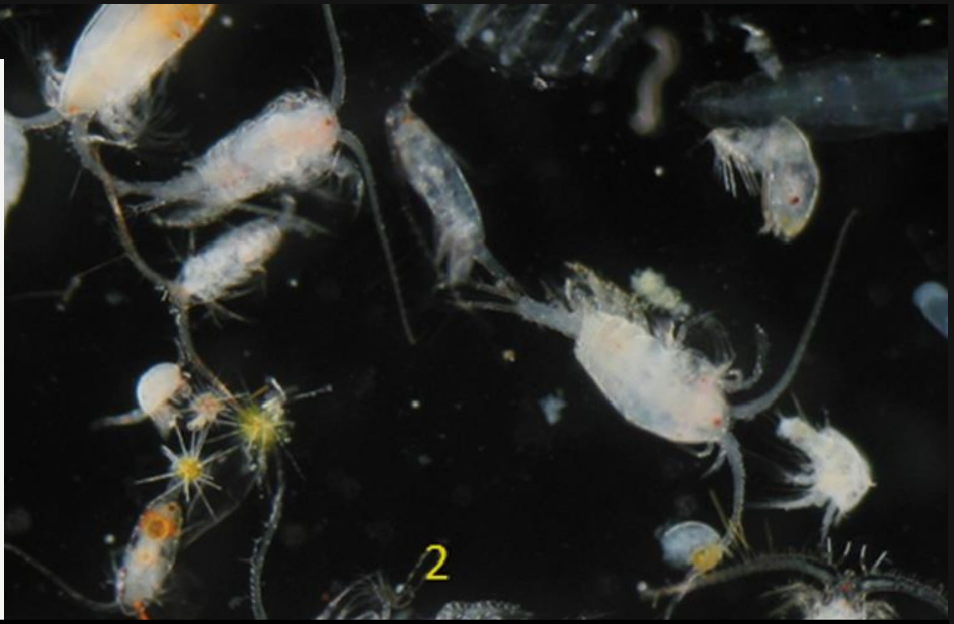
Multivariate Regression Trees

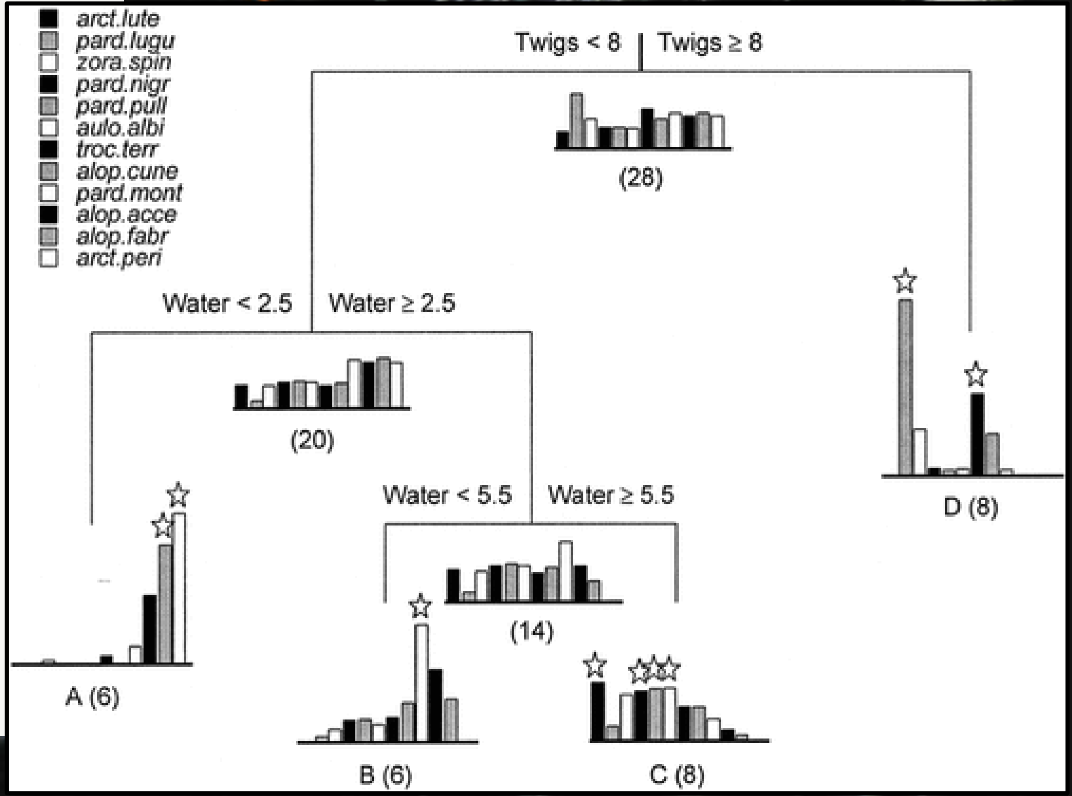
Multivariate Regression Trees (MRT) extend traditional regression trees to handle multiple dependent variables.

Simultaneously predicts several outcomes by recursively partitioning the predictor space.

Goal is to **minimize multivariate variance**.

De'ath 2002

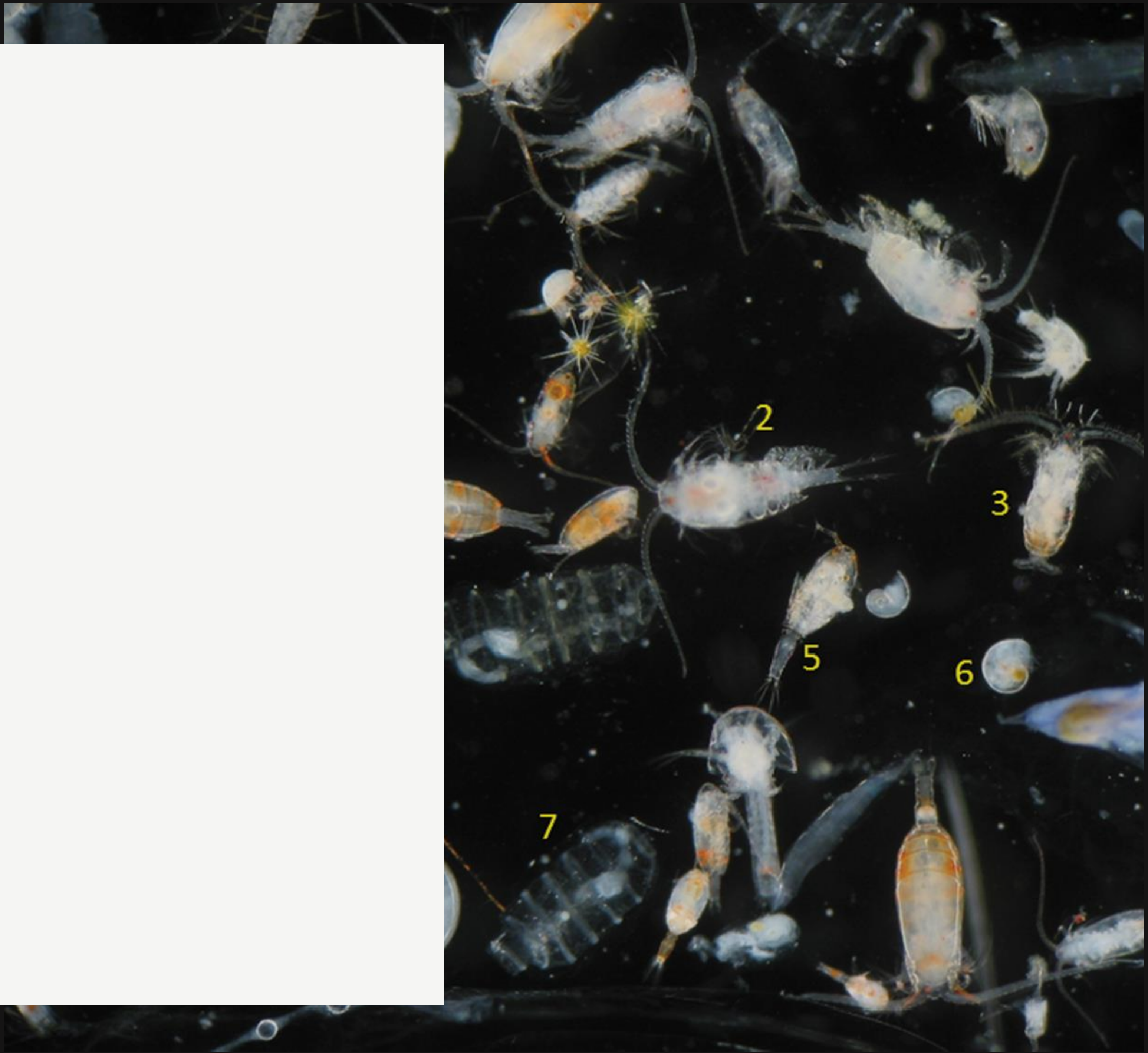




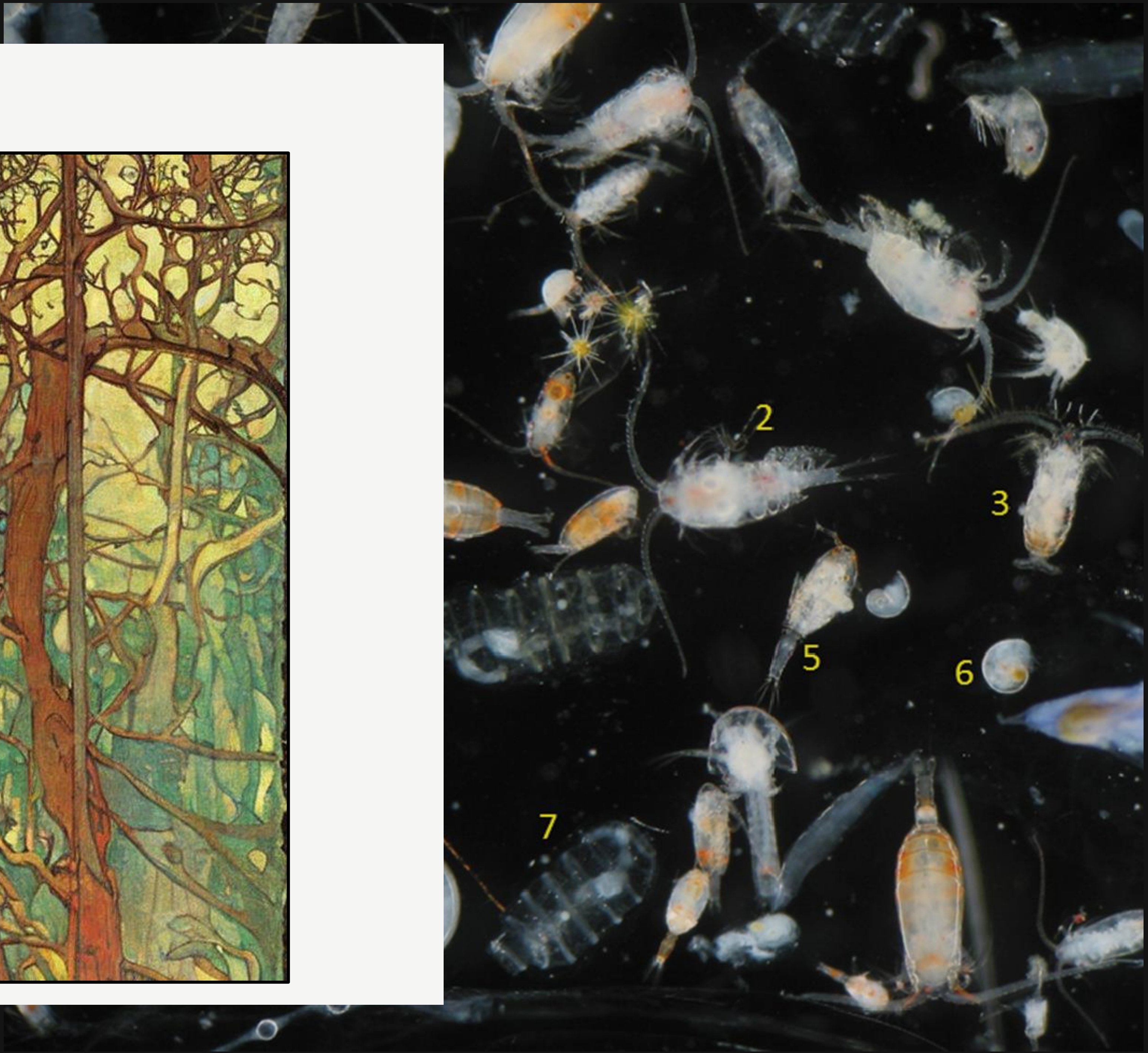
Multivariate Regression Trees

The `mvpart` package in R is dead.

Random Forests

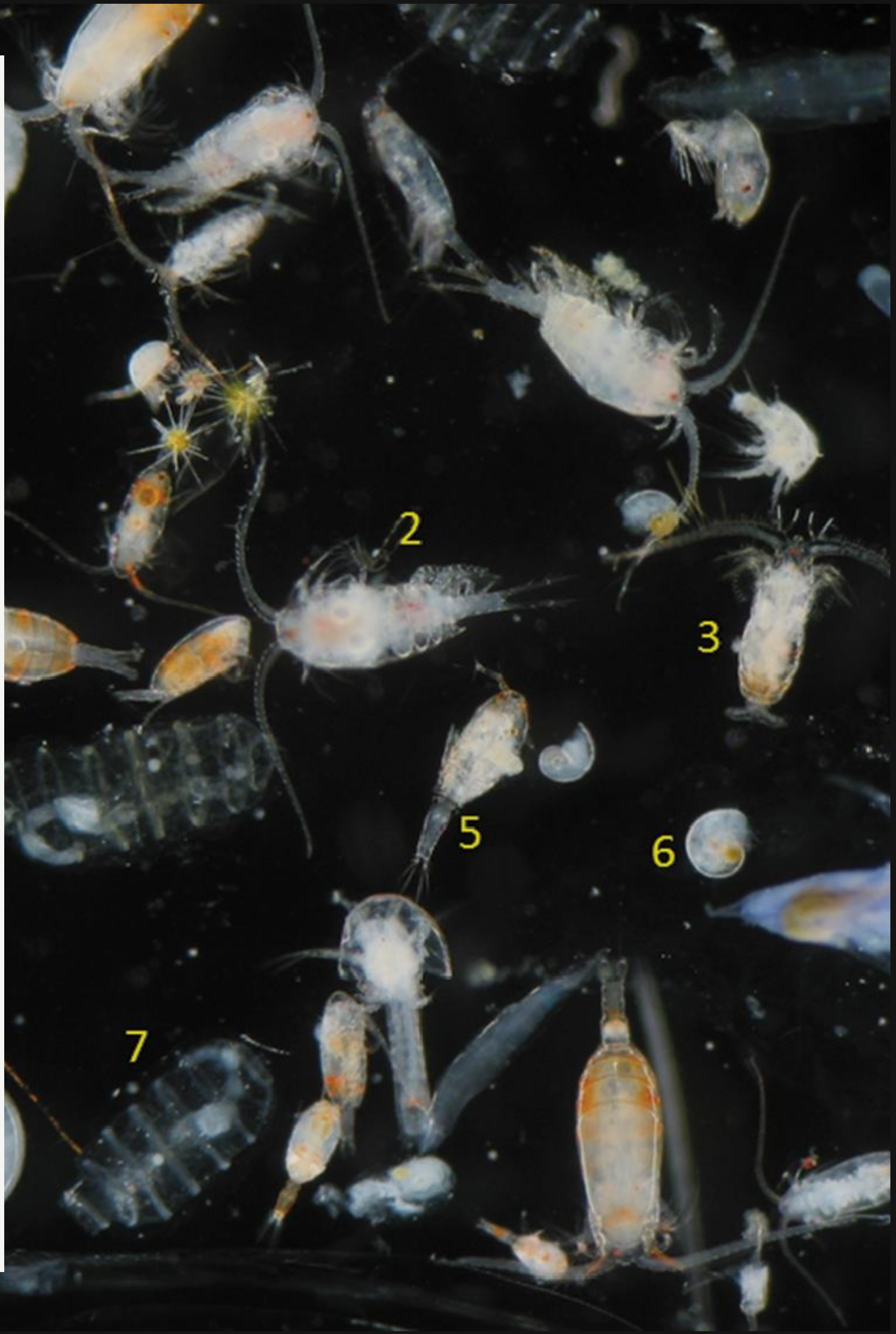


Random Forests



Random Forests

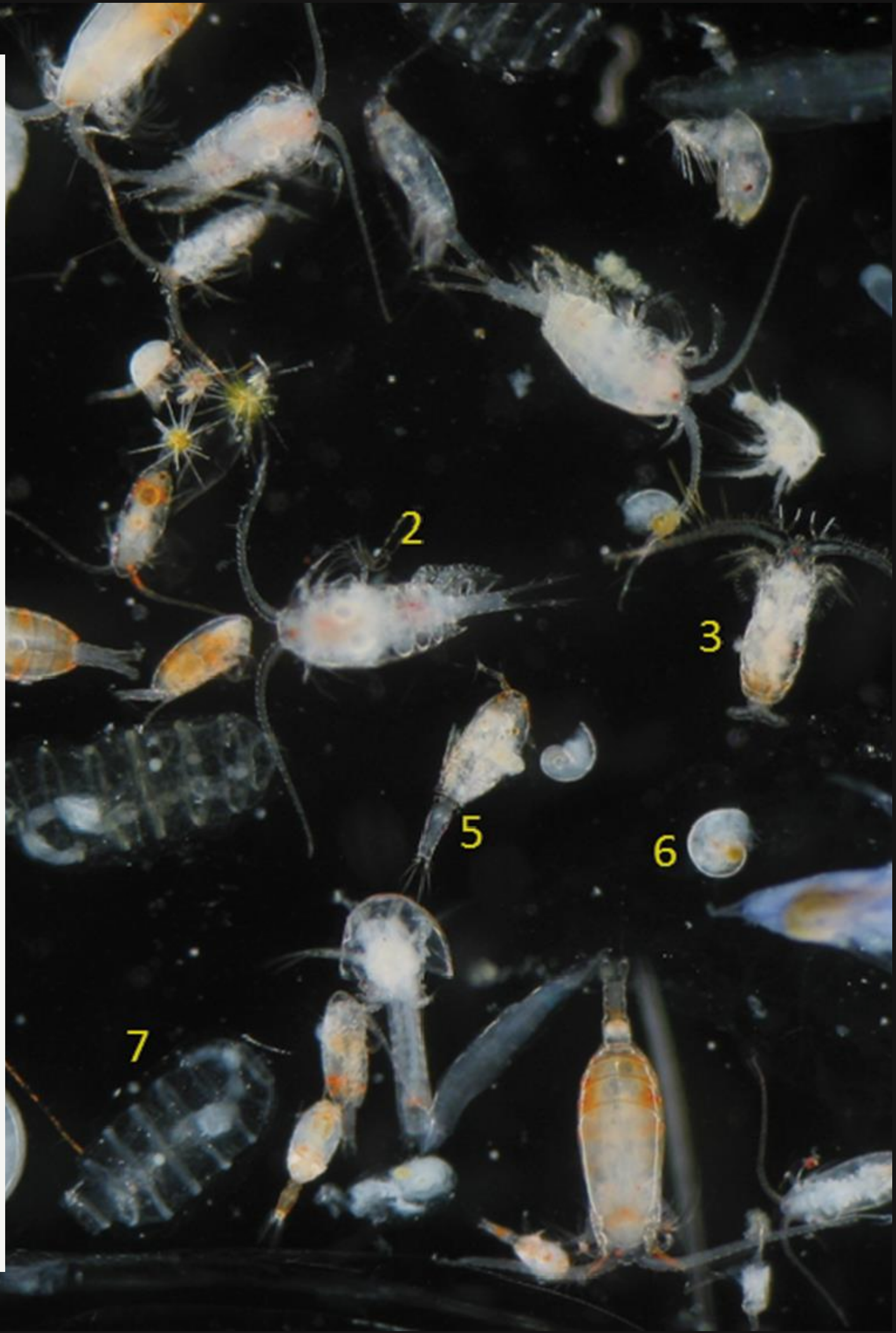
Random Forests is a machine learning method that combines multiple decision trees to improve prediction accuracy and robustness.



Random Forests

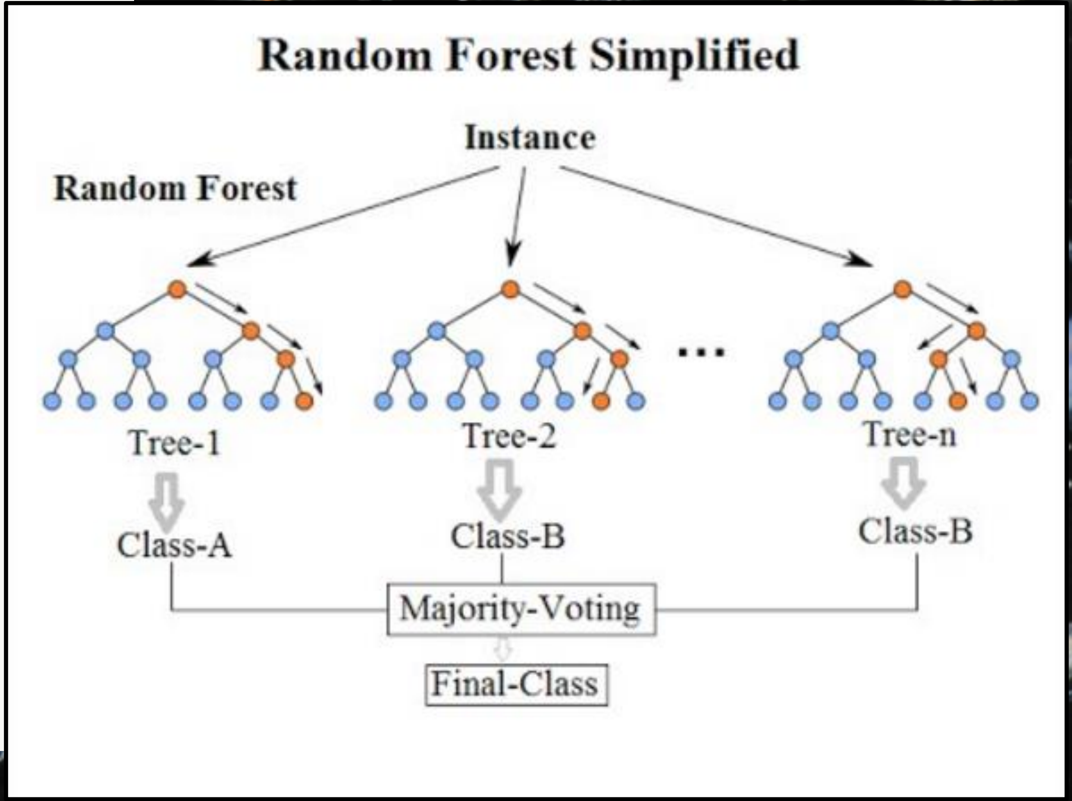
Random Forests is a machine learning method that combines multiple decision trees to improve prediction accuracy and robustness.

Each tree in the forest is trained on a different random subset of the data and features.



Random Forests

Random Forests uses the concept of **bagging** (**Bootstrap Aggregation**) to create multiple training datasets by randomly sampling with replacement.

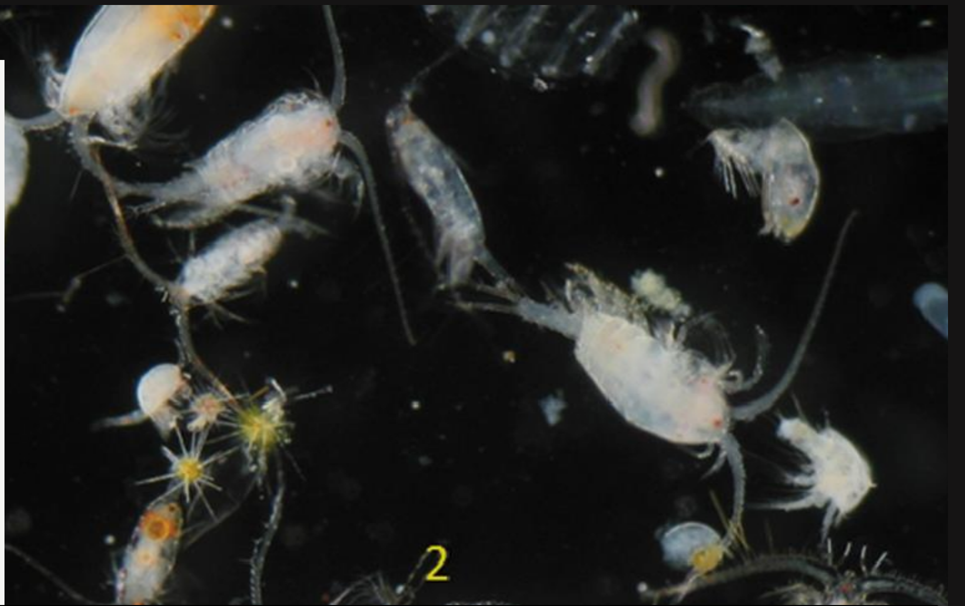


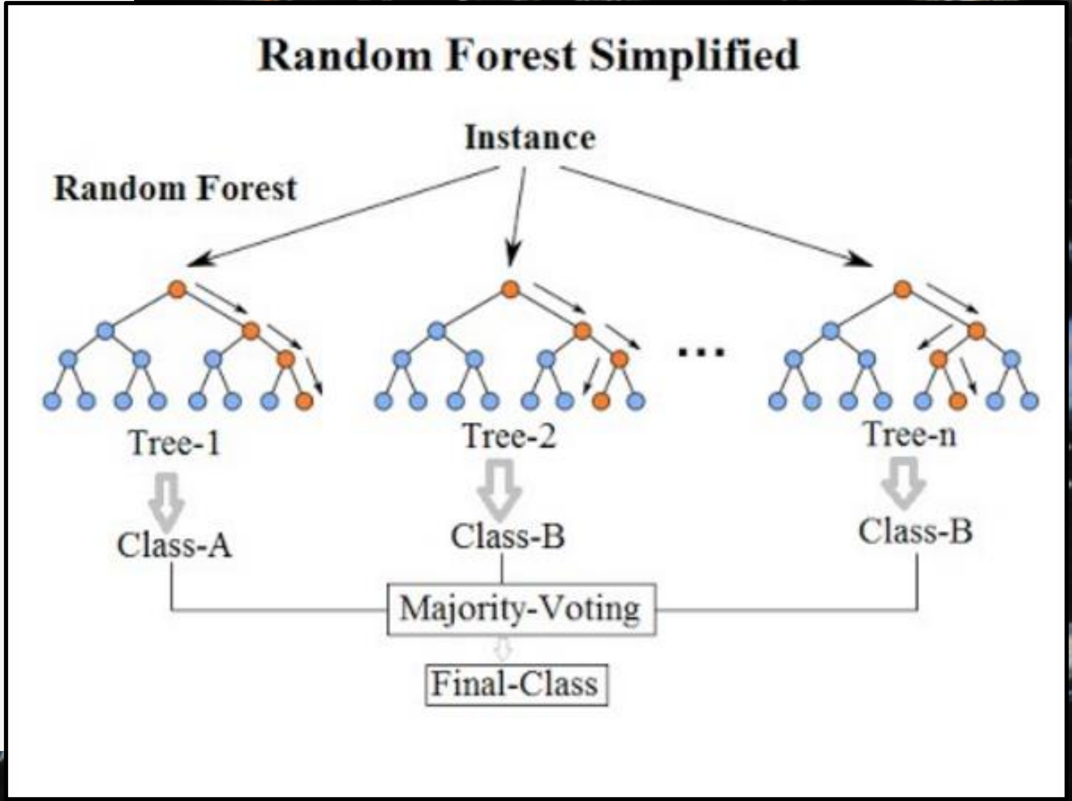
Random Forests

Random Forests uses the concept of **bagging** (**Bootstrap Aggregation**) to create multiple training datasets by randomly sampling with replacement.

Each tree is trained independently, and the final prediction is made by aggregating:

- For classification: Majority voting from all trees.
- For regression: Averaging the output from all trees.



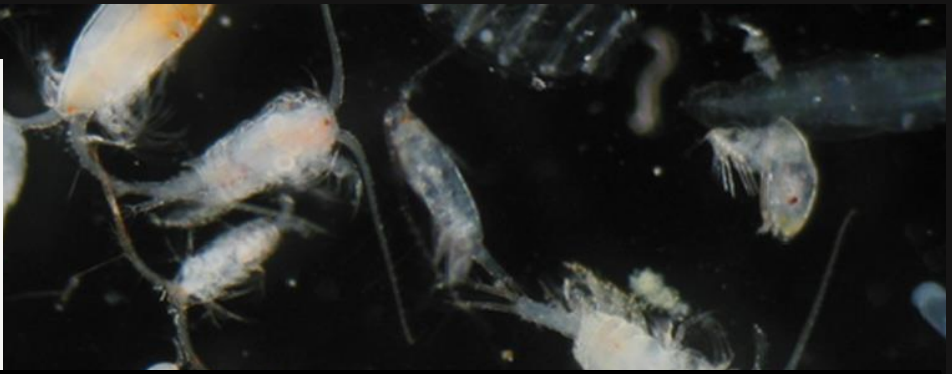


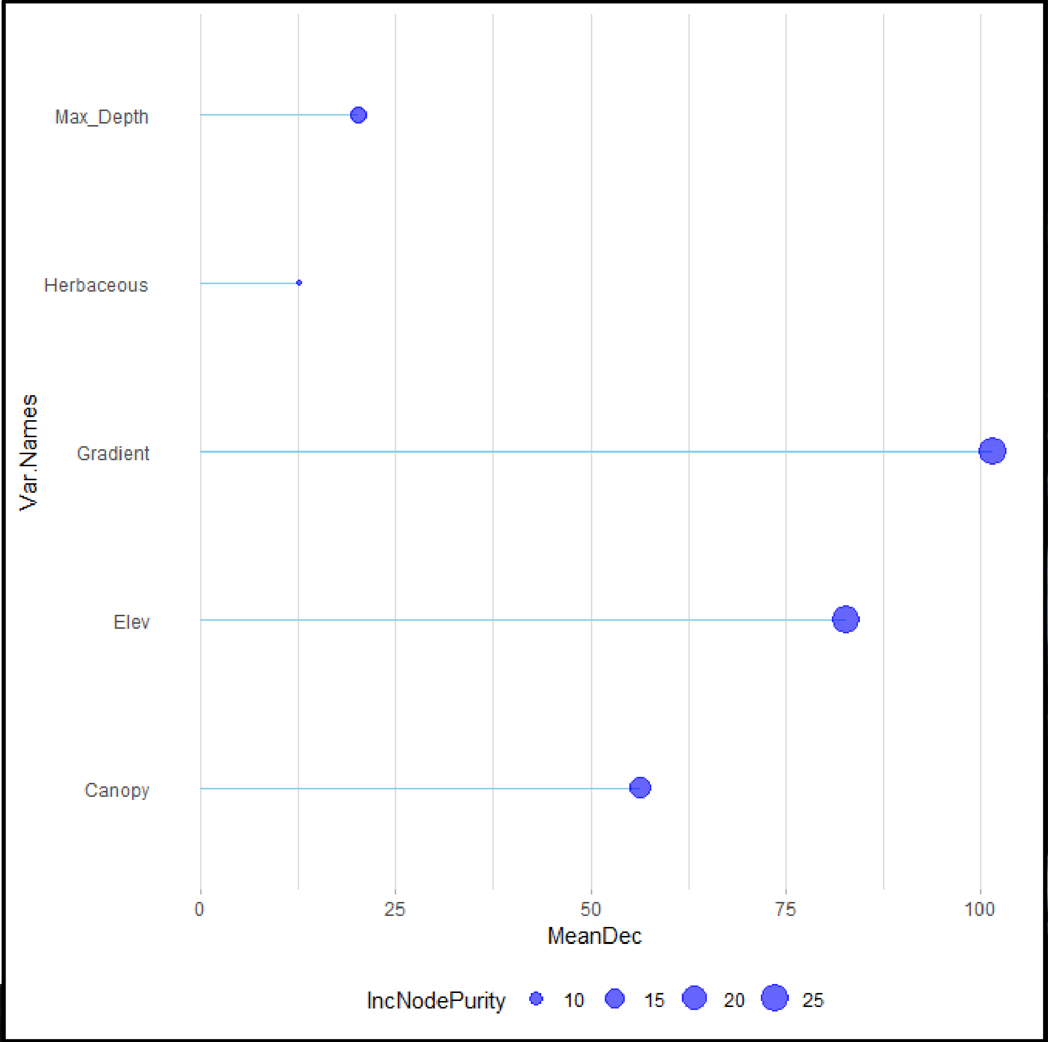
Random Forests

Random Forests uses the concept of **bagging** (**Bootstrap Aggregation**) to create multiple training datasets by randomly sampling with replacement.

Each tree is trained independently, and the final prediction is made by aggregating:

- For classification: Majority voting from all trees.
- For regression: Averaging the output from all trees.



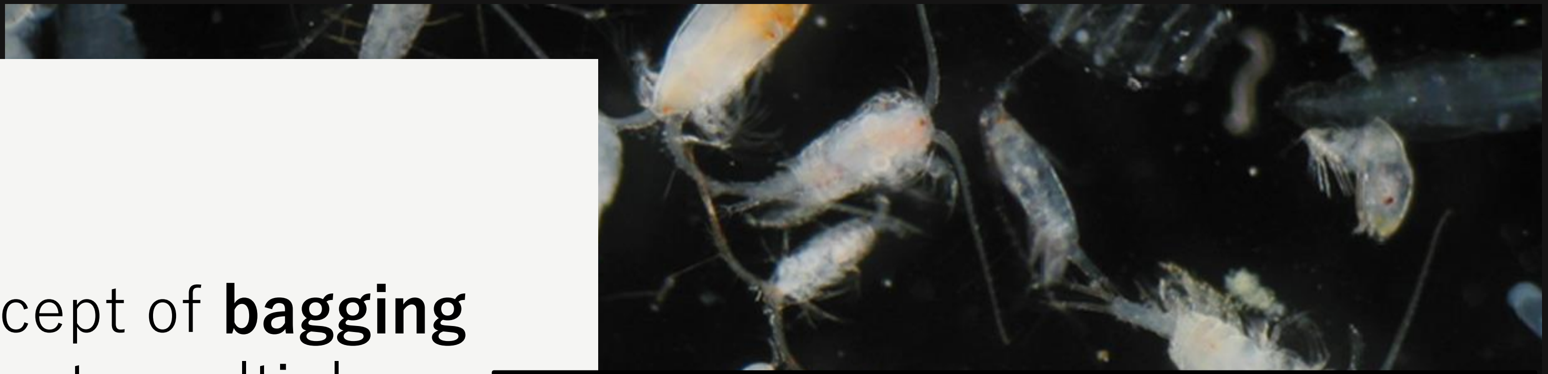


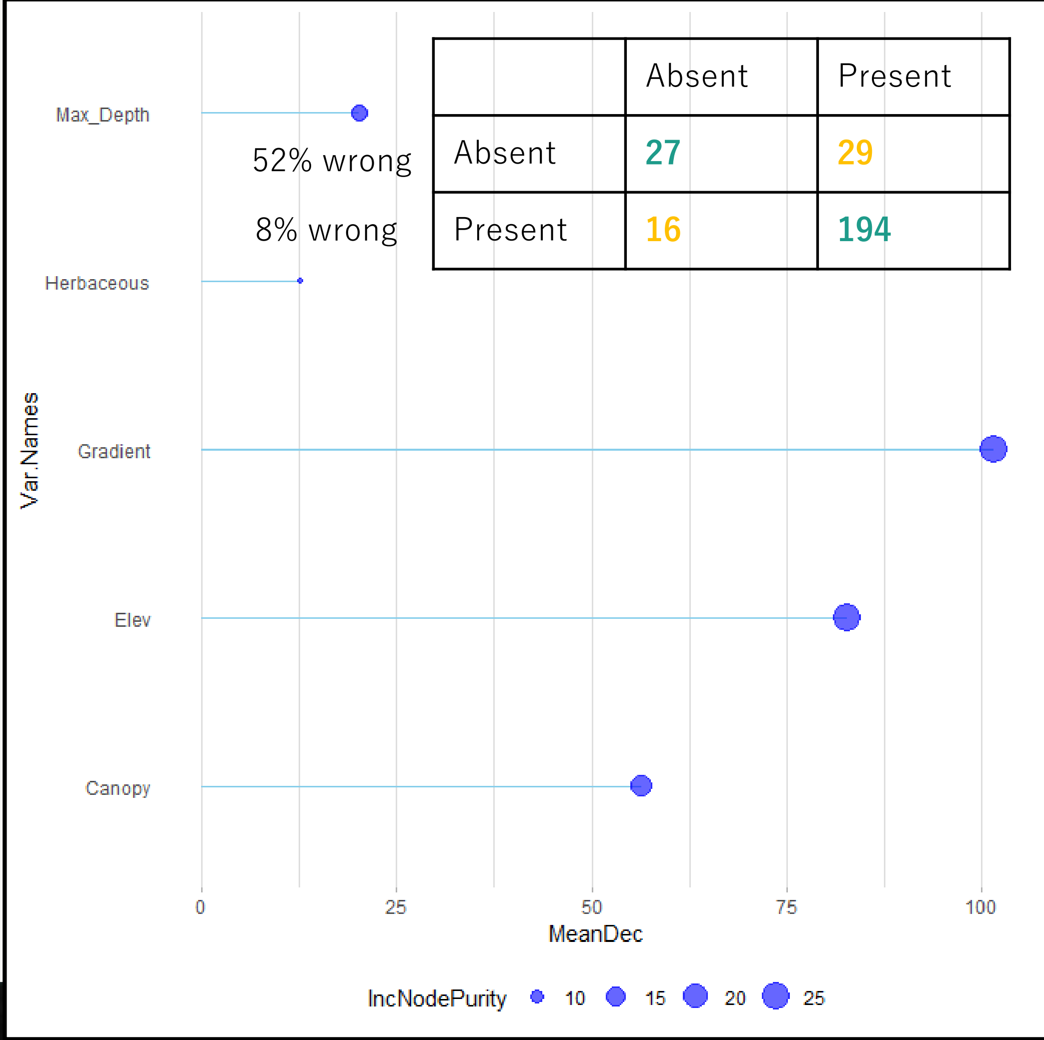
Random Forests

Random Forests uses the concept of **bagging** (**Bootstrap Aggregation**) to create multiple training datasets by randomly sampling with replacement.

Each tree is trained independently, and the final prediction is made by aggregating:

- For classification: Majority voting from all trees.
- For regression: Averaging the output from all trees.





	Absent	Present
Absent	27	29
Present	16	194

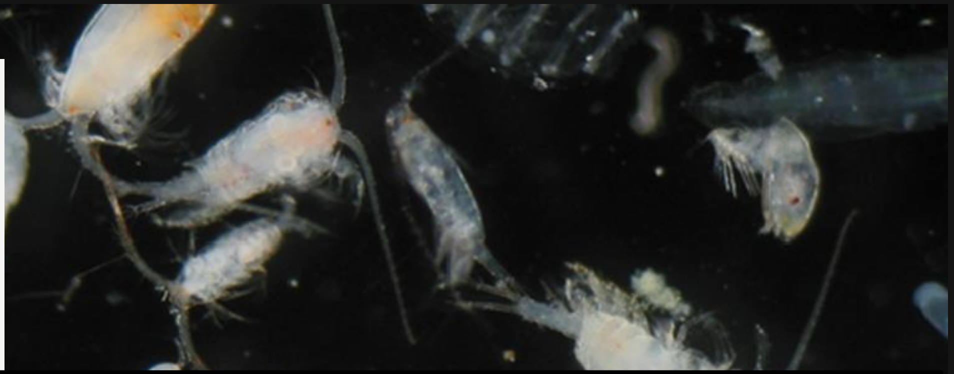
Random Forests

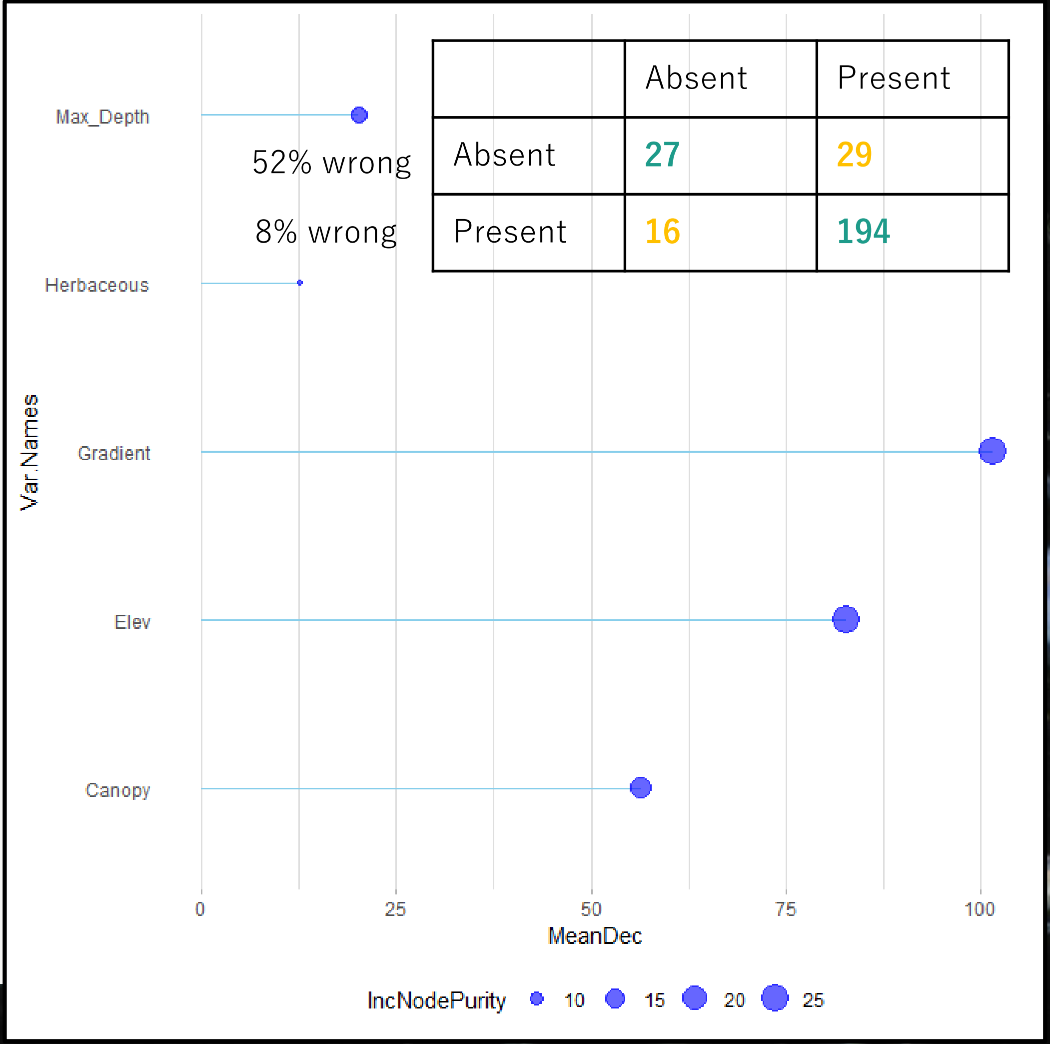
Random Forests uses the concept of **bagging** (**Bootstrap Aggregation**) to create multiple training datasets by randomly sampling with replacement.

Each tree is trained independently, and the final prediction is made by aggregating:

- For classification: Majority voting from all trees.
- For regression: Averaging the output from all trees.

Key Advantage: Reduces overfitting and variance compared to single decision trees by leveraging the power of multiple trees.



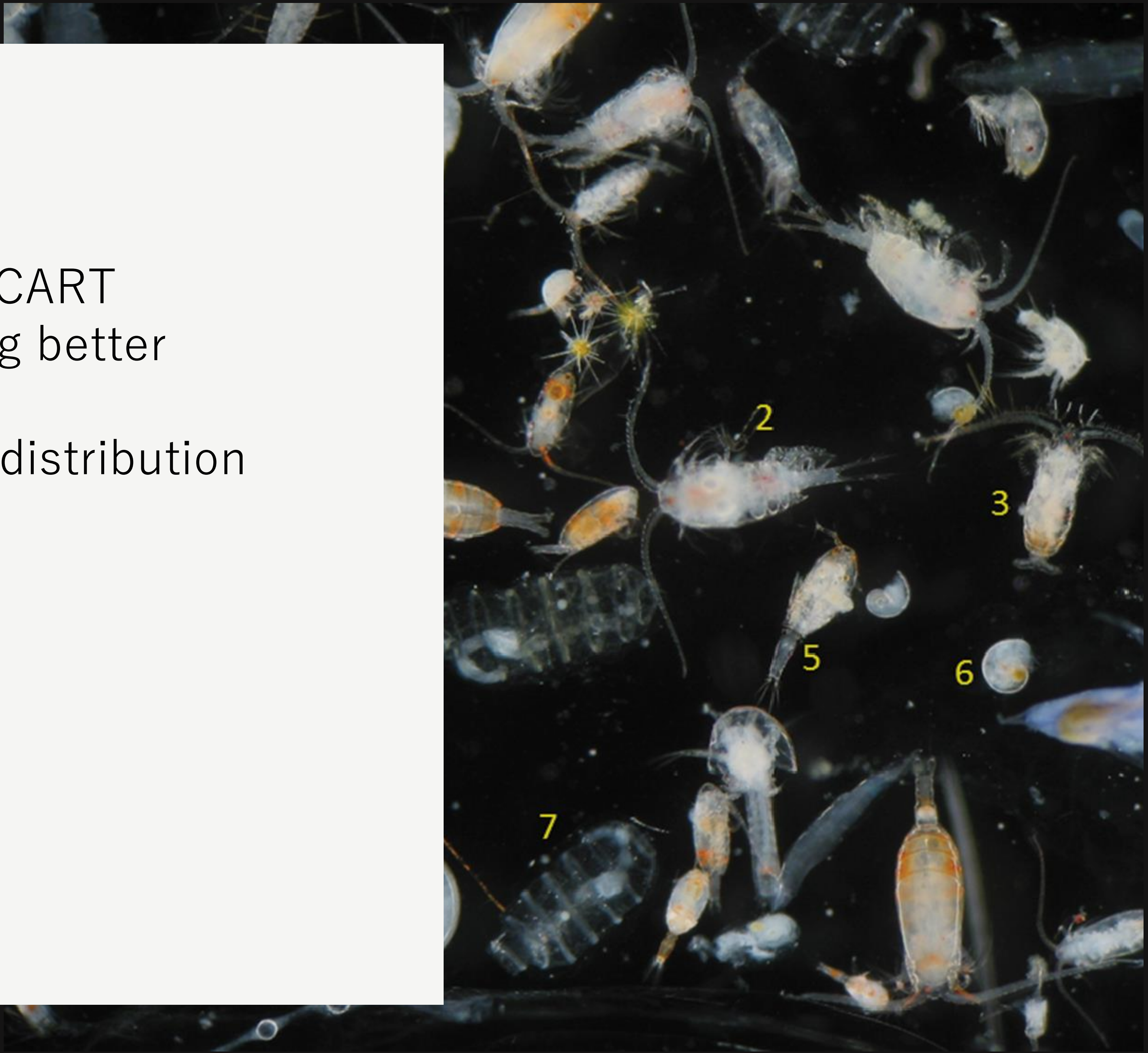


	Absent	Present
Absent	27	29
Present	16	194

Random Forests

Strengths:

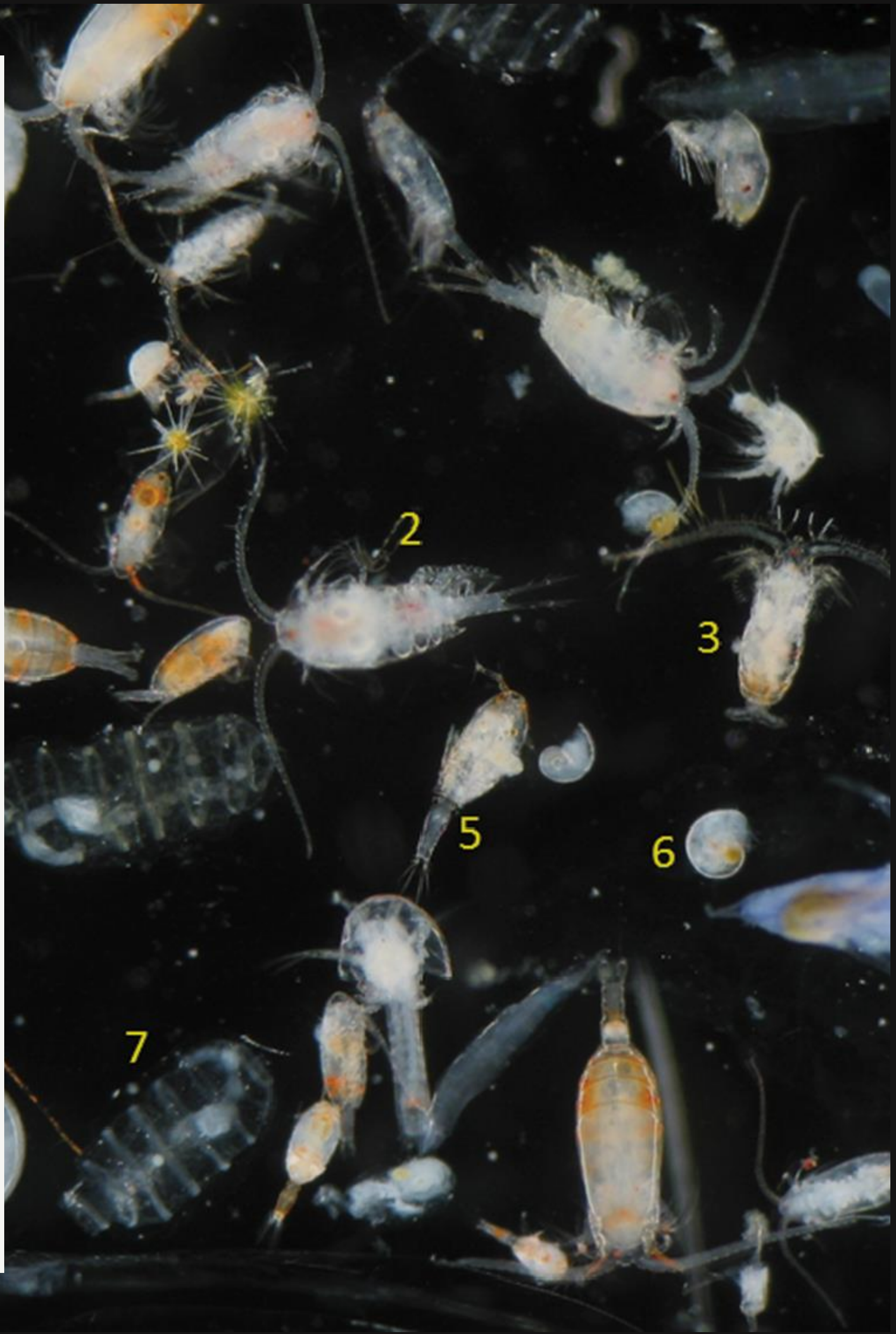
- High accuracy compared to CART
- Handles noise and overfitting better
- Can handle missing data
- No assumptions about data distribution



Random Forests

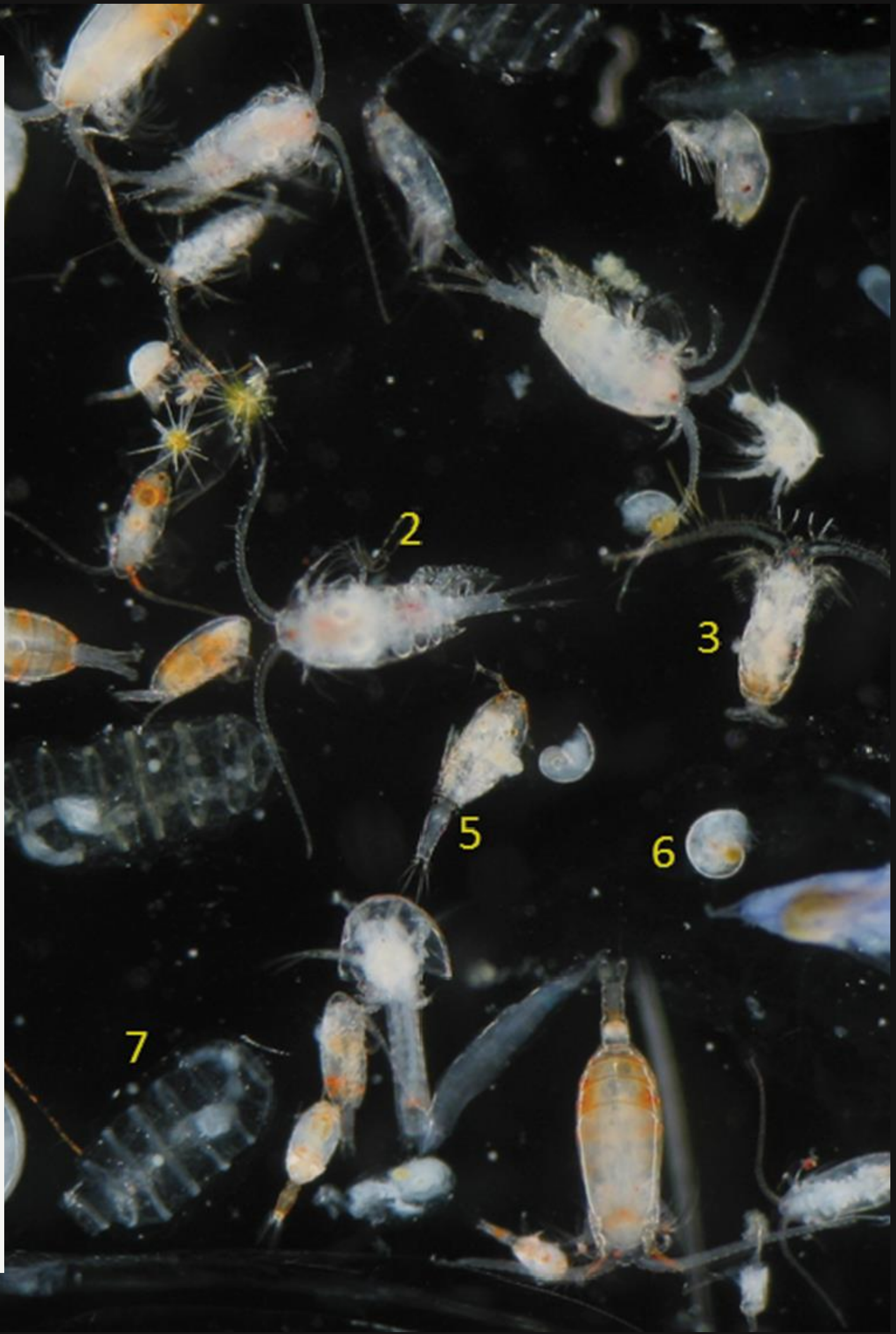
Strengths:
- High accuracy compared to CART
- Handles noise and overfitting better
- Can handle missing data
- No assumptions about data distribution

Limitations:
- Harder to interpret than CART
- Computationally expensive



Conclusion: Summary of Key Points

- **Classification and regression trees (CART)** are examples of **supervised classification** methods
- **Classification trees** classify discrete outcomes
- **Regression trees** classify continuous outcomes
- **Multivariate regression trees** classify multiple response variables at once (but the package is broken ☹)
- **Random forests** is a machine learning method that improves prediction accuracy by combining multiple decision trees



Questions?