Life Sciences and
Facility Management

Institute of
Computational Life Sciences

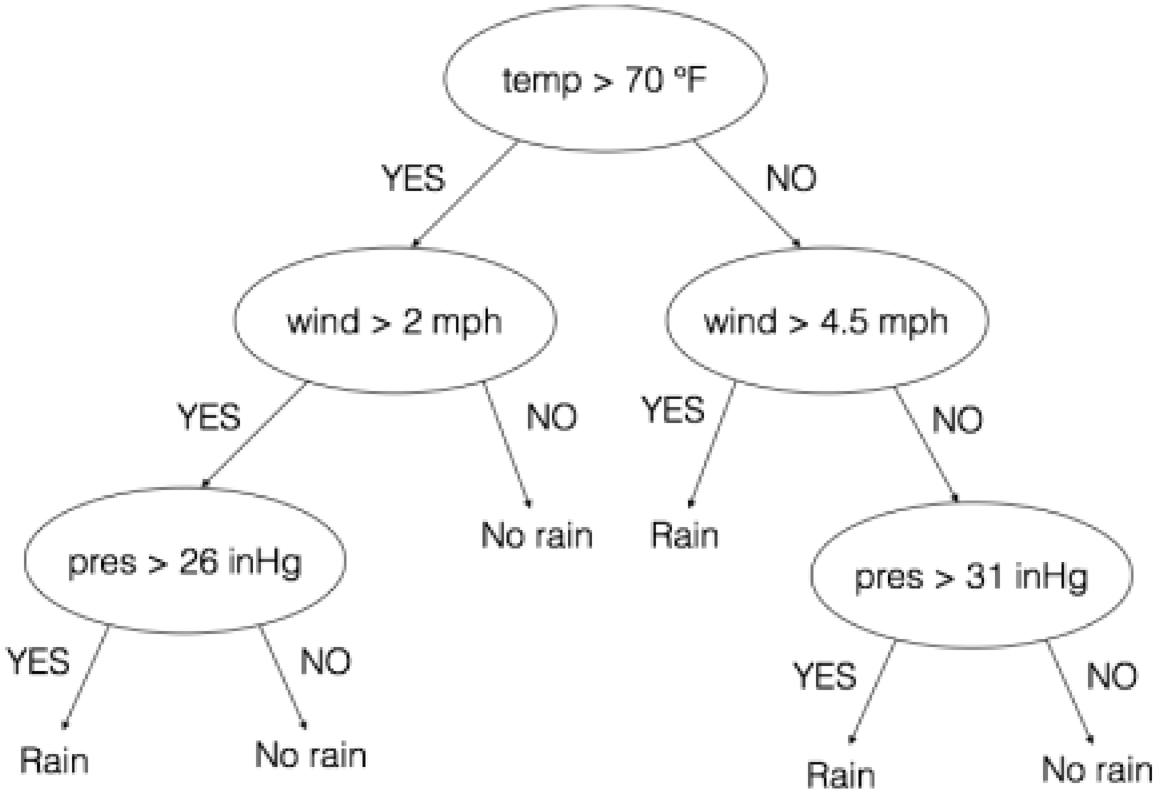
Ensemble learning & Random forests

Machine Learning

Norman Juchler

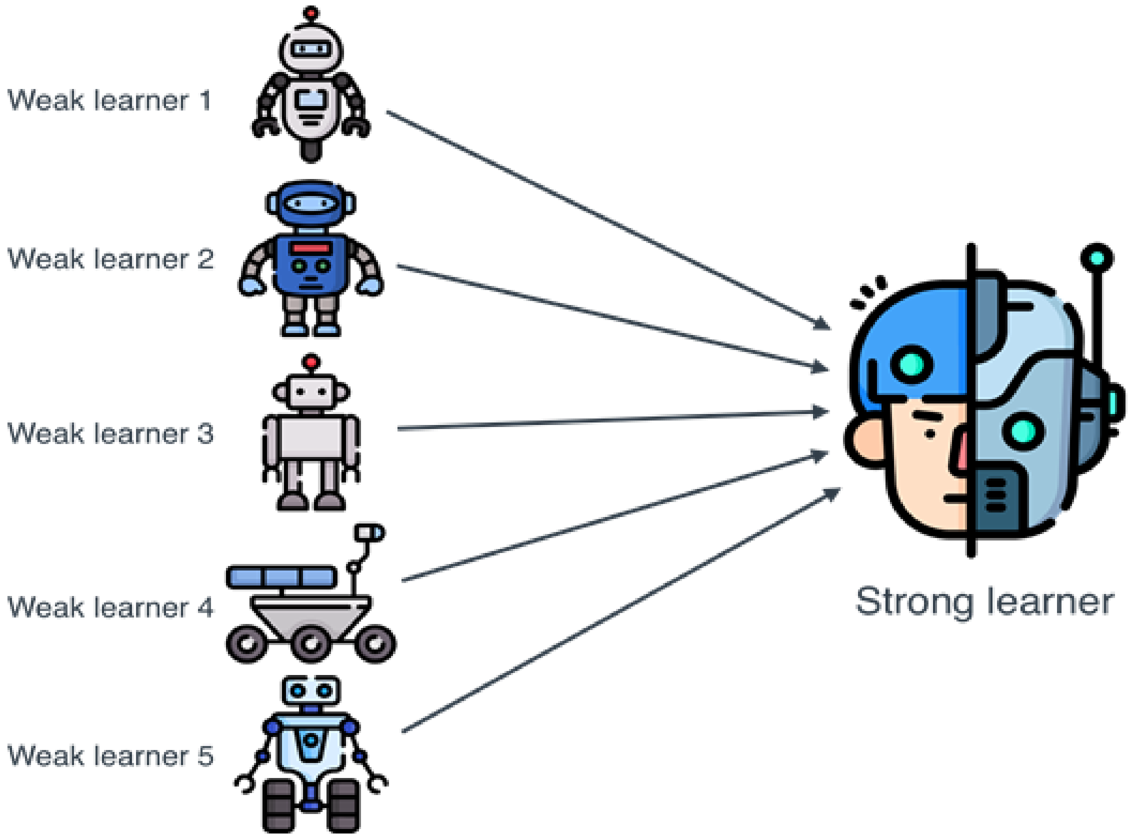
Last week...

Decision trees are simple and versatile, but they are also prone to overfitting and suffer from high variance



Outline

- Theme of today: Combine predictions from multiple models to improve accuracy, stability, and robustness compared to individual models.

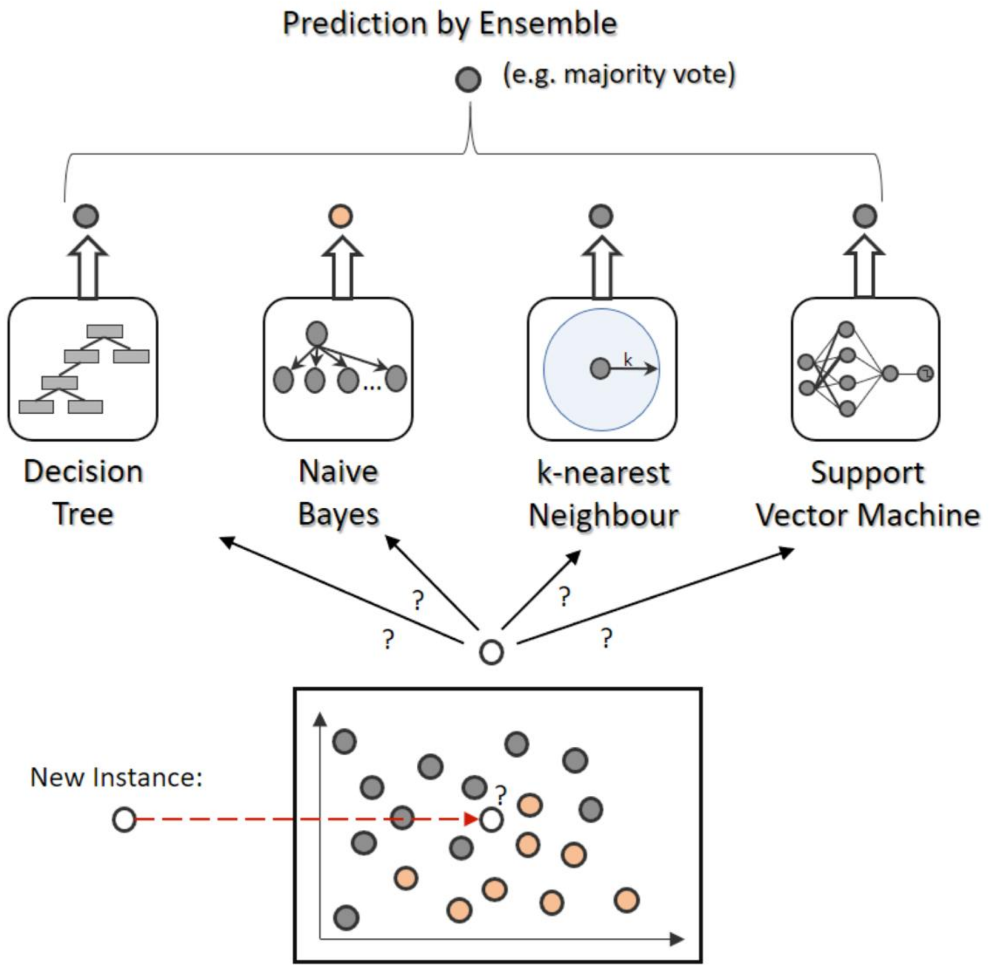


Netflix challenge

- In 2006 Netflix offered a prize of $ 1 Million for an algorithm that could outperform their own algorithm to predict movie ratings of users by 10%.
- The prize was awarded in Sept. 2009.
- The winner team heavily relied on ensemble learning to combine many different algorithms.
- Interesting observations:
  - Adding information about movie genres was not useful for predicting user ratings (probably because the genre is learnt already indirectly).
  - Time of rating turned out to be useful (people who rate a movie immediately after watching prefer different movies than people who rate them later).

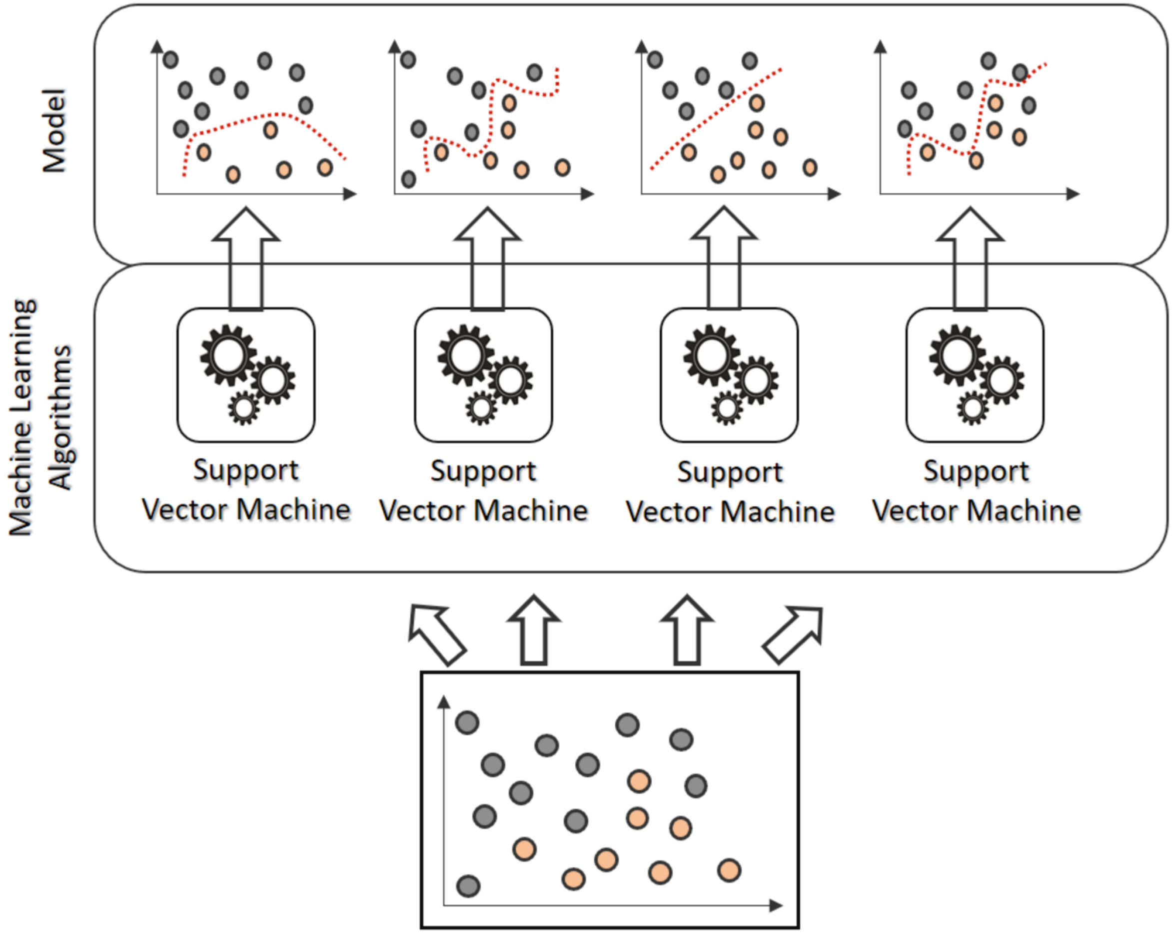
Model ensembles

- Different algorithms have different strengths.
- Each algorithm may perform better in certain regions of the feature space than others.
- By merging their predictions, ensembles can achieve greater accuracy than individual models alone.



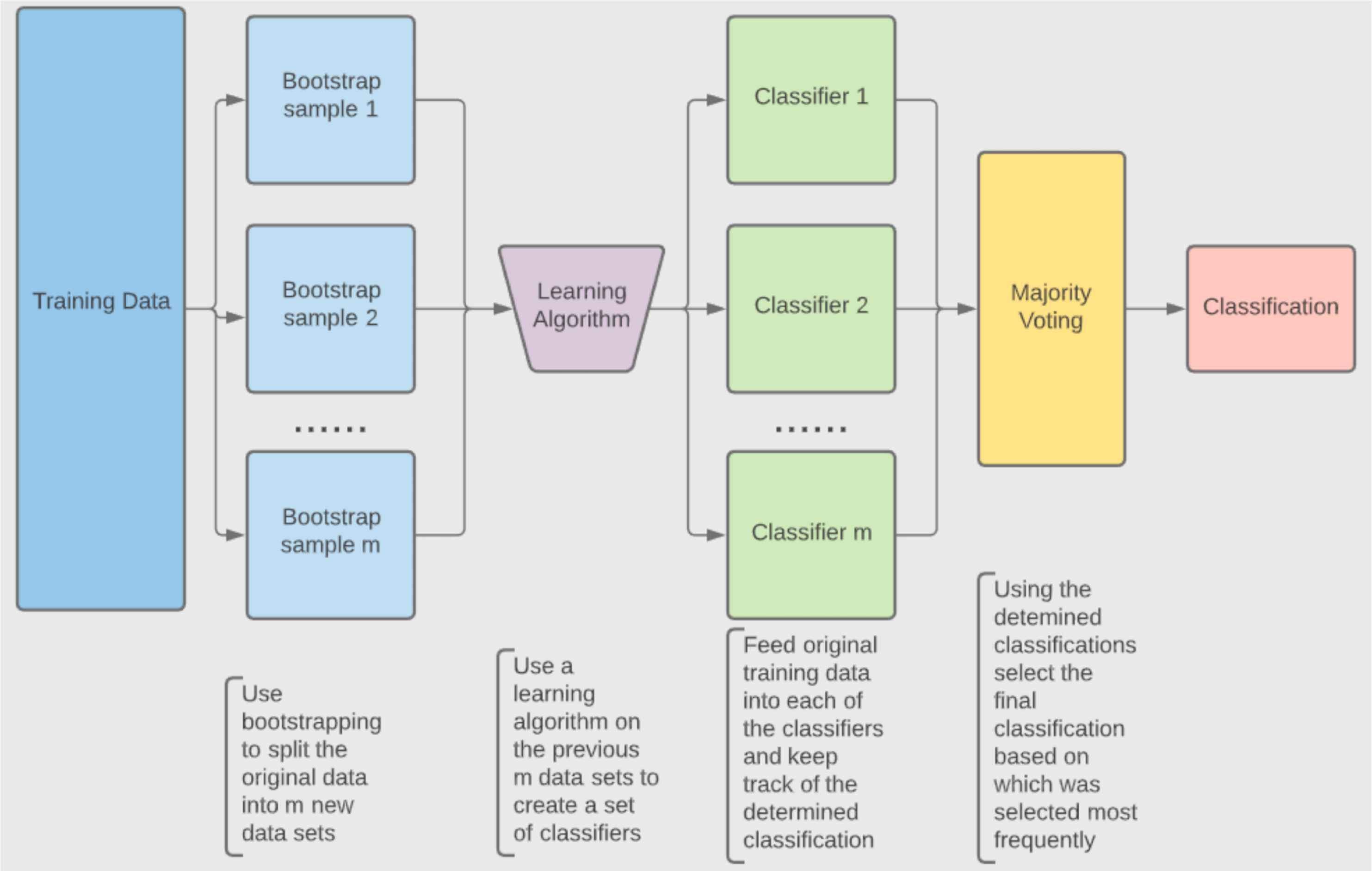
Bagging

- Short for bootstrap aggregating
- An ensemble technique where the constituent models are
  - of the same type, but
  - trained on different randomly sampled data subsets.
  - Bootstrapping ⇔ sampling with replacement
  - Final prediction by aggregation (clf: majority vote, reg: or averaging)
- Overall model becomes more robust/reduces overfitting.
- All constituent models can be trained in parallel.



Bagging

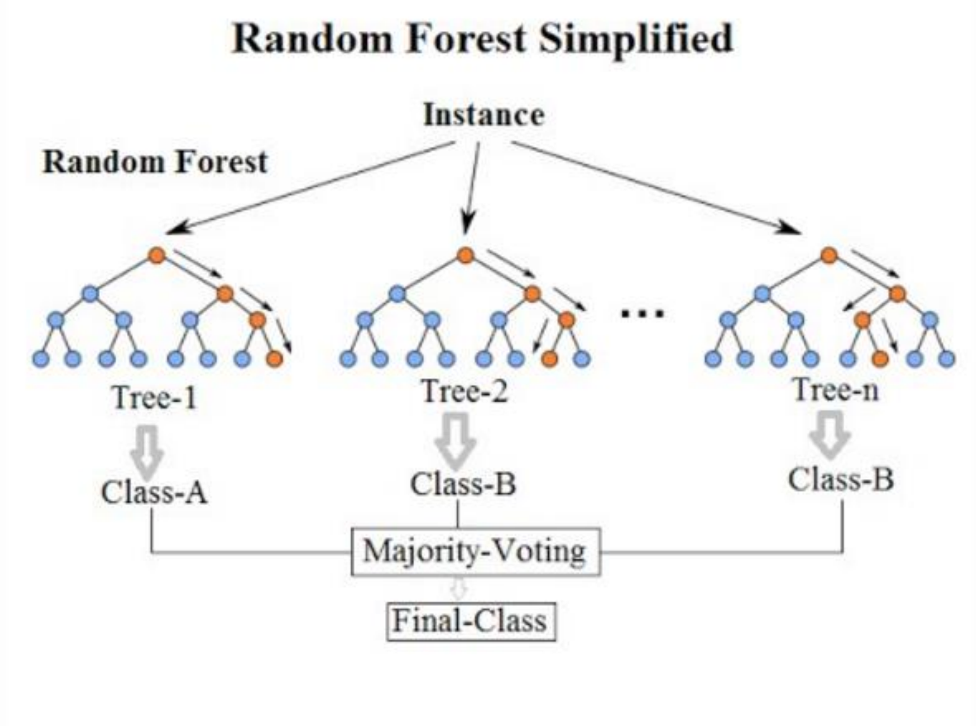
- Short for bootstrap aggregating
- Bootstrapping: Create new datasets by random sampling <u>with replacement</u>



Source of illustration: Wikipedia

Random forest: Algorithm

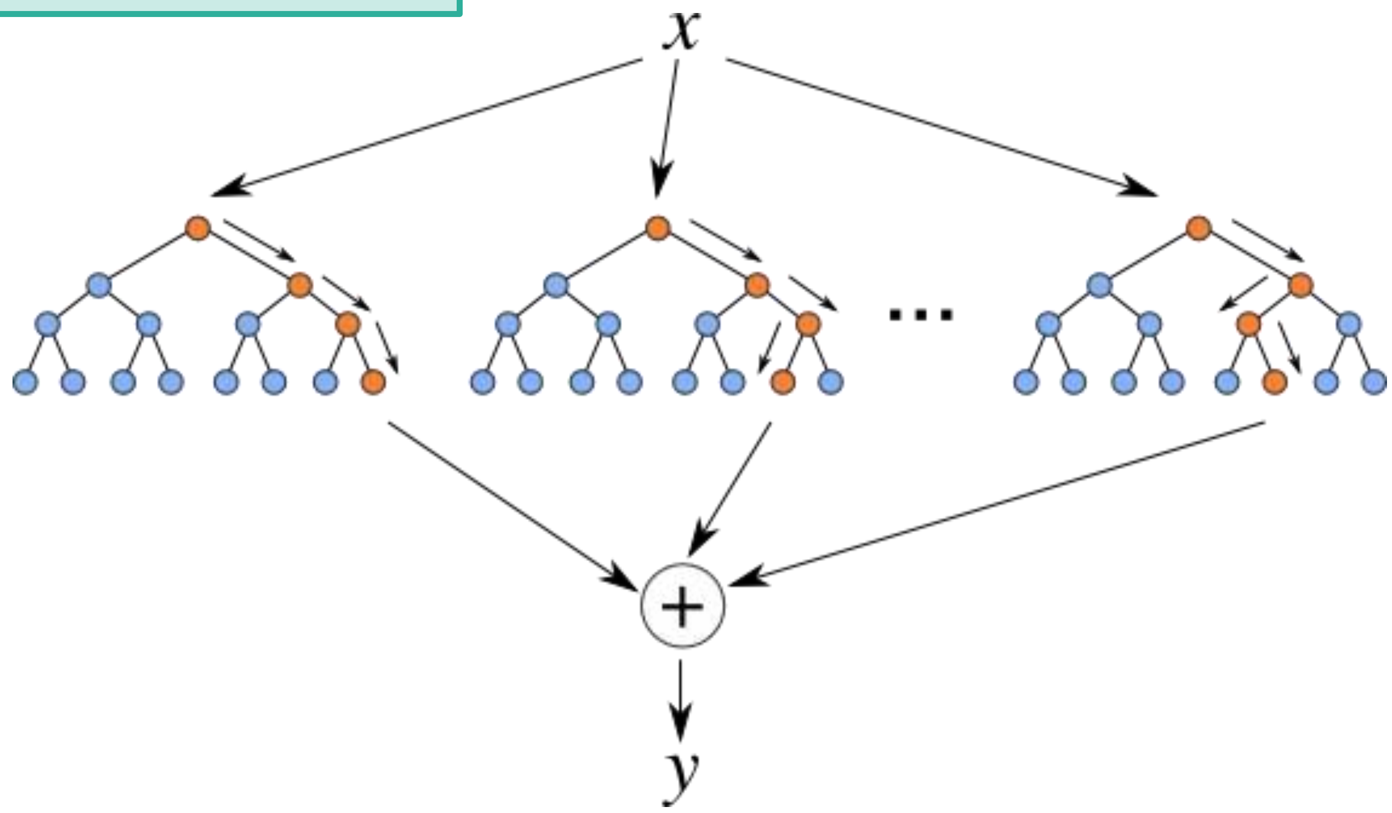
- An ensemble of decision trees trained with bagging.
- Many decision trees are generated during training.
- Tree bagging is used to train trees (reduces variance and overfitting).
- Feature bagging: Each tree only uses a random subset of features, which ensures that trees remain decorrelated.
- Used for both regression and classification
- Training is easily parallelizable
- Robust to noise and outliers



Random forest: Parameters

- **n_estimators**:
- **max_depth**:
- **max_samples**:
- **max_features**:
- **random_state**:

What meaning do these parameters have?



Random forest: Parameters

- **n_estimators**: The number of trees in the forest; More is better (until saturation) but takes longer to train.
- **max_depth**: Reducing it might help against overfitting and increase speed; Increasing it enables higher model complexity.
- **max_samples**: Reducing it from its default value of 1.0 will increase the diversity of trees.
- **max_features**: How many features to use per tree. Default value of the square root of the number of input features.
- **random_state**: Makes the model’s output replicable.

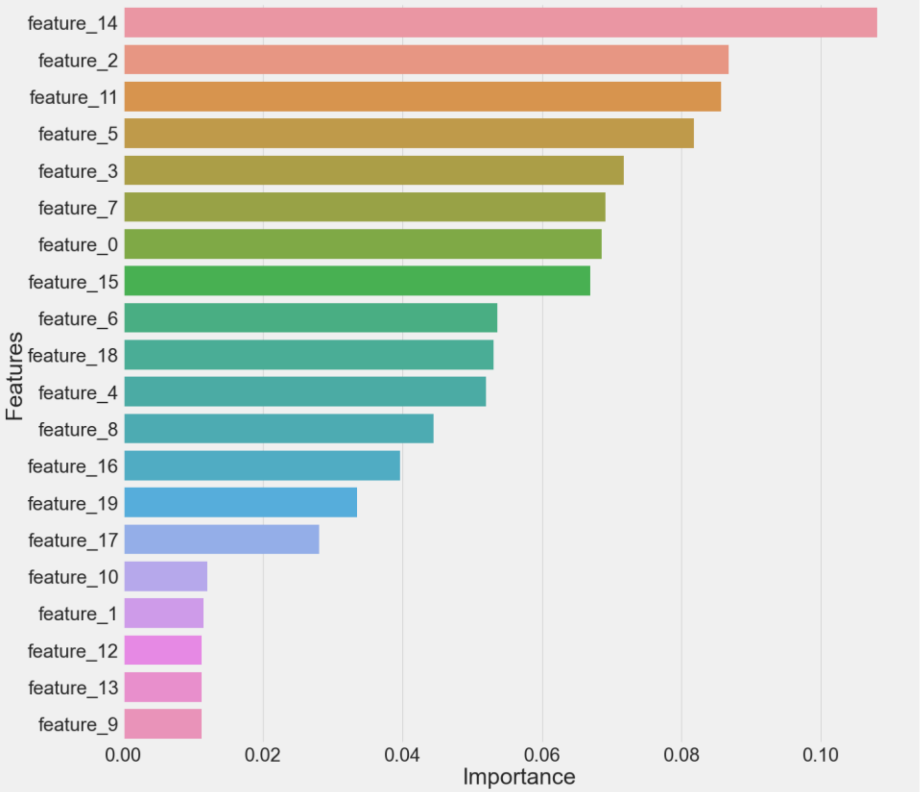
Fight the fire with fire...

- Predictions of a decision tree are highly sensitive to noise in its training set.
- The average of many trees is less sensitive to noise, if the trees are not correlated.
- Simply training many trees on a single training set would give correlated trees.
- Bagging is a way of de-correlating the trees, so is feature-bagging.

Image source: Link

Random forest: Feature importance

- Feature importance: A measure of how influential each feature is in predicting the target outcome.
- Computation:
  - Measure the reduction in impurity (e.g., Gini impurity or entropy) provided by each feature at each split.
  - Aggregate across trees and nodes: The total importance of a feature is computed by summing its contributions across all nodes where it was used to make a split.
- Interpretation: Features that split the data better are more useful and "important"



Visualization of the importance of different features in a random forest. Source

Source

Random forest: Pros & cons

Advantages

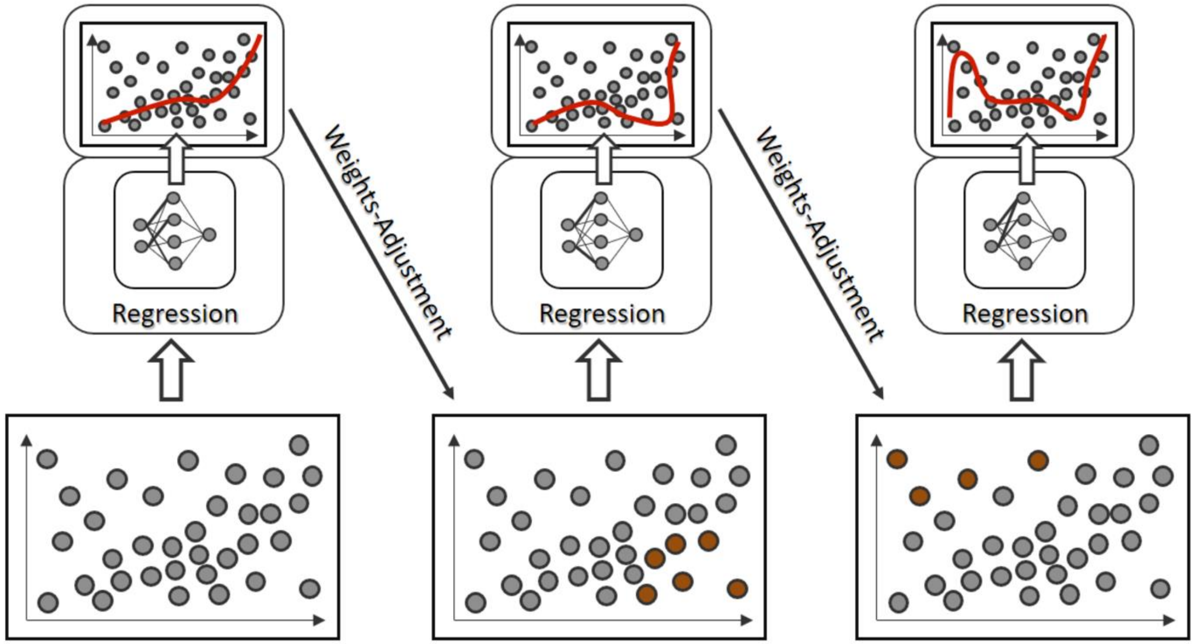
- Ease of use: RF often work well right away (with default parameters)
- High performance: RF demonstrate lower bias and variance (as DTs)
- Robustness: higher resilience to overfitting, noise and outliers than DTs
- Training and prediction parallelizable (trees can be processed independently)

Disadvantages

- Loss of interpretability compared to decision trees
- Computationally intensive, especially for large datasets or many trees

Boosting

- An ensemble method in which the models are trained sequentially, with more weight given to misclassified samples.
- Advantage: Performance often higher than bagging
- Disadvantages (compared to bagging):
  - More susceptible to overfitting
  - Not parallelizable



- Popular methods:
  - AdaBoost (focus on misclassified cases)
  - Gradient boosting (focus on residuals)
  - XGBoost (popular instance of GB)

Further ~~reading~~ watching

- StatQuest: Decision and classification trees (18 min)
- StatQuest. Decision trees part 2 (5 min)
- StatQuest: Regression trees (22 min)
- StatQuest: Random forests part 1 (9 min)
- StatQuest: Gradient-boosted trees part 1 (15 min)
- StatQuest: AdaBoost (21min)
- StatQuest: XGBoost part 1 (25 min)