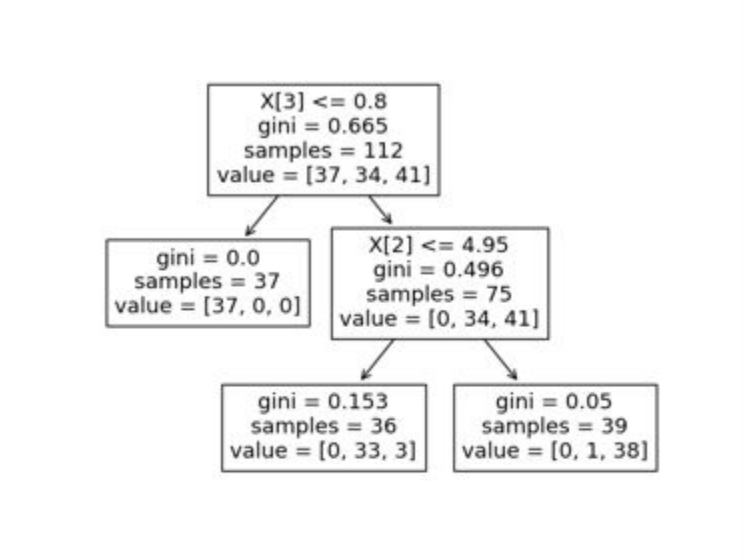
Click-through prediction with decision tree

2021.02.24
Data eXperience Lab, Winter Seminar
2018312824 Ryu Chaeeun

OUTLINE

1. Implementation of Decision Tree (Concept)
2. Random Forest, Bagging, Boosting
3. Click-Through prediction with decision tree classifier



Implementation of Decision Tree (Concept)

- Bagging: Averaging Trees
- Random Forests: Cleverer Averaging of Trees
- Boosting: Cleverest Averaging of Trees

Bagging, RandomForest, Boosting

Ensemble

Ensemble: ML technique that **combines several base models** in order to produce one optimal predictive model. The main principle behind the ensemble model is that **a group of weak learners** come together to **form a strong learner.**

Bagging, Random Forest, Boosting are some of Ensemble methods.

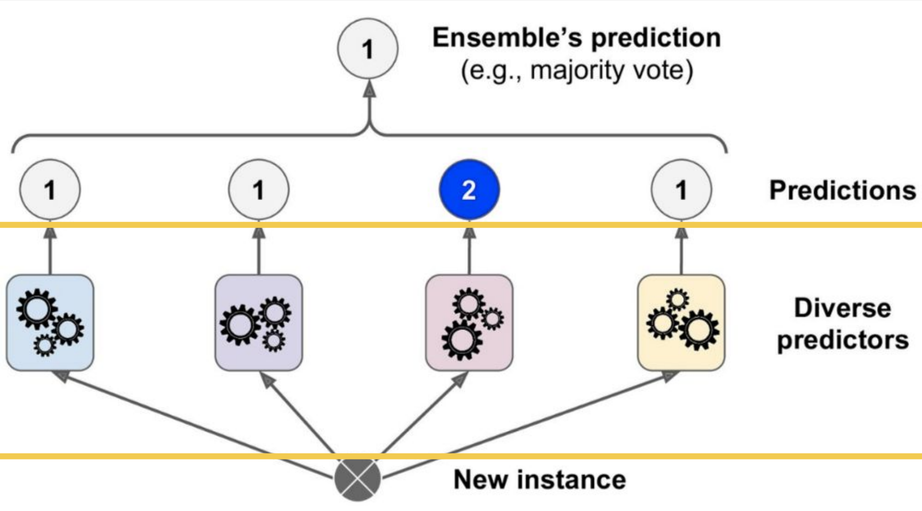


Figure 7-2. Hard voting classifier predictions

Multiple classifiers

Recap: Properties of Trees

Properties of Trees

<Advantages>

O. **Fast** to train

O. Small trees are easy to **interpret**

O. **Simple** to understand

O. robust to **noisy data** (outliers are prone to get pruned)

<Disadvantages>

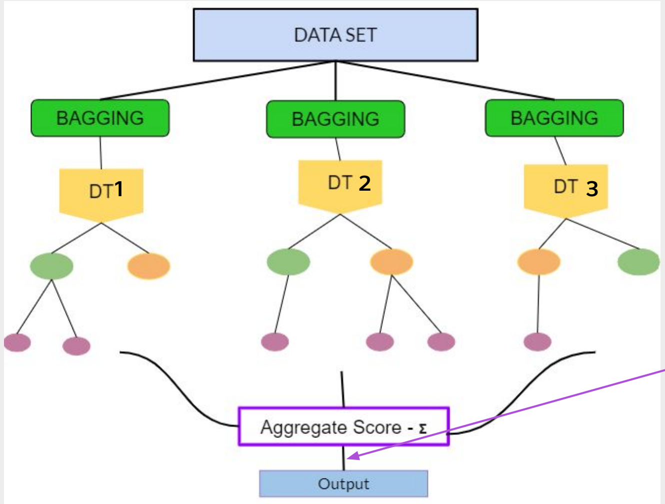
X. large trees are hard to interpret

X. Often, prediction performance is poor(high variance -> prone to **overfitting**)

Bagging

Bagging (Bootstrap Aggregation)

1. Draw different sets of training samples randomly with replacement
2. Each set is used to train an individual classifier
3. Result of the classifiers are combined -> majority vote to make a final decision



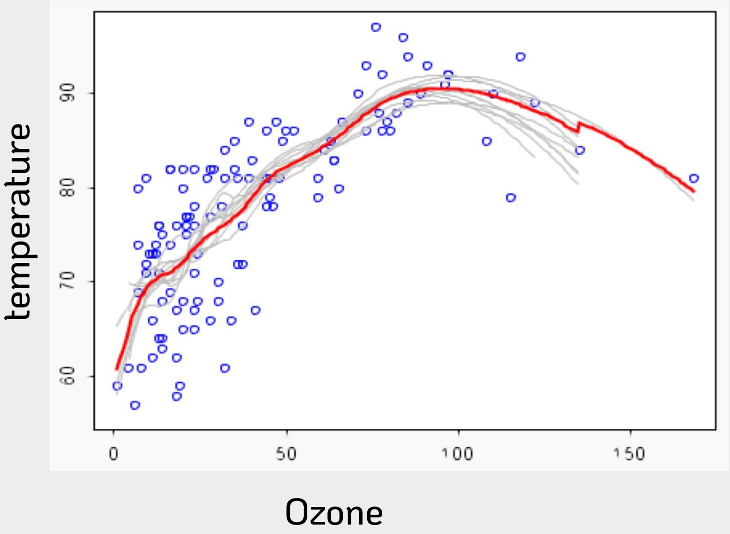
Majority voting
ex.)
DT1: a, DT2: a, DT3: b
Result: a

Bagging

Bagging Advantage

- High expressiveness: by using full trees, able to **approximate complex functions** and decision boundaries.

Bagging improves prediction accuracy at the expense of interpretability.



Bagging

Bagging Advantage

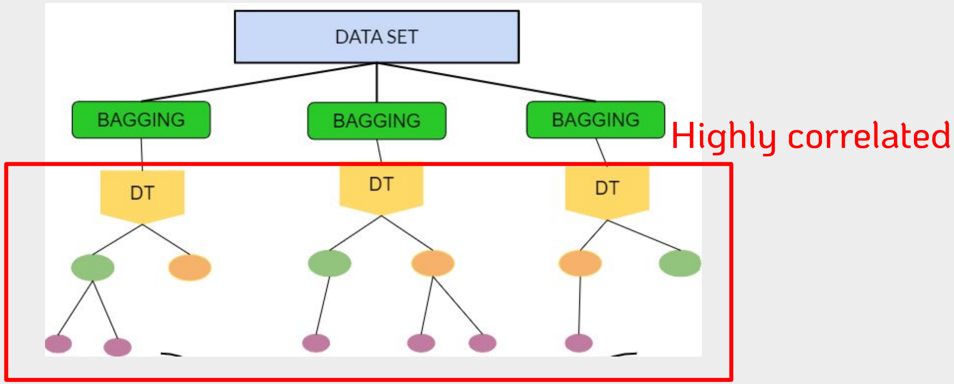
- High expressiveness: by using full trees, able to approximate complex functions and decision boundaries.

Bagging Drawback

- When one or more features are **strong** indicators, each tree's output becomes **<u>highly correlated.</u>**
  -> Aggregating multiple trees will **not make much difference** compared to one tree classifier.

Let's reduce correlation

<u>Random Forest</u>



Random Forest

Random Forest(Feature-based Bagging)

: Bagging, but considers **random subset of features** when searching for best splitting point at each node

-> at each tree split, **a random sample of m features** is drawn, and only those m features are **considered for splitting.**

Features: height, weight, race, gender

<Bagging>

Node

Consider **all** features in sampled dataset
-> consider: height, weight, race, gender

<Random Forest>

Node

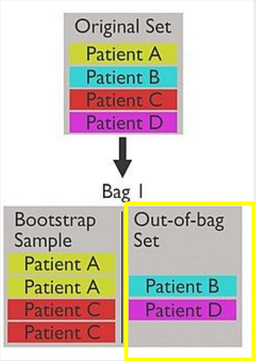
Consider **random subset of** features in sampled dataset
-> consider: weight, gender

Random Forest

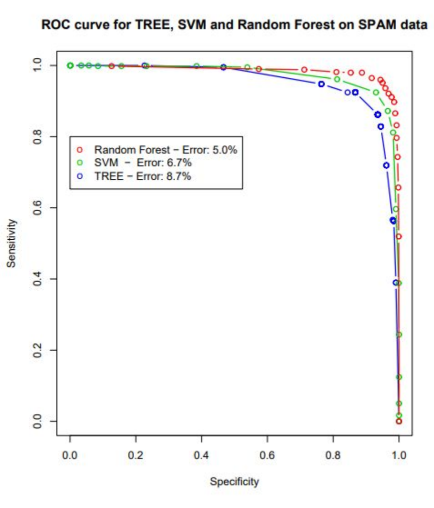
How to calculate performance of RF: <u>Out-of-bag error rate</u>

: the error calculated using the bootstrap sample left out.

Cease training RF when out-of-bag error stabilizes



Evaluate performance

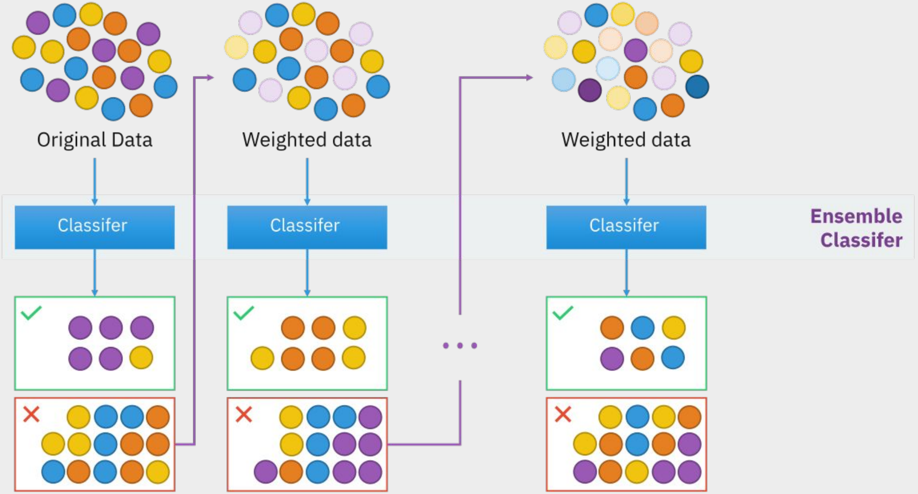


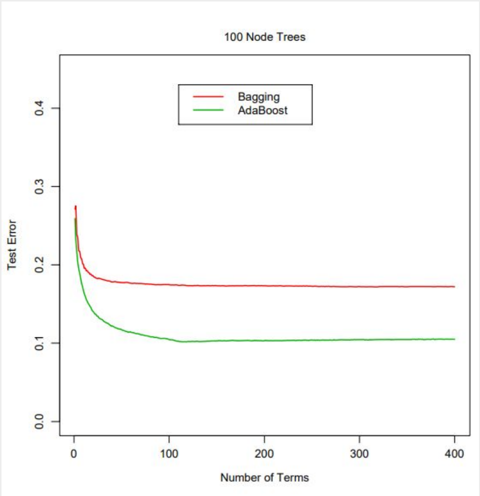
Boosting

Boost

- Iteratively add simple **weak classifiers**(slightly better than random) to improve classification performance
- Assign each training example equal weights
- Train a weak classifier ->**reweight** (higher weight on mistakes, lower weight on correct ones)->repeat
- Add weak classifiers to final strong classifier(when adding, weighted in a way that is related to the weak learners' accuracy.)

Python Libraries: LightGBM, XGBoost, AdaBoost





Click-through prediction with decision tree classifier

Bagging vs. Boosting

Bagging	Boosting
The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
Aim to decrease variance, not bias.	Aim to decrease bias, not variance.
Each model receives equal weight.	Models are weighted according to their performance.
Each model is built independently.	New models are influenced by the performance of previously built models.
Different training data subsets are randomly drawn with replacement from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.
If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
Example: The Random forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques