Outline

- ❖ Introduction
- ❖Ensemble Learning
- ❖Boosting Algorithms
- ❖ Demo in Python

Part 1:
Overview of Decision Trees

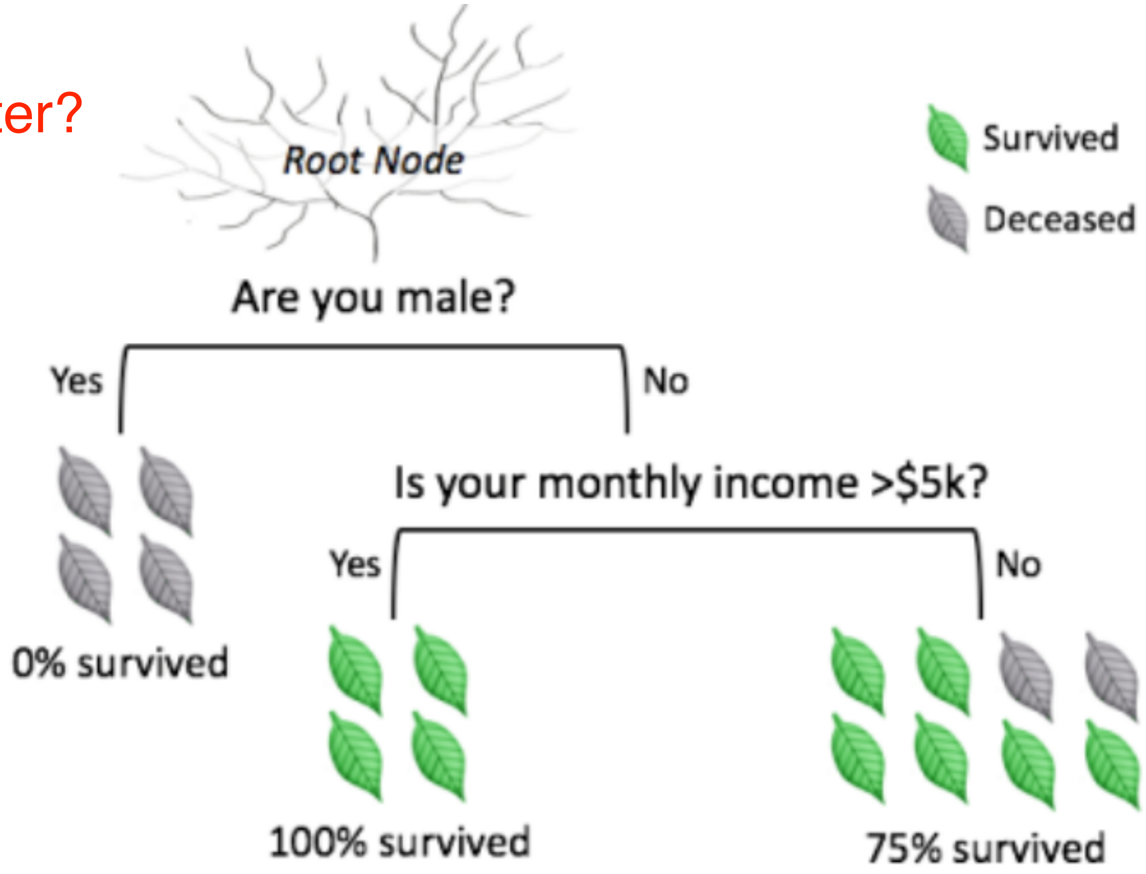
Decision Trees

Would you survive a disaster?

https://www.kdnuggets.com/2016/09/decision-trees-disastrous-overview.html

Decision Trees

Would you survive a disaster?



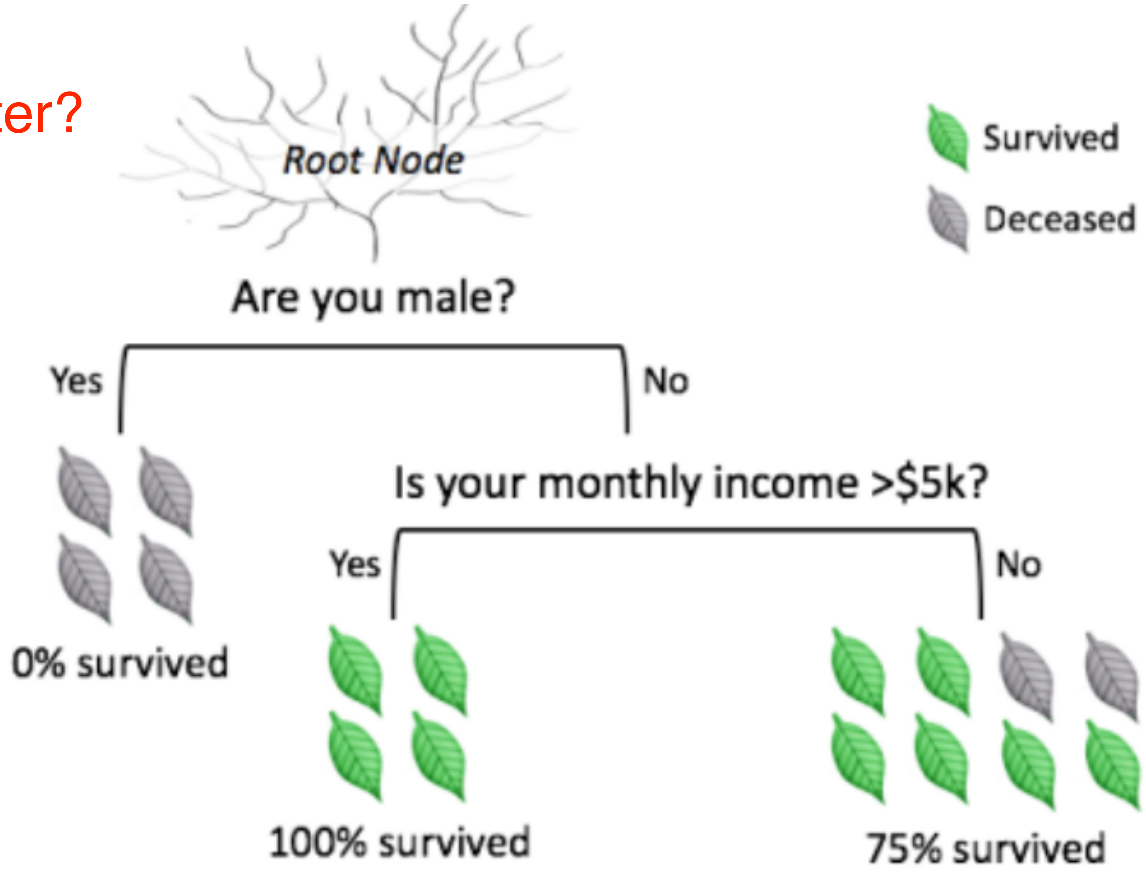
https://www.kdnuggets.com/2016/09/decision-trees-disastrous-overview.html

Decision Trees

Would you survive a disaster?

Stopping criteria:

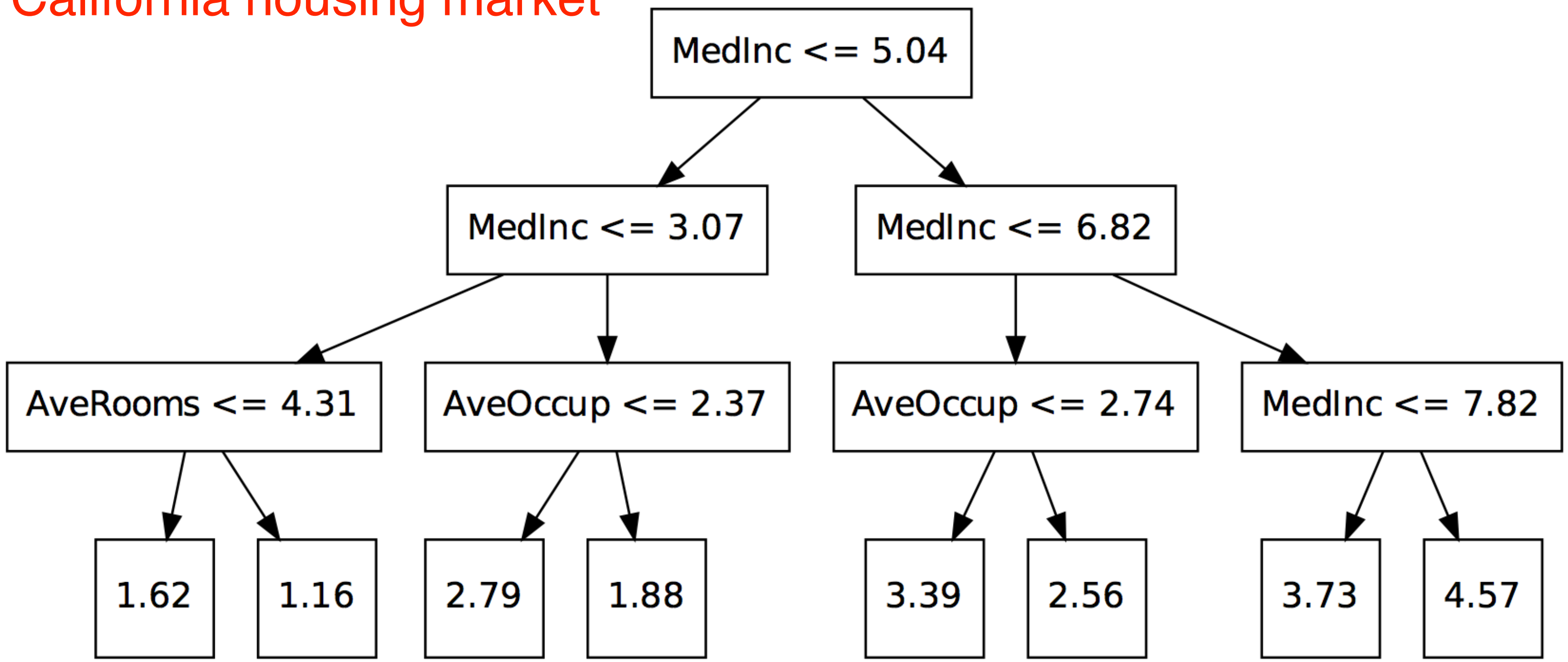
- Stop when data points at the leaf are all of the same class
- Stop when the leaf contains less than K data points
- Stop when further branching does not improve homogeneity beyond a minimum threshold



https://www.kdnuggets.com/2016/09/decision-trees-disastrous-overview.html

Decision Tree Overview

Predictions for California housing market



Source: http://www.slideshare.net/DataRobot/gradient-boosted-regression-trees-in-scikitlearn

Decision Trees: Practical Use

Strengths

- Non linear
- Robust to correlated features
- Robust to feature distributions
- Robust to missing values
- Simple to comprehend
- Fast to train
- Fast to score

Weaknesses

- Poor accuracy
- Cannot project
- Inefficiently fits linear relationships

https://github.com/mlandry22/boosting-austin-sigkdd-talk-20160803

Part 2:
Overview of Ensemble Learning

Ensemble Learning

A **Mixture** of experts or "base learners" combined to solve the same learning problem.

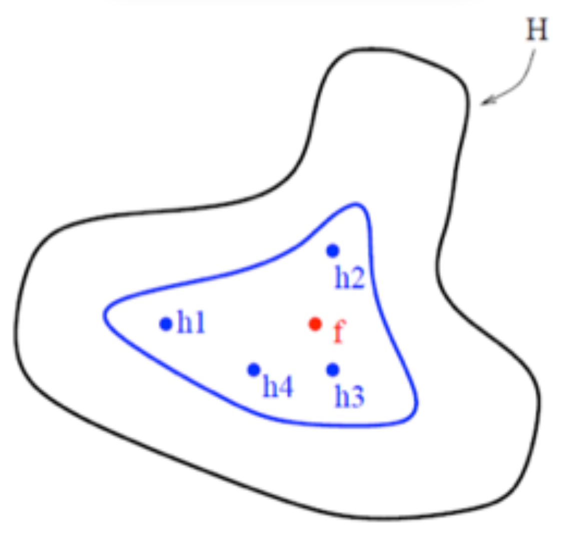
Construct **a set of hypotheses** and combine them to use .

The **generalization** ability is much stronger than that of base learners.

- Generating the base learners
  - Bagging
  - Boosting
  - Stacking
- Combining the base learners
  - Voting
  - Averaging
- Ensemble Model
  - Classification
  - Regression

Why Ensembles Superior to Singles?

Statistical

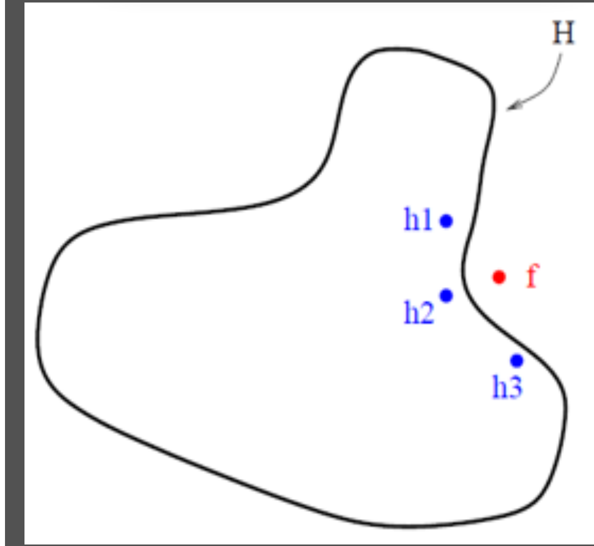


No sufficient training to choose a single best learner.

Computational

Imperfect search processes result in sub-optimal hypotheses.

Representational

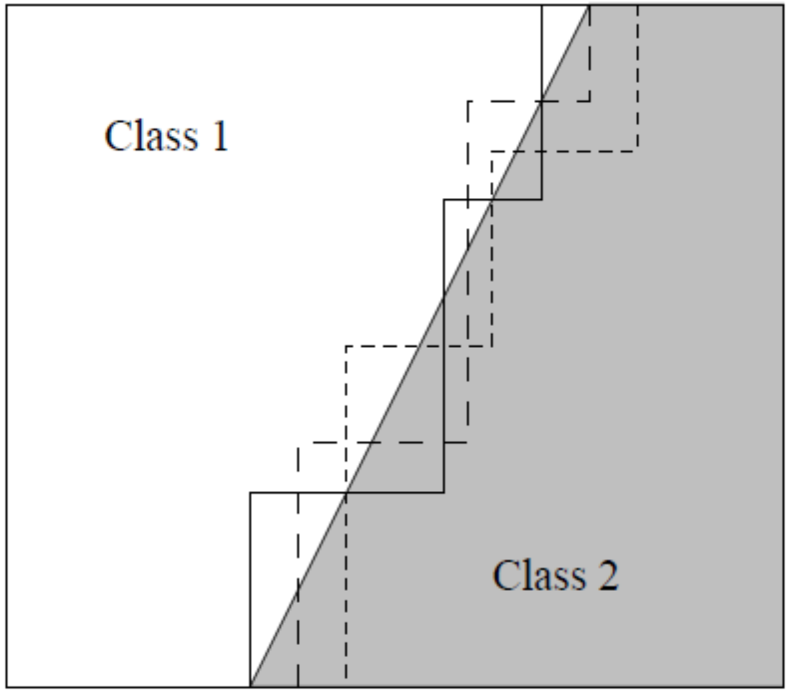


The true function cannot be represented by any of the hypothesis in H.

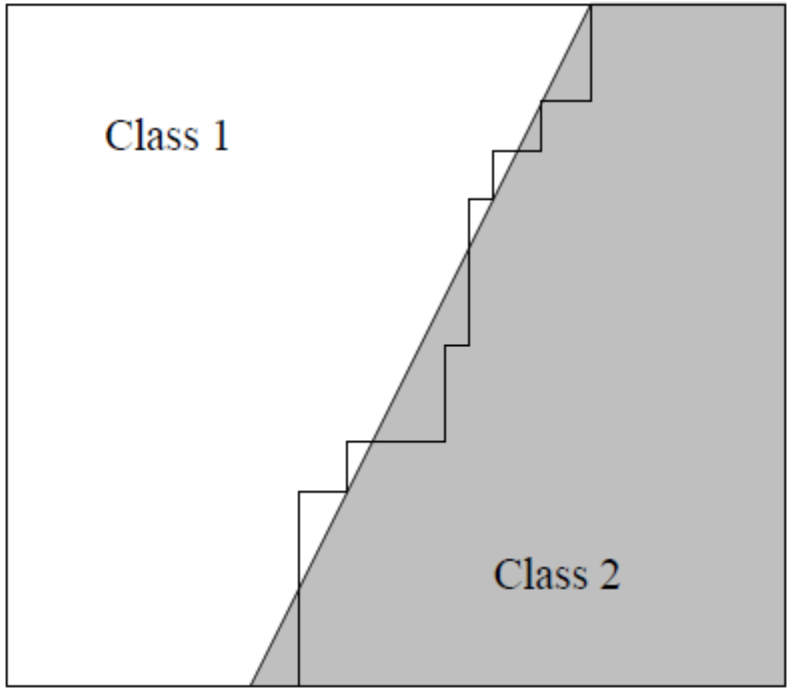
Dietterich, T. Ensemble methods in machine learning. Multiple classifier systems, Vol. 1857 of Lecture Notes in Computer Science. 2000

Illustration of the representation issue

Diagonal Decision boundary with decision trees base learners



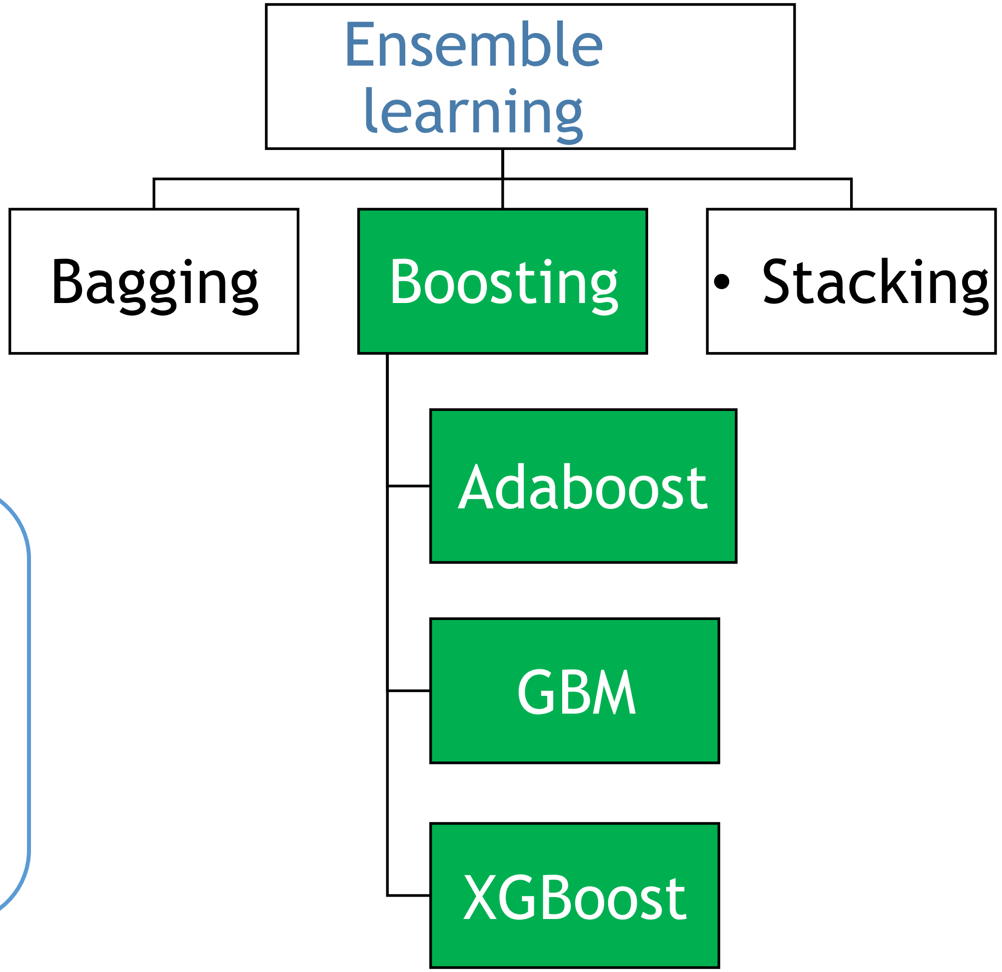
Three staircase approximations



The voted decision boundary

Dietterich, T. Ensemble methods in machine learning. Multiple classifier systems, Vol. 1857 of Lecture Notes in Computer Science. 2000

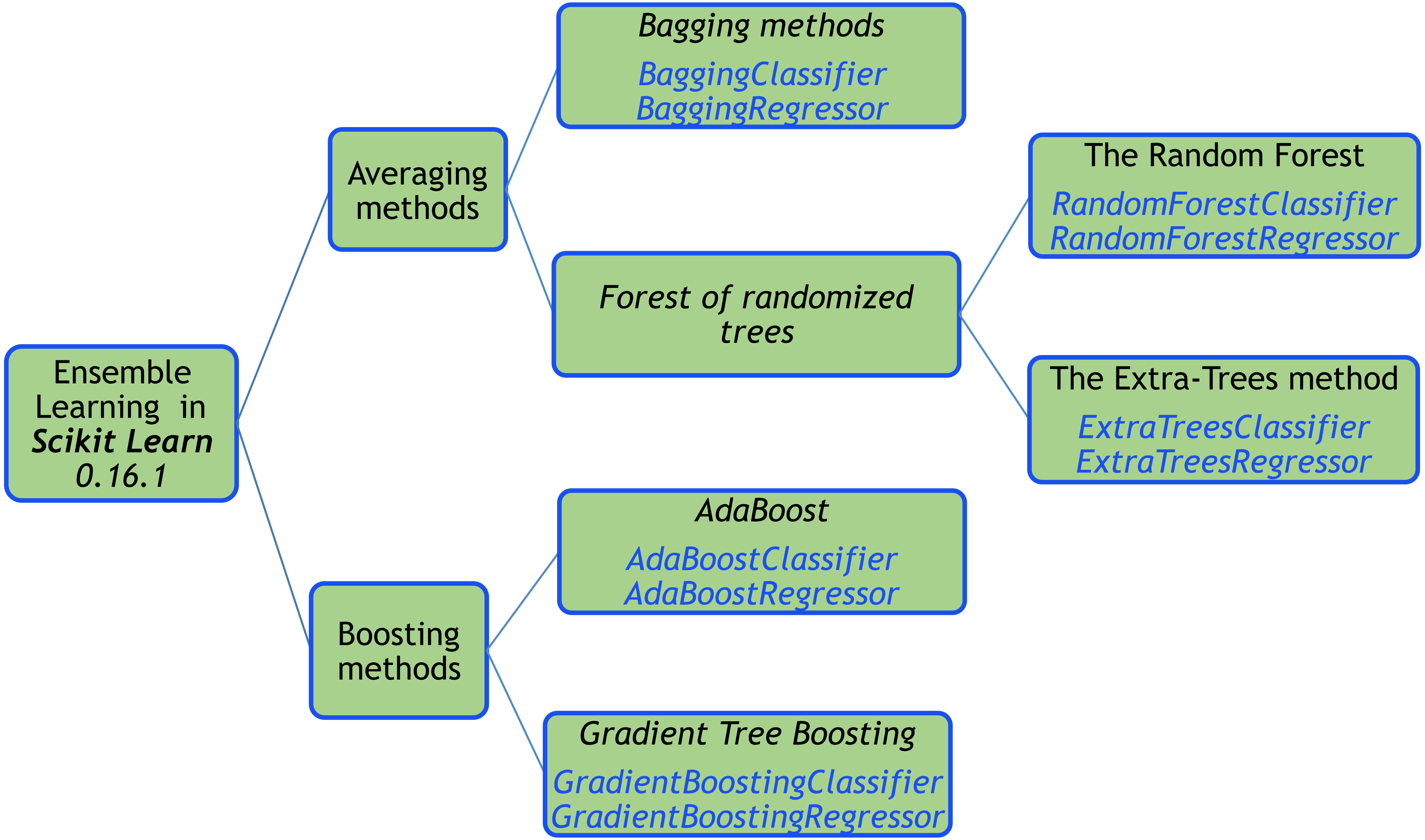
Boosting is one Type of Ensemble Learning



- Bagging & Boosting are considered homogeneous. They use a single base learning algorithm.

- Staking uses different kinds of base learners

Ensemble Learning in SKlearn



Bagging

- Trains a number of base learners each from a different *bootstrap sample (***B**ootstrap **AGG**regat**ING)**

- Each dataset is generated by sampling from the total N data examples, choosing N items uniformly at random **with replacement.**

- For a bootstrap sample, some training examples may appear but some may not.

- The outputs of the models are combined by:
  - Averaging (in the case of regression)
  - Voting (in the case of classification)

- Example: Random forests

Bagging

Algorithm Bagging

Input: Required ensemble size T

Input: Training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$

for $t = 1$ to T **do**

  Build a dataset S_t, by sampling N items, randomly *with replacement*

  Train a model h_t using S_t, and add it to the ensemble.

end for

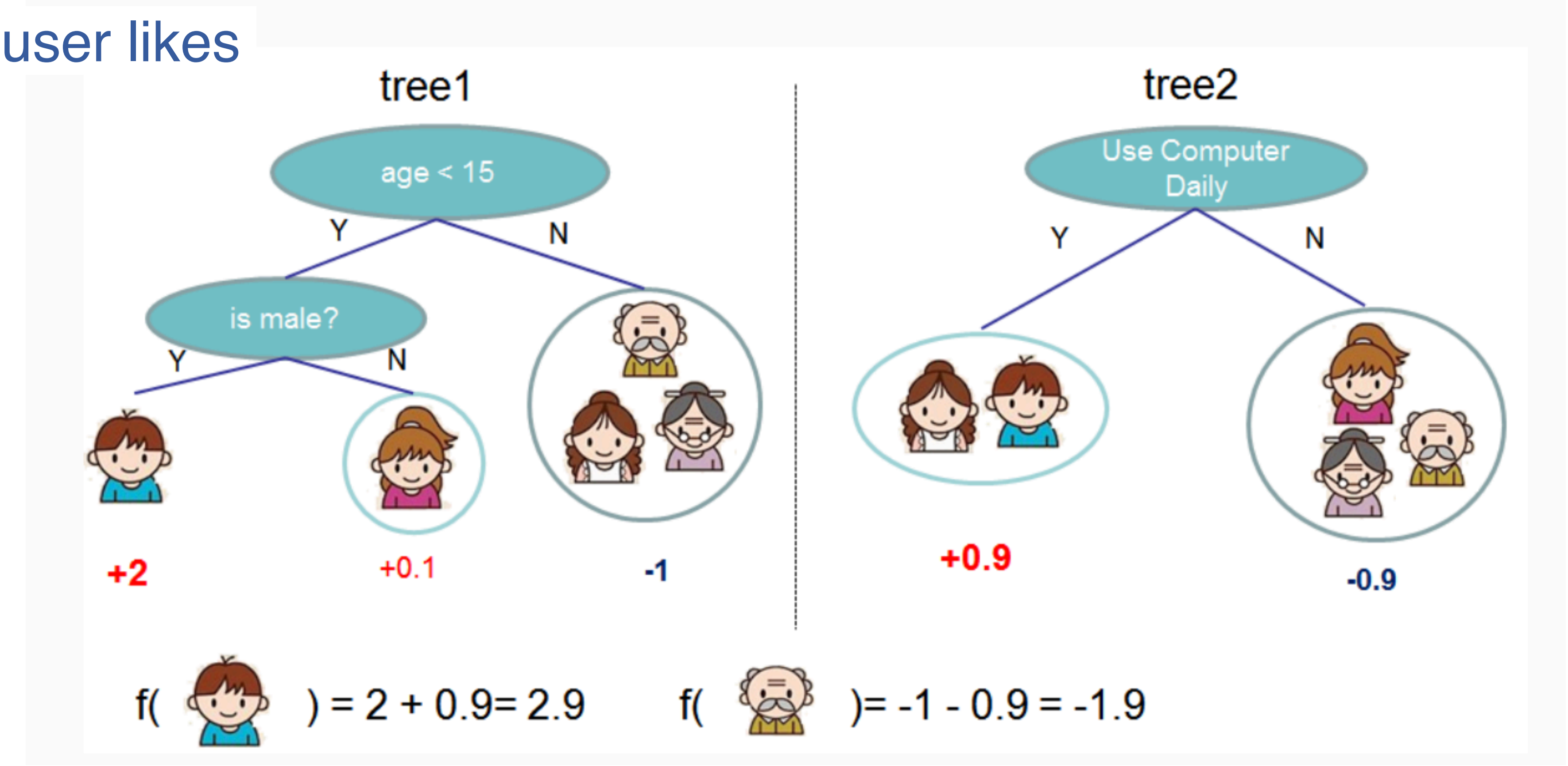
For a new testing point (x', y'),

If model outputs are continuous, combine them by averaging.

If model outputs are class labels, combine them by voting.

Tree Ensemble Model

Predict whether a given user likes computer games or not



https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html

Decision Trees & Random forests in SKlearn

`sklearn.tree`.**DecisionTreeClassifier**

Input parameters:

criterion
max_depth
min_samples_split
min_samples_leaf
max_features
max_leaf_nodes
…

`sklearn.ensemble`.**RandomForestClassifier**

Input parameters:
n_estimators
+
criterion
max_depth
min_samples_split
min_samples_leaf
max_features
max_leaf_nodes
…

Variants of Bagging

Pasting	When random subsets of the dataset are drawn as random subsets of the samples. L. Breiman, “Pasting small votes for classification in large databases and on-line”, Machine Learning, 36(1), 85-103, 1999.
Bagging	When random samples of the dataset are drawn with replacement L. Breiman, “Bagging predictors”, Machine Learning, 24(2), 123-140, 1996
Random Subspaces	When random subsets of the dataset are drawn as random subsets of the features •T. Ho, “The random subspace method for constructing decision forests”, Pattern Analysis and Machine Intelligence, 20(8), 1998.
Random Patches	When base estimators are built on subsets of both samples and features G. Louppe and P. Geurts, “Ensembles on Random Patches”, Machine Learning and Knowledge Discovery in Databases 2012
Random Forests	A hybrid of the Bagging and the Random Subspace Method Uses Decision Trees as the base classier with random splits L. Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.

Random Forests

- The scikit-learn implements by averaging instead of voting.
- The split that is picked is the best split among a **random subset of the features**.

Extremely Randomized Trees

- Randomness goes one step further in the way splits are computed.
- As in random forests, a random subset of candidate features is used.
- **Thresholds are drawn at random** for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule.

Main parameter | number of trees in the forest. → **"n_estimators"**

Feature importance | 1. Top of the tree. 2. Used in many trees → **Feature Selection**

Stacking

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
First-level learning algorithms $\mathcal{L}_1, \cdots, \mathcal{L}_T$;
Second-level learning algorithm $\mathcal{L}$.

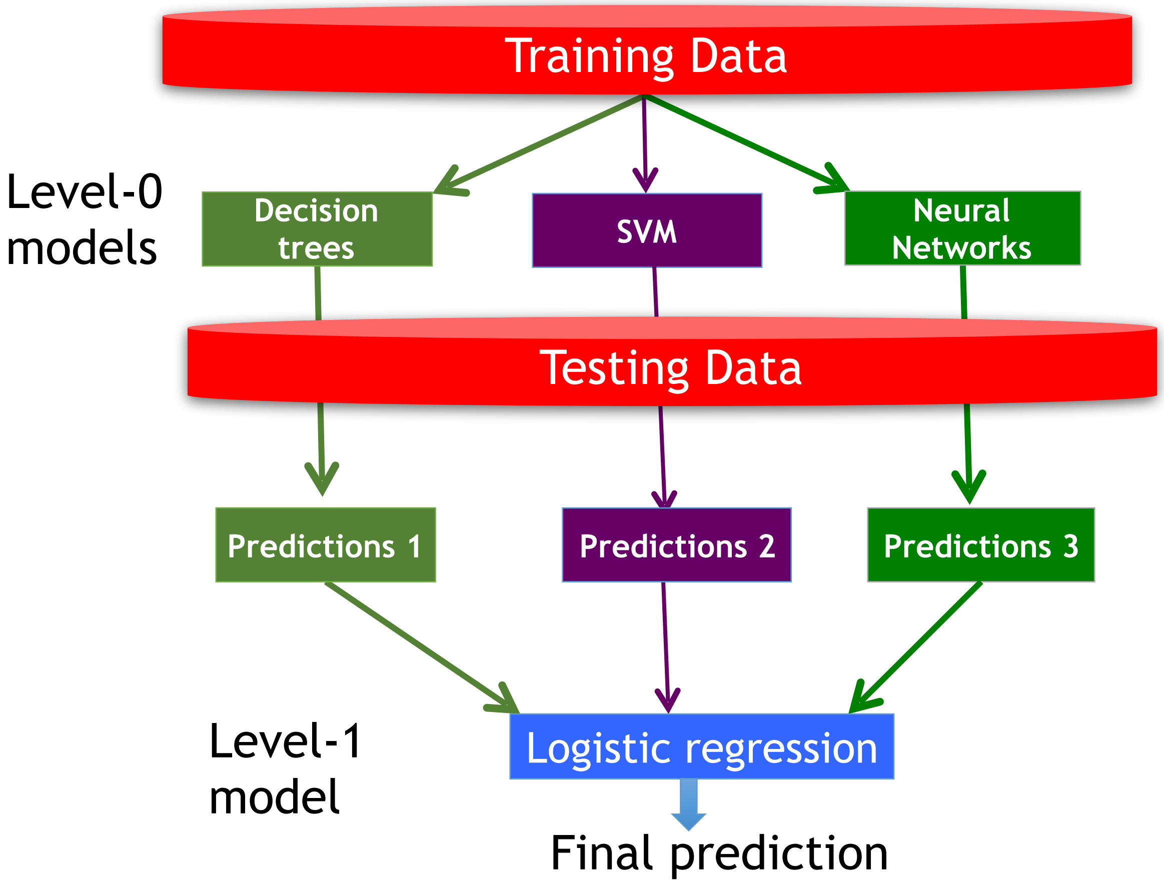
Process:

for $t = 1, \cdots, T$:
    $h_t = \mathcal{L}_t(\mathcal{D})$ % Train a first-level individual learner h_t by applying the first-level
end; % learning algorithm $\mathcal{L}_t$ to the original data set $\mathcal{D}$
$\mathcal{D}' = \emptyset$; % Generate a new data set
for $i = 1, \cdots, m$:
    for $t = 1, \cdots, T$:
        $z_{it} = h_t(\boldsymbol{x}_i)$ % Use h_t to classify the training example $\boldsymbol{x}_i$
    end;
    $\mathcal{D}' = \mathcal{D}' \cup \{((z_{i1}, z_{i2}, \cdots, z_{iT}), y_i)\}$
end;
$h' = \mathcal{L}(\mathcal{D}')$. % Train the second-level learner h' by applying the second-level
% learning algorithm $\mathcal{L}$ to the new data set $\mathcal{D}'$

Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \cdots, h_T(\boldsymbol{x}))$

Stacking

Stacked generalization (or stacking) is used to combine models of different types.



Part 3:
Boosting Algorithms

AdaBoost

- It Creates a ‘weak’ classifier that its accuracy is only slightly better than random guessing.
- A succession of models are built iteratively.
- **Records that were misclassified by the previous model are given more weight.**
- Finally, all of the successive models are weighted according to their success.
- Uses decision stumps as the base learners

Schapire, R.E.: The strength of weak learnability. Machine Learning 5(2) (1990)

AdaBoost

Algorithm Adaboost

Input: Required ensemble size T

Input: Training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where $y_i \in \{-1, +1\}$

Define a uniform distribution $D_1(i)$ over elements of S.

for $t = 1$ to T **do**

Train a model h_t using distribution D_t.

Calculate $\epsilon_t = P_{D_t}(h_t(x) \neq y)$

If $\epsilon_t \geq 0.5$ break

Set $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$

Update $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

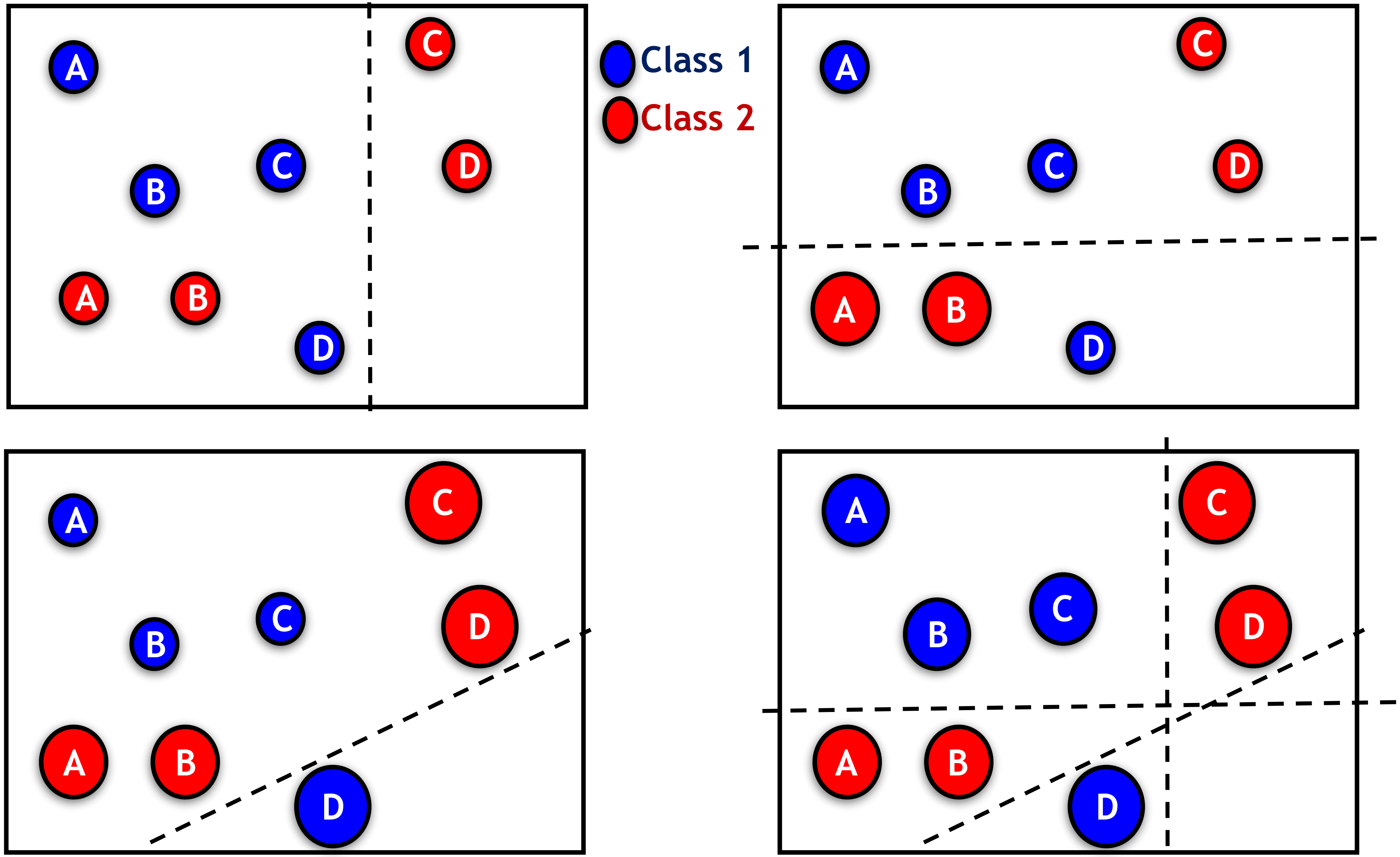
where Z_t is a normalization factor so that D_{t+1} is a valid distribution.

end for

For a new testing point (x', y'),

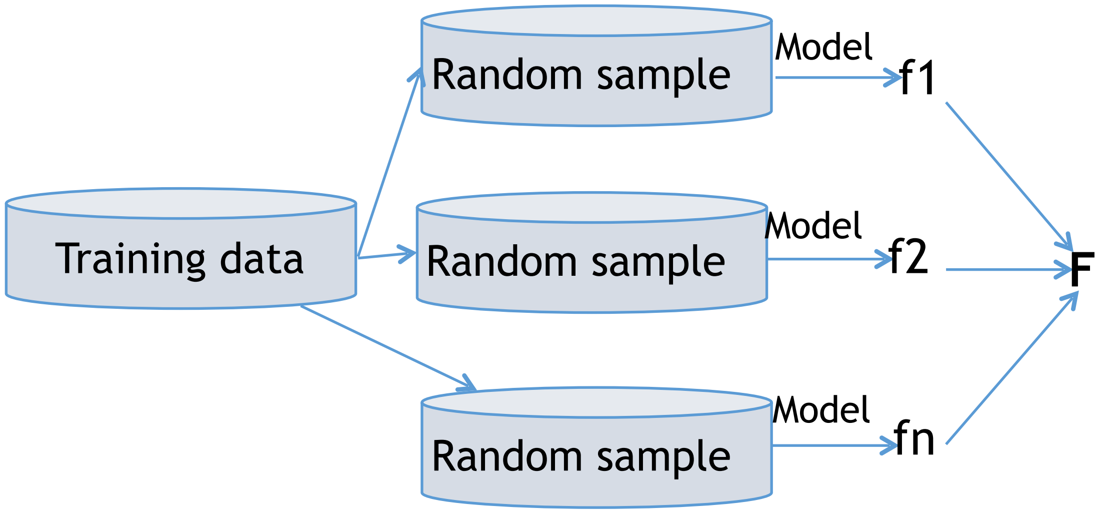
$H(x') = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x') \right)$

Boosting illustration



Bagging VS Boosting

Bagging

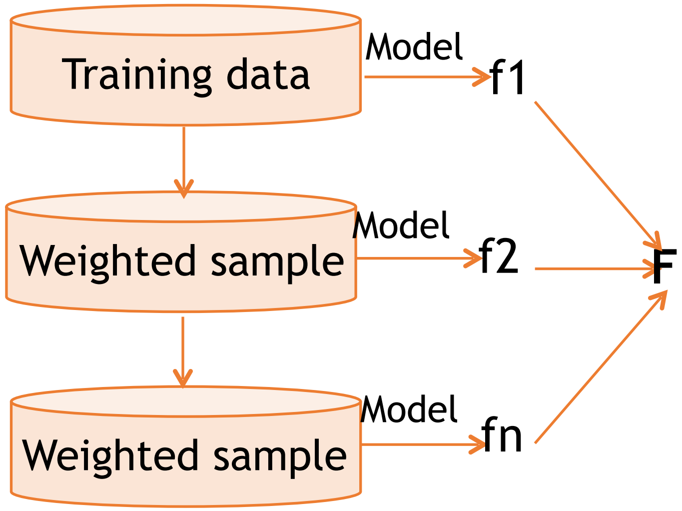


Resampling

Uniform distribution

Parallel Style

Boosting



Reweighting

Non-uniform distribution

Sequential Style

Gradient Boosted Models (GBM's)

- In gradient boosting, it trains many model sequentially. Each new model gradually minimizes the loss function using Gradient Descent method.

- The learning procedure consecutively fit new models to provide a more accurate estima of the response variable.

- *In Adaboost, the weights are derived from the misclassifications of the previous mod the resulting increased weights assigned to misclassifications.*

- The result of Gradient Boosting is an altogether different function from the beginning, because the result is the addition of multiple functions.

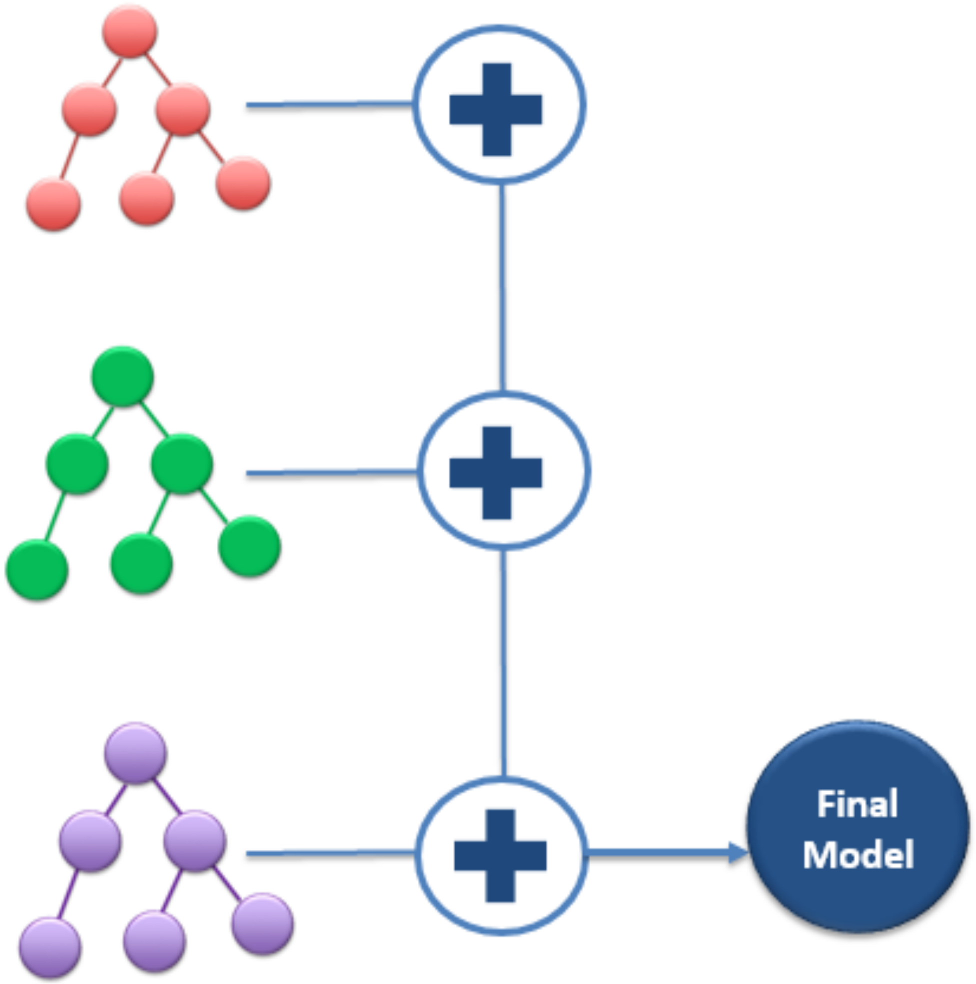
https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html

Gradient Boosting Trees

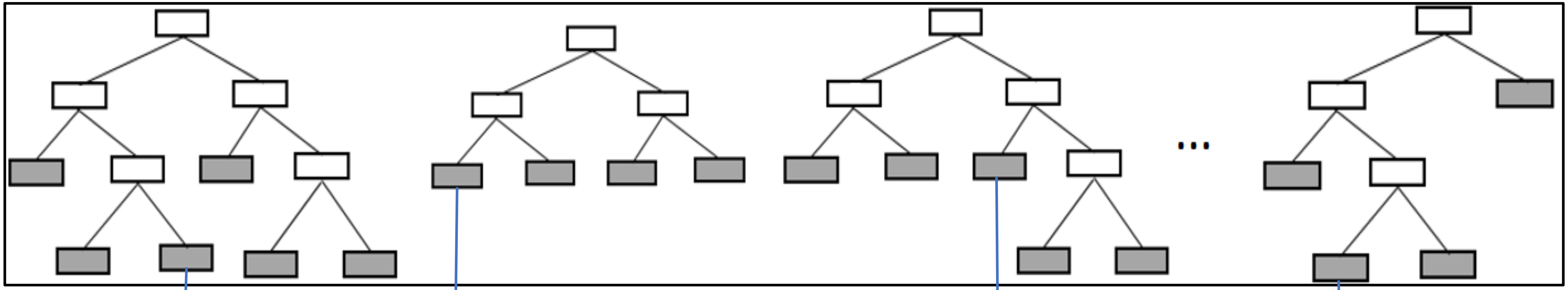
At each iteration:

- Draws a subsample of the training data (**without replacement**)
- Constructs a regression tree from the sample
- All trees are added together to get the final model

Jerome H. Friedman, "Stochastic gradient boosting", Computational Statistics & Data Analysis 2002



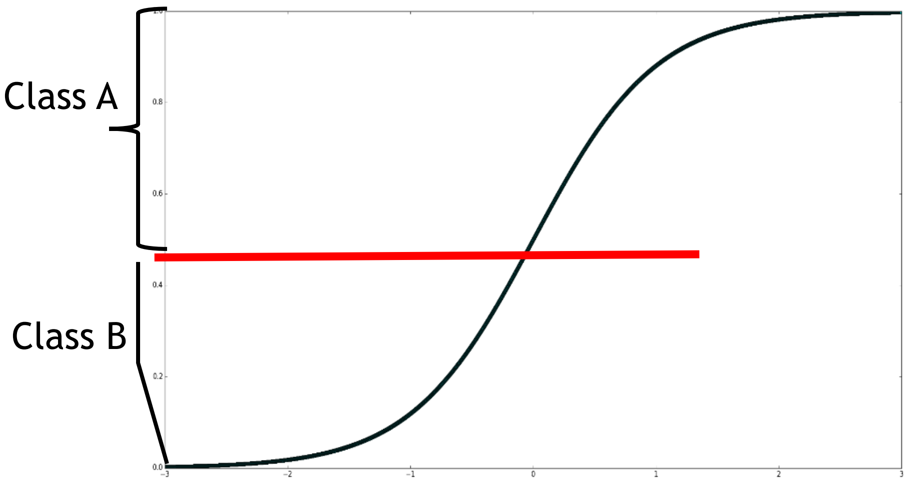
Prediction with GBM



GBM_Score= response1 +response2 +response3 +... +responseM

$$Pred = \frac{e^{GBM\_score}}{e^{GBM\_Score} + e^{-GBM\_Score}}$$

monotonic function



AdaBoost & GBM in SKlearn

`sklearn.ensemble`.**AdaBoostClassifier**

Input parameters:

base_estimator
n_estimators
Learning_rate
…

`sklearn.ensemble`.**GradientBoostingClassifier**

Input parameters:

loss
n_estimators
learning_rate
n_estimator
max_depth
Mx_features
…

XGBoost

- XGBoost is an implementation of GBM, with major improvements.

- GBM's build trees sequentially, but XGBoost is parallelized. This makes XGBoost faster.

- XGBoost is an open-sourced machine learning library available in Python, R, Julia, Java, C++, Scala.

XGBoost: A Scalable Tree Boosting System, Tianqi Chen, Carlos Guestrin. 2016

XGBoost features

1. **Split finding algorithms: approximate algorithm**
    - Candidate split points are proposed based on the percentiles of feature distribution.
    - The continuous features are binned into buckets that are split based on the candidate split points.
    - The best solution for candidate split points is chosen from the aggregated statistics on the buckets.
2. **Column block for parallel learning**
    - To reduce sorting costs, data is stored in in-memory units called ‘blocks’.
    - Each block has data columns sorted by the corresponding feature value.
    - This computation needs to be done only once before training and can be reused later.
    - Sorting of blocks can be done independently and divided between parallel threads.
    - The split finding can be parallelized as the collection of statistics for each column is done in parallel.

https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html

XGBoost features ..cont.

3. Sparsity-aware algorithm:

- XGBoost visits only the default direction (non-missing entries) in each node.

4. Cache-aware access:

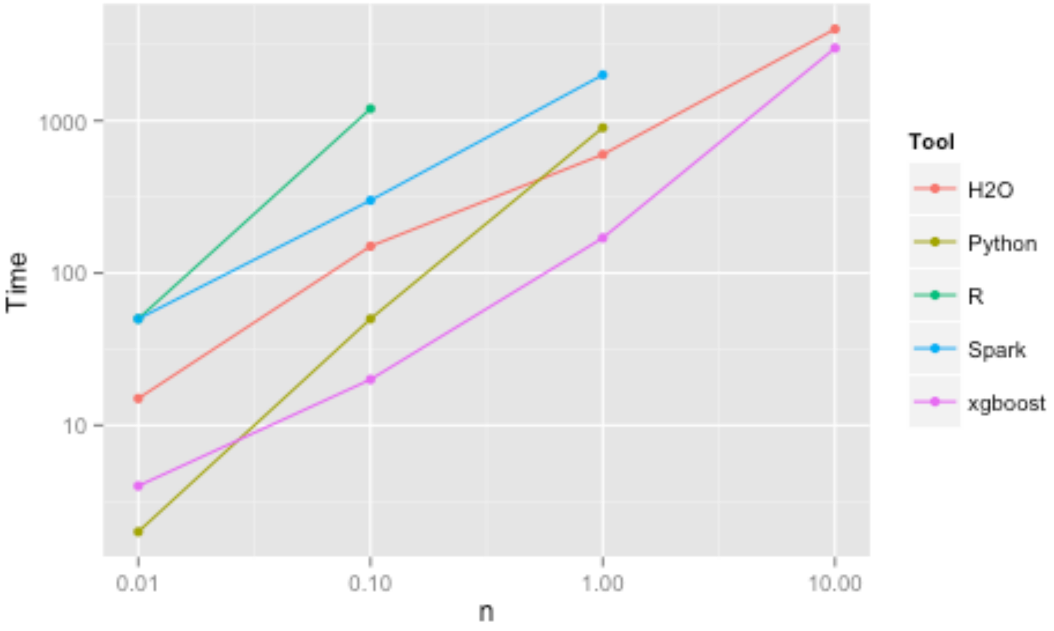
- Optimizes how many examples per block.

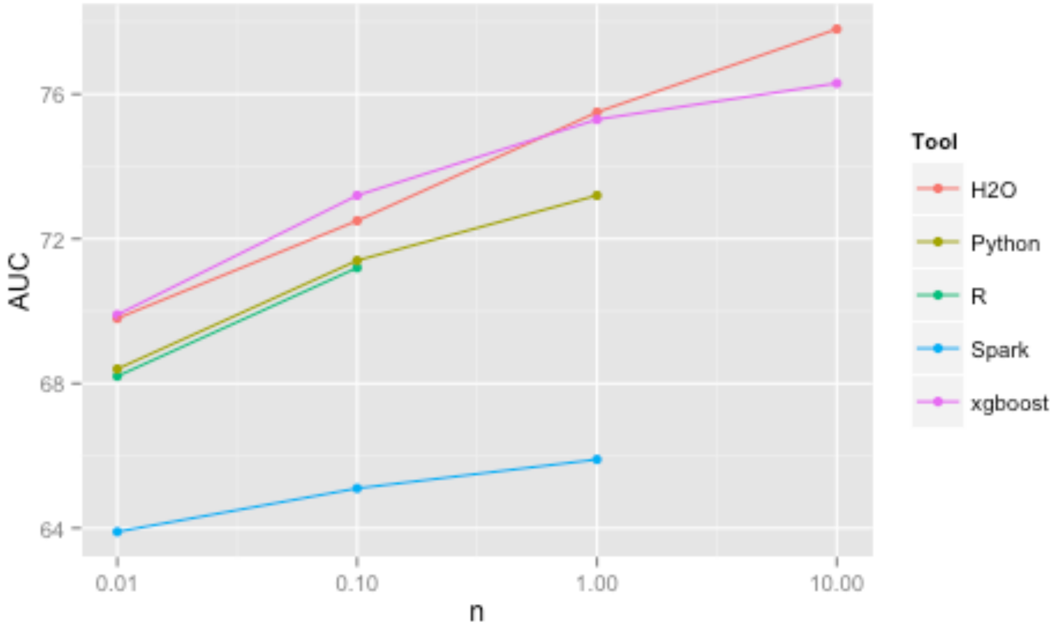
5. Out-of-core computation:

- For blocks that does not fit into memory, they are compressed on the disk .
- The blocks are decompressed on the fly (In parallel).

https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html

Xgboost VS Other tools





http://datascience.la/benchmarking-random-forest-implementations/

Light GBM…the latest of Boosting algorithms

- A fast, distributed, high performance gradient boosting framework.
- Used for ranking, classification and many other machine learning tasks.
- It is under the umbrella of the DMTK(http://github.com/microsoft/dmtk) project of Microsoft.
- Similar to XGBoost but it is faster

https://github.com/Microsoft/LightGBM/blob/master/docs/Experiments.rst#comparison-experiment

Advantages Ensemble Learning

Accuracy

less
Variance

less
Overfitting

Diversity

In Ensemble learning, the large variance of unstable learners is `averaged out' across multiple learners.

Different classifiers to work on different random subsets of the full feature space or different subset of the training data.

Imagine we have:

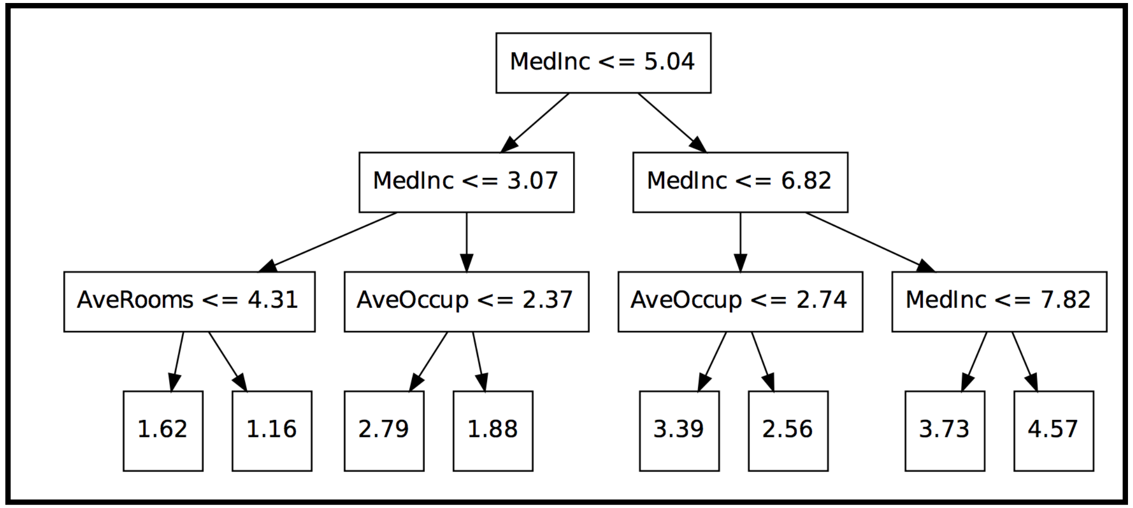
- An ensemble of 5 independent classifiers.
- Accuracy is 70% for each

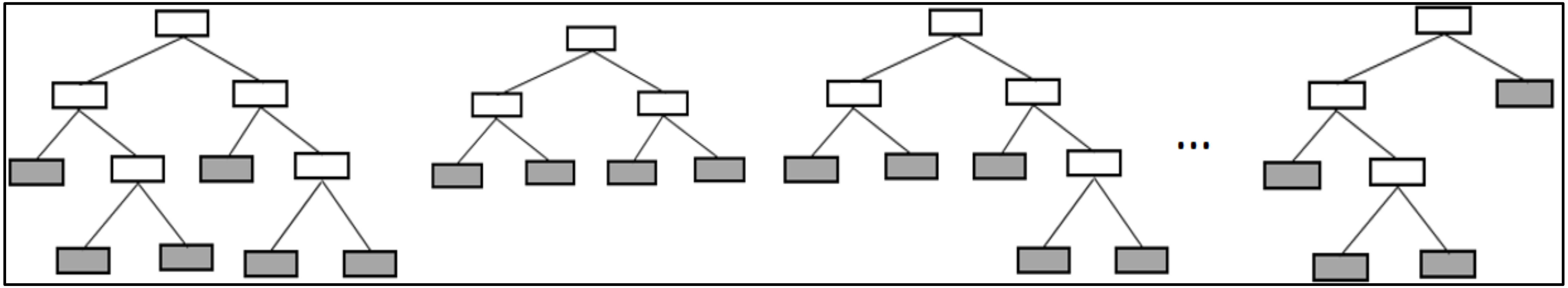
What is the accuracy for the majority vote?

10 (.7^3)(.3^2)+5(.7^4)(.3)+(.7^5)
83.7% majority vote accuracy

How about if we have101 such classifiers
99.9% majority vote accuracy

Are ensembles easy to understand?





Are ensembles easy to understand?

- Decision trees are easy to understand
- Ensemble models are considered complex model, interpreting predictions from them is a challenge.

LIME - Local Interpretable Model-Agnostic Explanations
Approximate the complex model near a given prediction. - Ribeiro et al. 2016

SHAP (SHapley Additive exPlanations)
A unified approach to interpreting model predictions, Scott Lundberg, Su-In Lee 2017

DEMO

References

- Zhou, Zhi-Hua, “Ensemble Learning”, Encyclopedia of Biometrics, 2009
- *Dietterich, T. Ensemble methods in machine learning. Multiple classifier systems, Vol. 1857 of Lecture Notes in Computer Science. 2000*
- Ho, Tin Kam, The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.
- VALENTINI, Giorgio, and Francesco MASULLI, 2002. Ensembles of Learning Machines. 2002. Revised Papers, Volume 2486 of Lecture Notes in Computer Science. Berlin: Springer, pp. 3–19.
- FREUND, Yoav, and Robert E. SCHAPIRE, 1996. Experiments with a New Boosting Algorithm. In: Lorenza SAITTA, ed. (ICML ’96). San Francisco, CA: Morgan Kaufmann, pp. 148–156.
- SCHAPIRE, Robert E., 1990. The Strength of Weak Learnability. Machine Learning, 5(2), 197–227. Oxford University Press.
- BREIMAN, Leo, 1996. Bagging Predictors. Machine Learning, 24(2), 123–140.Sewell M (2011) Ensemble learning, Technical Report RN/11/02. Department of Computer Science, UCL, London
- G Brown, “Ensemble Learning”, Encyclopedia of Machine Learning, 2010
- http://scikit-learn.org/0.16/modules/ensemble.html
- http://scikit-learn.org/stable/modules/multiclass.html
- http://xgboost.readthedocs.io/en/latest/

KDD Association for Computing Machinery
bringing together the data mining, data science and analytics community

Austin ACM SIGKDD - Austin's Big Data Machine Learning Group

Boosting Algorithms

Omar Odibat, Data Scientist, VISA

Thank you!!!