Week 6

Machine Learning and Big Data - DATA622

CUNY School of Professional Studies

Decision Trees

A simple but powerful approach

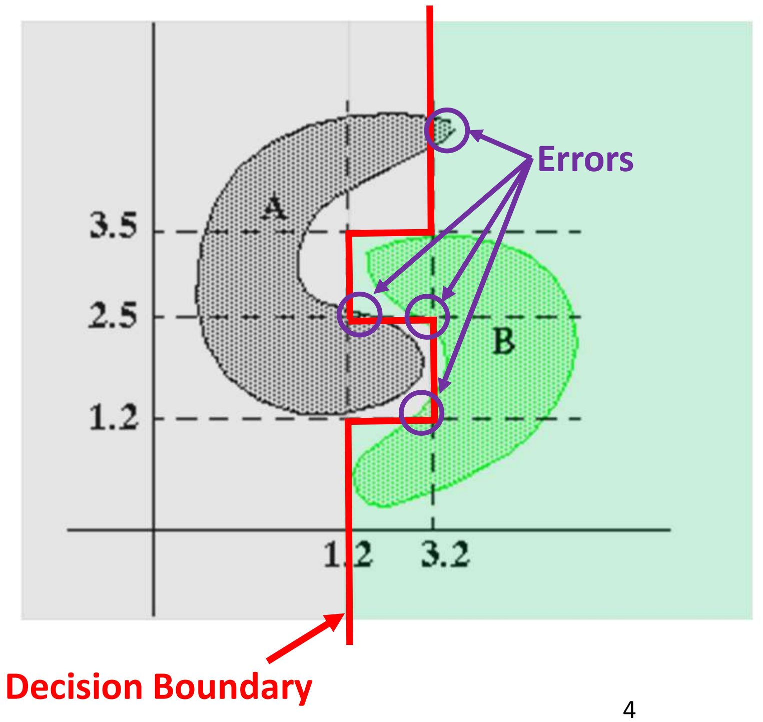
CUNY School of Professional Studies

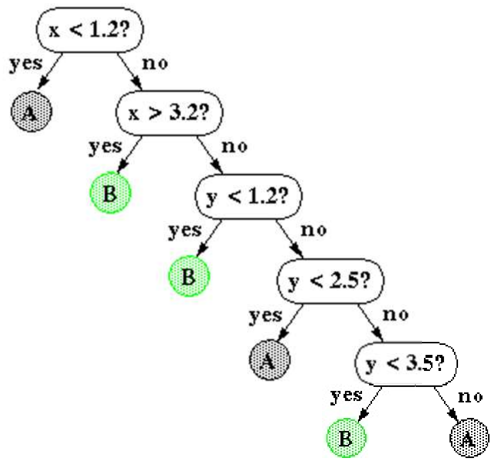
Decision Trees

- Simple, yet widely used classification technique
  - For nominal target variables
  - There also are regression trees, for continuous target variables
- Predictor Features: binary, nominal, ordinal, discrete, continuous
- Evaluating the model:
  - One metric: error rate in predictions
- Decision trees attempt to classify a pattern through a sequence of questions
- Decision trees are very powerful and can give excellent performance

Nonlinear Decision Surfaces

- Decision trees can produce nonlinear decision surfaces
- Can give arbitrarily high levels of precision on the training data





CUNY School of Professional Studies

Decision Tree Classifier

- Decision tree models are relatively more descriptive than other types of classifier models
  - Easier interpretation
  - Easier to explain predicted values
  - Data-driven learning
- Exponentially many decision trees can be built
  - Which is best?
    Some trees will be more accurate than others
  - How to construct the tree?
    Computationally infeasible to try every possible tree.

Decision Tree Nodes

- A decision tree has three types of nodes:
  1. A **root** node that has no incoming edges and zero or mode outgoing edges
  2. **Internal nodes**, each of which has exactly one incoming edge and two or more outgoing edges
  3. **Leaf nodes** (or **terminal nodes**), each of which has exactly one incoming edge and no outgoing edges
- Each leaf node is assigned a class label
- Non-terminal nodes contain attribute test conditions to separate records that have different characteristics

Building a Decision Tree

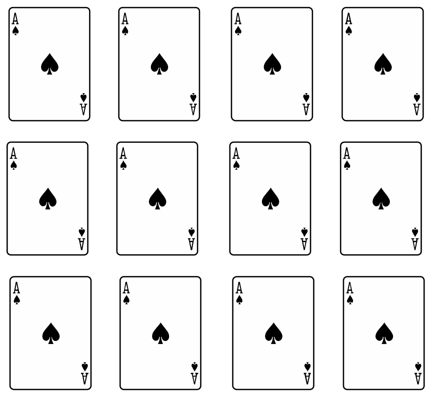
- Referred to as **decision tree induction**.
- Exponentially many decision trees can be constructed from a given set of attributes
  - Infeasible to try them all to find the optimal tree
- Different “decision tree building” algorithms
- Usually a **greedy strategy** is employed

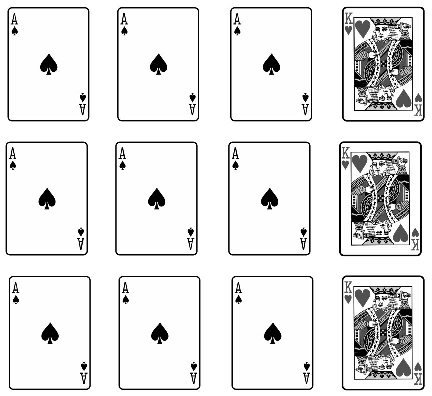
Decision Tree Induction

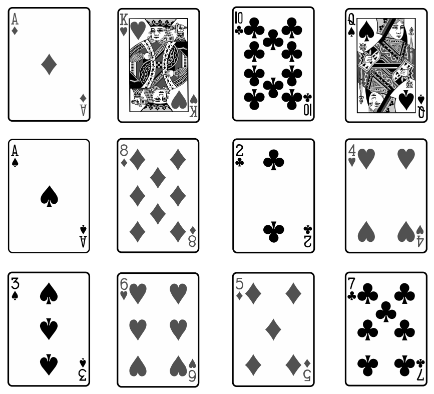
- How should the training records be split?
  - ○ **Greedy strategy**: split the records based on some attribute test (always choose immediate best option)
  - Need to evaluate the "goodness" of each attribute test and select the best one.
- How should the splitting procedure stop?
  - ○ For now, we'll keep splitting until we can't split anymore.

Entropy

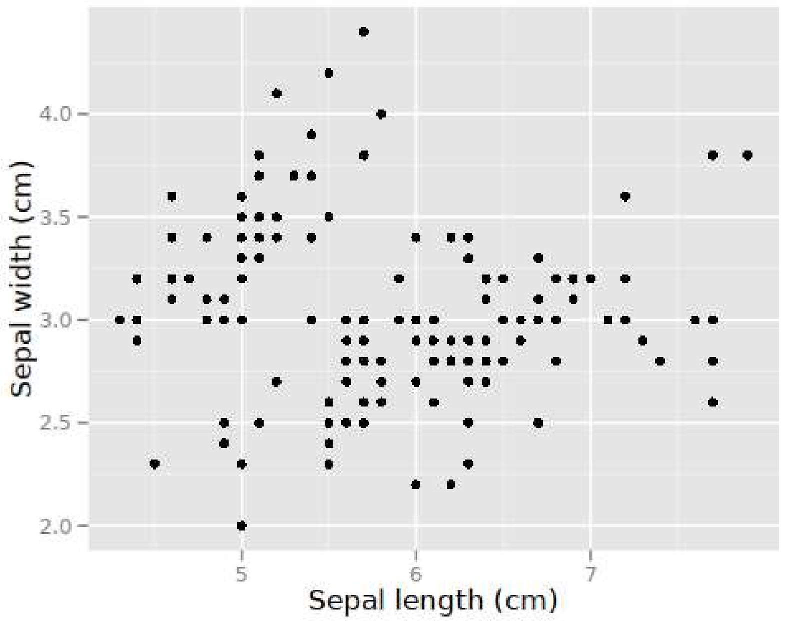
- Defined by mathematician Claude Shannon
- Measures the impurity (**heterogeneity**) of the elements of a set
- "uncertainty of guessing the result of the random selection from a set?"


Very pure. Not impure.

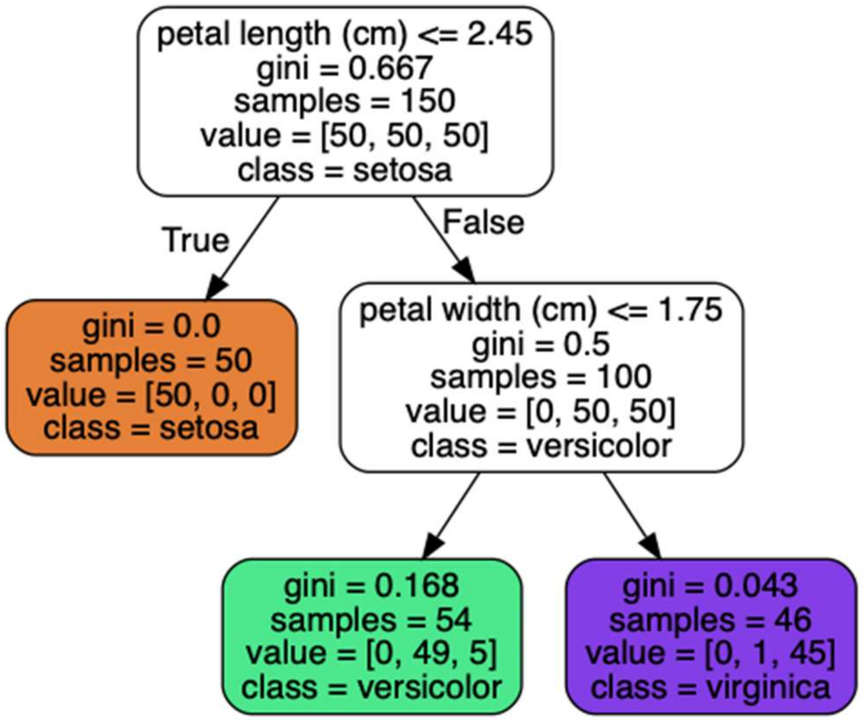

Little impure.


Completely impure.

Iris dataset



Iris dataset



CUNY School of Professional Studies

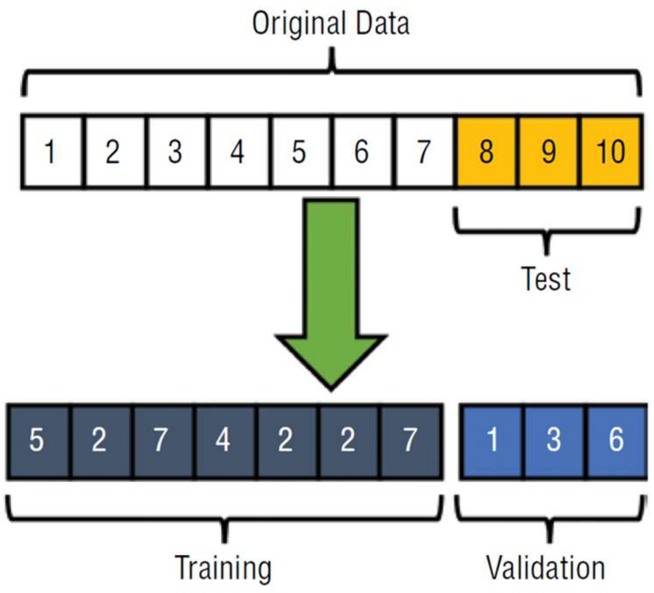
Bootstrapping

Bootstrap Sampling

CUNY School of Professional Studies

Bootstrapping

- Refers to **Bootstrap Sampling**
- Create a new training dataset from the original data
- Randomly sample data, **with replacement**



Bootstrapping

- The probability of picking one row out of n is *1/n*.
- Therefore the probability of not picking it is *1-(1/n)*
- After n trials the probability of not picking it is *(1-(1/n))n*
- As n approaches infinity *(1-(1/n))n* becomes e^{-1}*=0.368*.
- Hence, approximately 63.2% of datapoints are uniquely selected in a bootstrap

Bagging

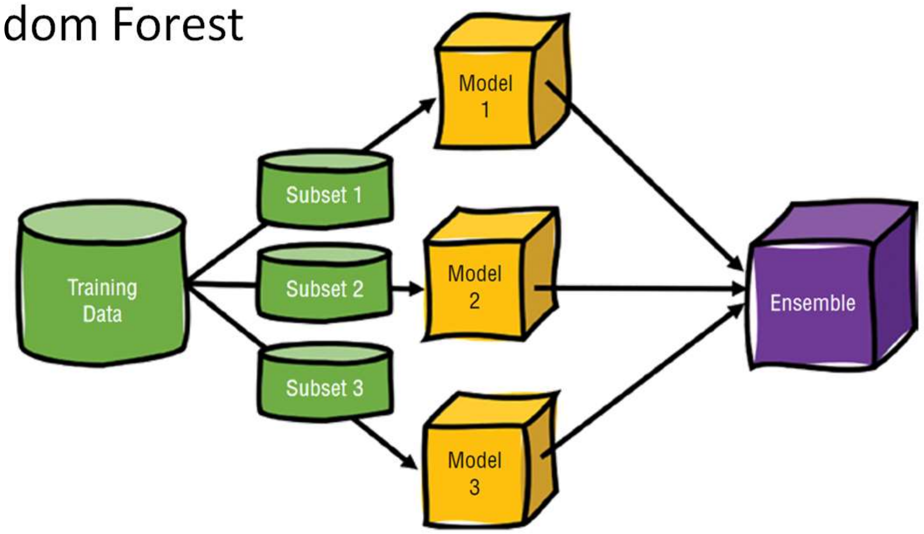
Bootstrap Aggregation

Bootstrap Aggregation (Bagging)

- Ensemble Learning Method
- Combining the results of multiple learners (for instance, all decision trees) to get a generalized result
- Multiple learners of the same algorithm are trained with bootstrapped data
- Uses Bootstrapping. Sample Size: same as training set
- Action: repeatedly sample with replacement according to uniform probability distribution
    - Every instance has equal chance of being picked
    - Some instances may be picked multiple times; others may not be chosen

Bagging

- Boosting works by iteratively generating learners and adding them to the ensemble
- Iteration stops when a predefined number of learners have been added
- In Bagging the result is obtained by averaging the responses of the N learners (or majority vote)
- Example: Random Forest



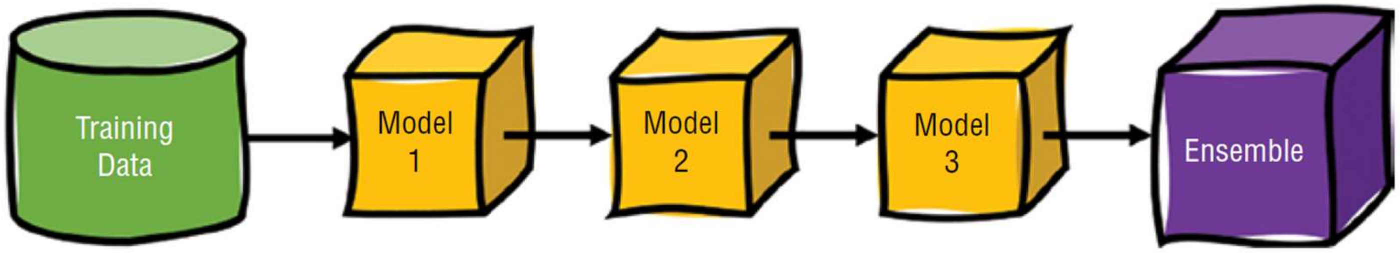
Boosting

Co-dependent models

CUNY School of Professional Studies

Boosting

- Boosting works by iteratively creating models and adding them to the ensemble
- Iteration stops when a predefined number of models have been added
- Each new model added to the ensemble is biased to pay more attention to instances that previous models misclassified (weighted dataset).



Boosting

- Sequential algorithm where at each step, a weak learner is trained based on the results of the previous learner (model)
- Each subsequent learner attempts to correct the errors of the previous learner
- Succeeding models are dependent on the previous model
    - Models are not randomly chosen but are mainly influenced by wrong classified entries of the previous model
- Two main types:
    - Adaptive Boosting: Re-weight datapoints based on performance of last weak learner. Focuses on points where previous learner had trouble. Example: AdaBoost.
    - Gradient Boosting: Train new learner on residuals of overall model. Constitutes gradient boosting because approximating the residual and adding to the previous result is essentially a form of gradient descent. Example: XGBoost.

CUNY School of Professional Studies

Boosting vs Bagging

Comparing approaches

Boosting

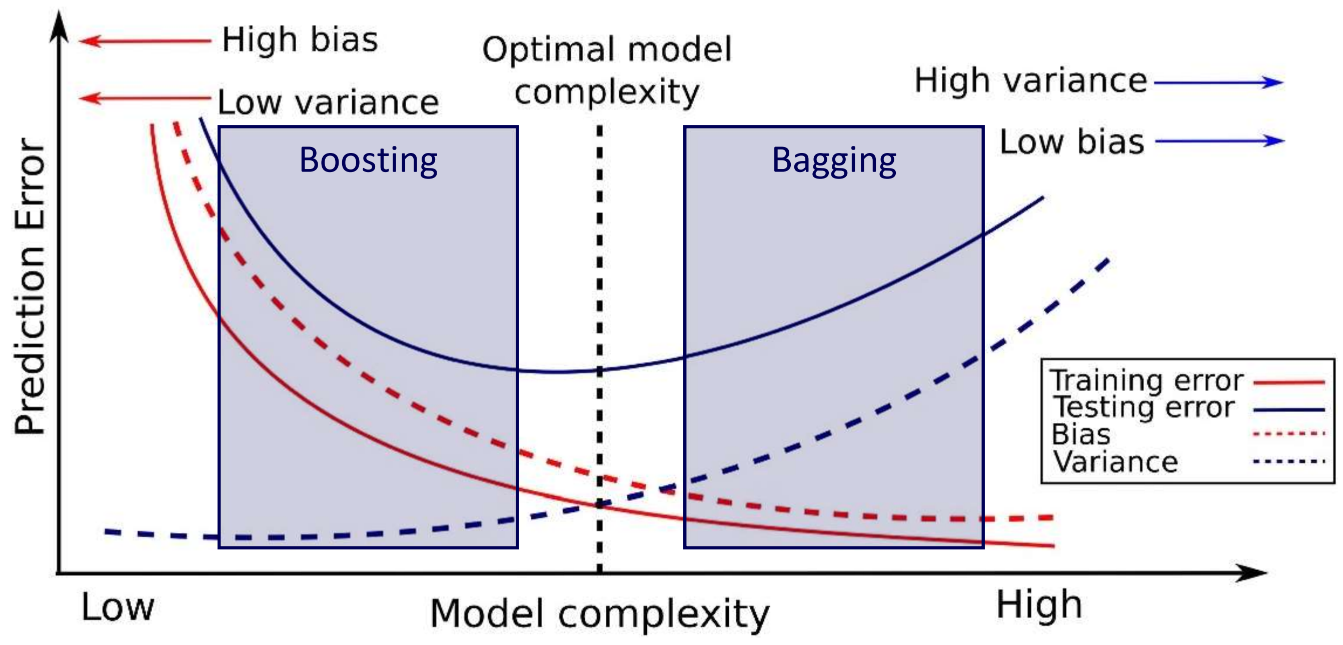
- Both Ensemble Learning methods
- Bagging helps to decrease the model's **variance**
  - Bagging helps in reducing the overfitting: we are averaging all the models outputs using the majority voting approach
- Boosting helps to decrease the model's **bias**
  - Each model tries to reduce the errors of the previous model in the chain
  - Can use multiple loss functions to reduce the error in each sequential models
  - Boosting method may overfit as the models build sequentially, all the models try to reduce the training error
- Regression:
  - Bagging: For regression models the predicted value won't be optimized if any one of the models is deviating more as the output value will be average of all the models.

Comparison

- Different data partition
  - Bagging uses random samples of the training data for all models independently, boosting puts higher importance on misclassified data of the upcoming models. Therefore, the data partition is different here.
- Independent vs chain models | independent vs. sequences
  - Bagging creates independent models that are aggregated together. However, boosting updates the existing model with the new ones in a sequence.
- Variance vs. bias
  - Bagging aims to reduce the variance, but boosting tries to reduce the bias. Therefore, bagging can help to decrease overfitting, and boosting can reduce underfitting.
- Function | weighted vs. non-weighted
  - For bagging the function to predict the outcome uses equally weighted average or equally weighted voting aggregations. Boosting uses weighted majority vote or weighted average functions with more weight to those with better performance on training data.

CUNY School of Professional Studies

Comparison



CUNY School of Professional Studies

Impurity Functions

Gini Impurity

- Let us say that we are at a branch that splits the data into two subsets $\mathcal{D}_L$ and $\mathcal{D}_R$
- The Gini impurity of the tree will be:

$$G^T(\mathcal{D}) = \frac{|\mathcal{D}_L|}{|\mathcal{D}|} G^T(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} G^T(\mathcal{D}_R)$$

- NOTE: there are other measures of impurity for classification (for instance, using entropy