Trees

–Mannu (_mannuscript)

What are Tree Based Algorithms?

- Set of:
  - Non-linear,
  - Supervised algorithms
  - For classification and regression.
- Examples:
  - Decision trees,
  - Random forests,
  - Gradient boosting.

Why Trees?

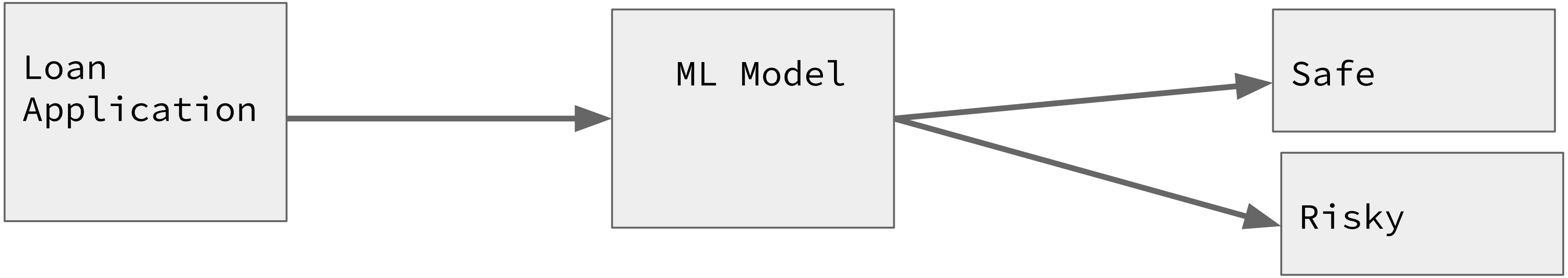
Why Trees?

- Easy to understand.
- Very less fancy statistics required.
- Not data hungry.
- Not affected by outliers.

Decision Trees

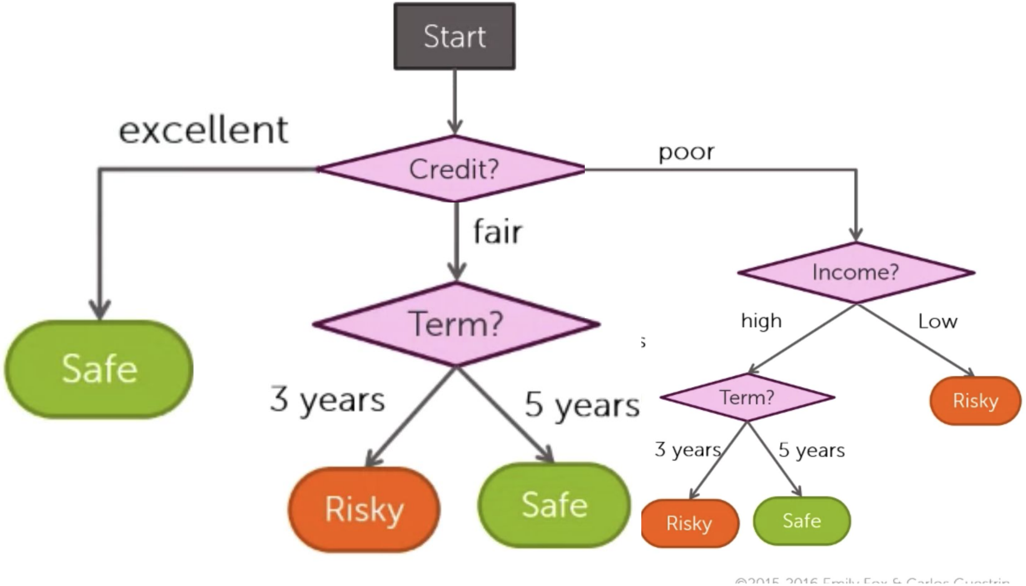
Problem

- Predict if a given loan application is safe or risky.
- Given the data:
  - Credit History
  - Income
  - Term
  - Personal Information



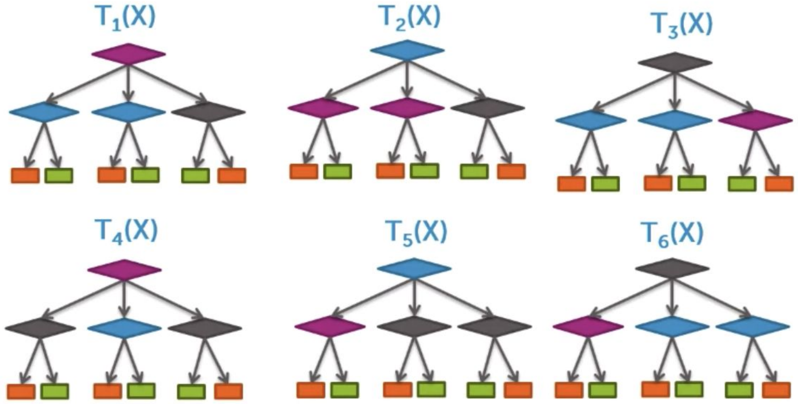
Problem

- Credit History
- Income
- Term
- Personal Information



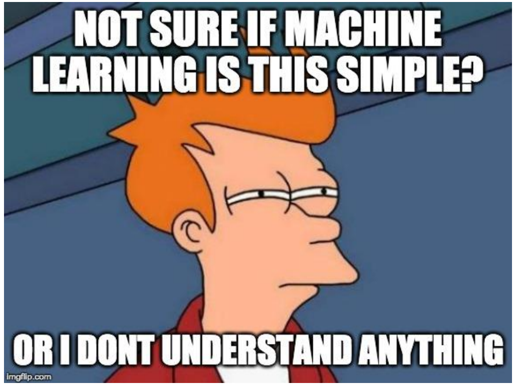
Decision Trees

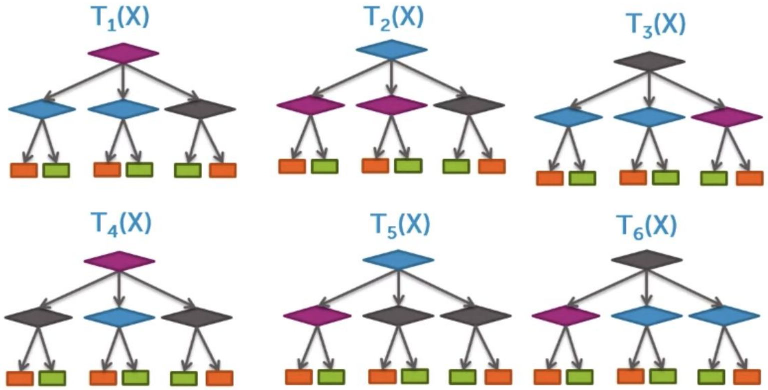
- Objective:
  - Given a dataset, create a Tree T(x) with minimum classification error on the training dataset.
  - Classification error = # of incorrect prediction / # of samples.
- But wait!
  - How to decide the feature/node placements?



Decision Trees

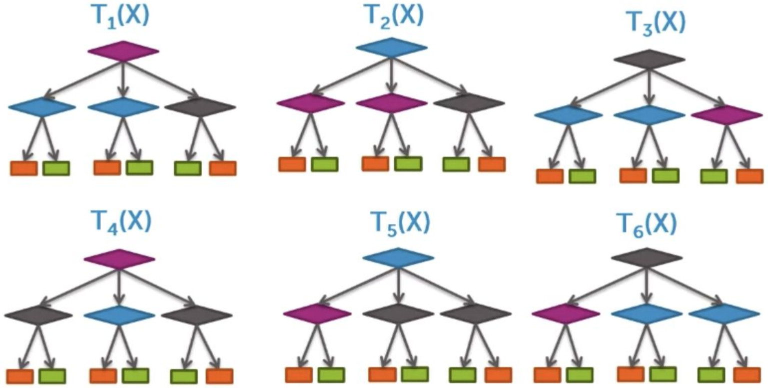
- Objective: find t from T to min(E).
  - T : Set of all possible decision tree for give features.
  - E : Classification error on training data.
  - t : Best tree.
- Easy: Just calculate E for all t in T!





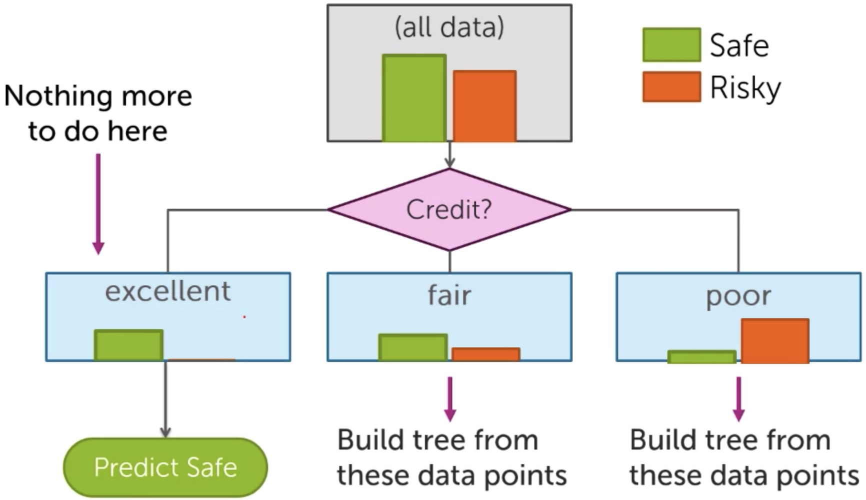
Decision Trees

- There is a problem:
  - NP-hard problem!!!
  - # of trees grows exponentially.
  - No approximate solution.
- Need some heuristic to build the tree:
  - Greedy algorithm.



Decision Trees: Greedy Algorithm

1. Start with an empty tree.
2. Select a feature to split.
3. For each split:
   a. If nothing more, stop the recursion.
   b. Otherwise, recurse, go back to step 2 and continue to split.



Decision Trees: Greedy Algorithm

1. Start with an empty tree.
2. Select a feature to split.
3. For each split:
    a. If nothing more, stop the recursion.
    b. Otherwise, recurse, go back to step 2 and continue to split.

- Two challenges:
    - How to decide which feature to split at step 2?
    - What are the stopping conditions at step 3.a?

Decision Trees

- What are the stopping conditions?
    - Nothing to split.
    - No more features left.
    - Constraint on max depth of the tree.

Decision Trees

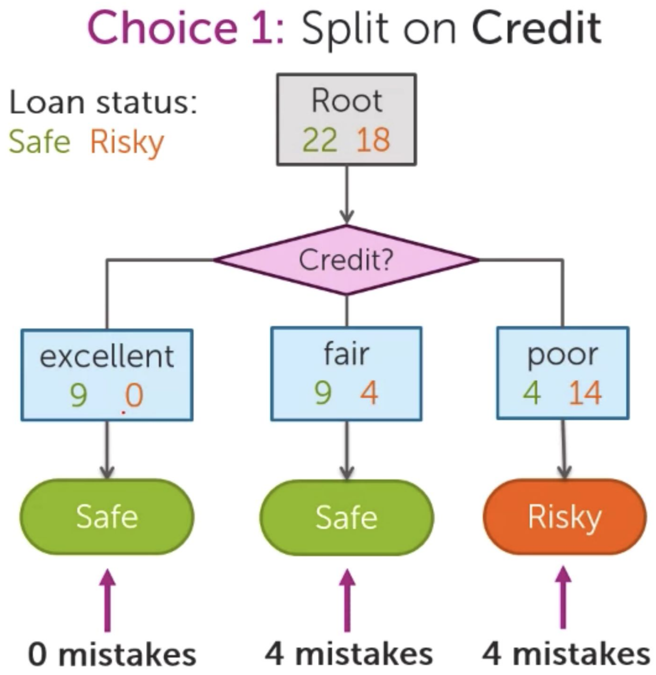
- How to decide which feature to split first?
  - Gini Index
  - Information Gain
  - Classification Error

Decision Trees: Feature Split

- Classification Error:
  a. For each (remaining) feature:
     - Split on the values
     - For each split:
       - Assign majority class as the prediction class
       - Calculate the misclassified examples M_s
     - Calculate $E^f = (\sum_s M^f)/\text{Total examples}$
  b. Choose the feature with min E^f.

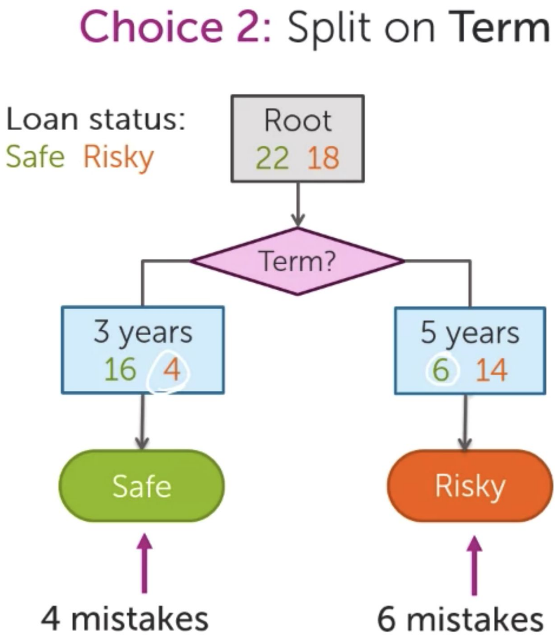
Decision Trees: Feature Split

- Classification Error:
  a. For each (remaining) feature: **(Choose Credit)**
     - Split on the values **(excellent, fair, poor)**
     - For each split:
       - Assign majority class as the prediction class. **(Safe, Safe, Risky)**
       - Calculate the misclassified examples M_s **(0, 4, 4)**
     - Calculate E^f **(0 + 4 + 4)/40**



Decision Trees: Feature Split

- Classification Error:
  a. For each (remaining) feature: **(Choose Term)**
     - Split on the values **(3y, 5y)**
     - For each split:
       - Assign majority class as the prediction class. **(Safe, Risky)**
       - Calculate the misclassified examples M_s **(4, 6)**
     - Calculate E^f = **(4 + 6)/40**



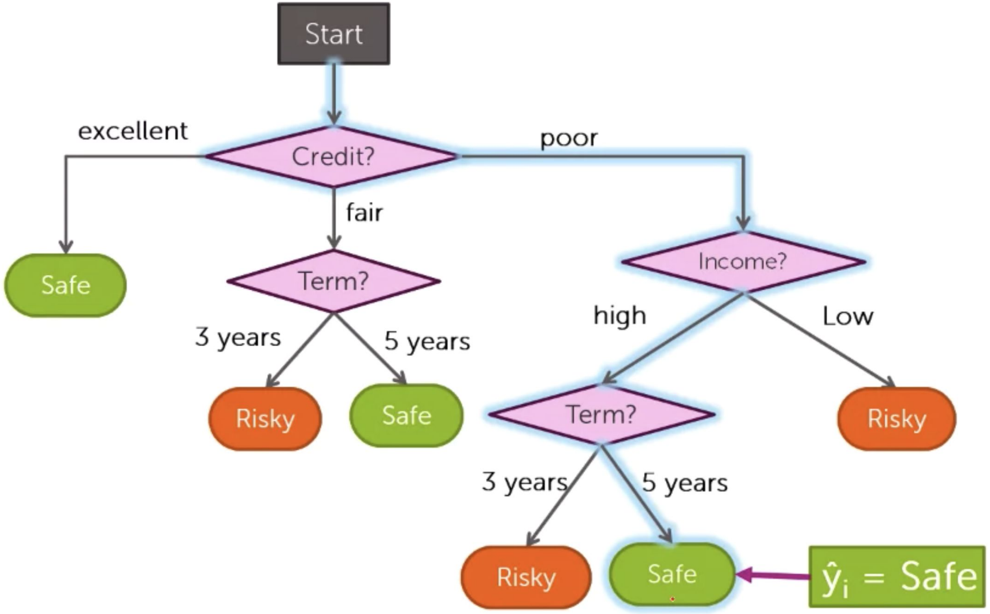
Decision Trees: Feature Split

- Classification error:
  - Credit: (0 + 4 + 4)/40 = .2
  - Term: (4 + 6)/40 = .25

1. Start with an empty tree.
2. Select a feature to split.
3. For each split:
   a. If nothing more, stop the recursion.
   b. Otherwise, recurse, go back to step 2 and continue to split.

Decision Trees: Predicting Values

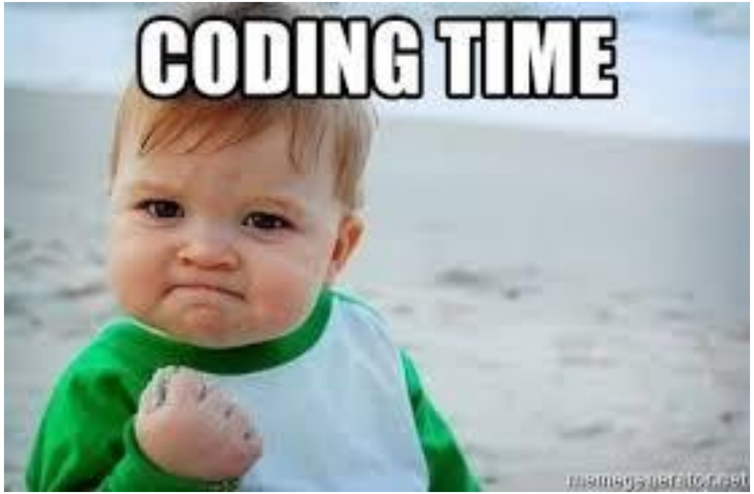
- Predicting = Traversing the tree
- E.g. (Credit = Poor, Term = 5y, Income = High)



Decision Trees: Continuous valued features

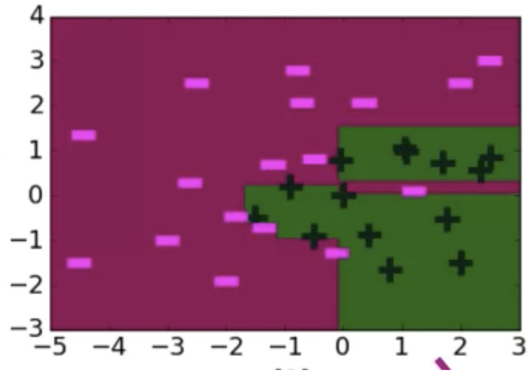
- Features are not only categorical.
- How to split now?
  a. Number of classes are infinite.
- Use threshold split.
- Algorithm:
  a. Sort the values
  b. For i = 1 to N-1:
     - $t_i = (N_i + N_{i+1})/2$
     - Compute classification error
- Choose the N_i with min(error) as the threshold split.

Decision Trees: Hands On



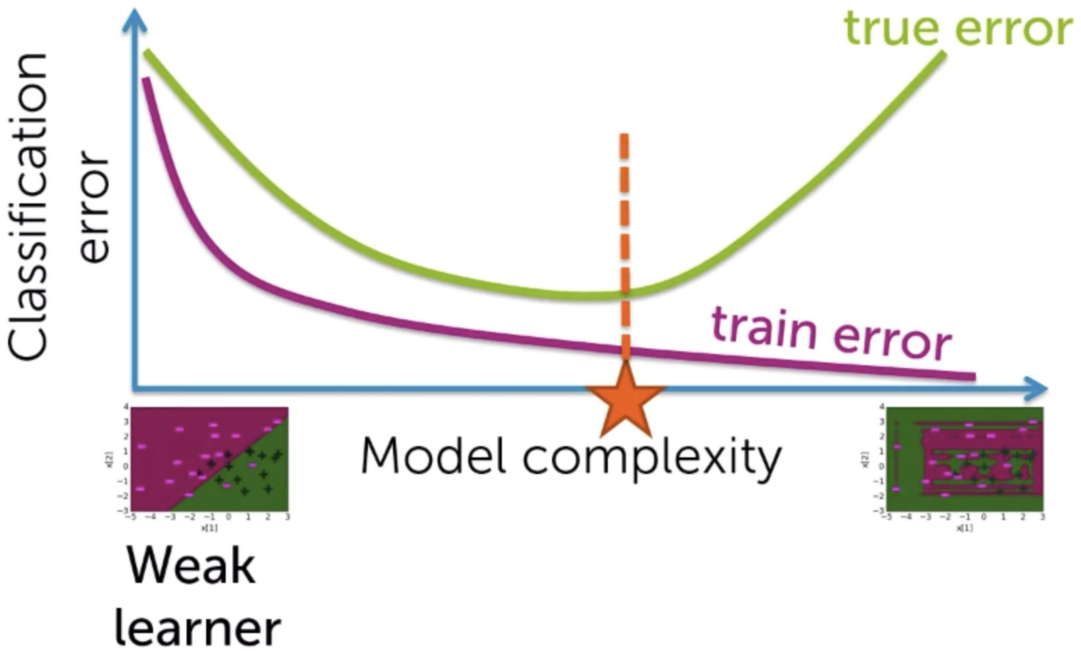
Overfitting in Decision Trees

- Early stopping
  - Limit the depth of the tree.
  - Not considering splits that do not reduce classification error
  - Do not split for less data.
- Pruning
  - Bottom up
  - Cost = Error(T) + lambda * #of leaves



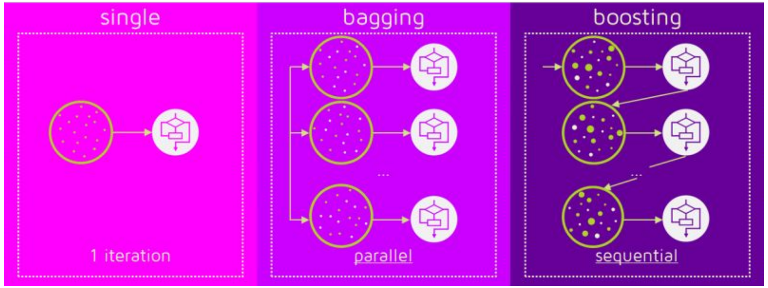
Boosting

- Week learner vs complex model.
- Need to find the sweet spot.
- Two ways:
    - Increase the complexity of the model. (Depth)
    - Boosting!



Boosting

- Combines(Boosts) multiple weak learners to increase performance.
- Keep committing mistakes and keep learning from them.
- Algorithm:
  - 1. Learn a weak classifier
  - 2. M = Find the misclassified examples
  - 3. Give higher weights to M
  - 4. Go back to 1



Boosting: Adaboost

- Ensembling: Pred = $w_1*h_1+w_2*h_2+w_3*h_3+...$
- AdaBoost algorithm:
  - Initialize weights α_i = 1/n
  - For t = 1 to T (where T is the number of week learners)
    - Learn h_t with data weights α_i
    - Compute coefficient w_t
    - Recompute weights α_i
  - Final model: pred = $w_1*h_1+w_2*h_2+w_3*h_3+...w_t*h_t$
- Challenges:
  - Finding w_t
  - Finding α_i for each step.

Boosting: Adaboost: Finding W_T

- Previous error = # of misclassified examples / # of examples.
- New error = weighted error =($\sum \alpha_i$ of misclassified examples)/($\sum \alpha_i$)
- But how to calculate w_t:
  - w_t is directly proportional to weighted error.
  - w_t = .5 * ln((1-weighted_error(f_t))/weighted_error(f_t))

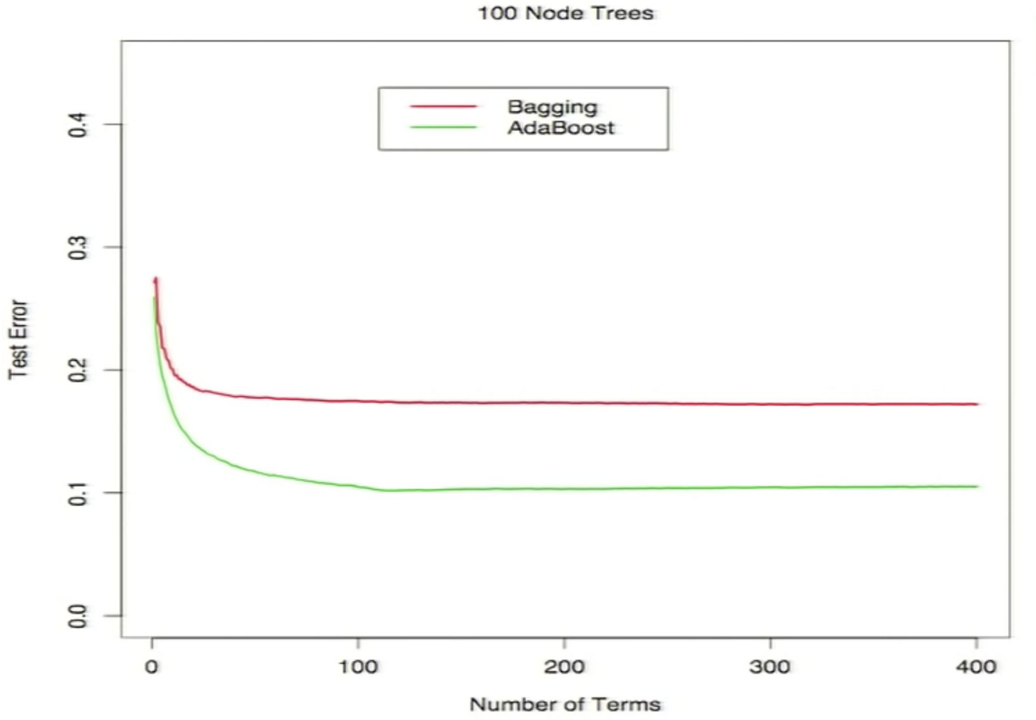
Boosting: Adaboost: Finding α_i

- New weights are inversely proportional to weight.
- $\alpha_i = \alpha_i \ e^{-w_t}$ (correct)
- $\alpha_i = \alpha_i \ e^{w_t}$ (incorrect)

Bagging: Random Forest

- Ensemble of decision trees
- Wisdom of crowd, poll unrelated models.
- For t = 1 to T (T is number of trees in the forest):
  - Select k features randomly.
  - Train h_t
- For prediction: majority vote!

Boosting vs Bagging:



Homework:

1. Programming:
    a. Extend the decision tree code for multiclass and continuous valued features.
    b. Extend it to implement random forest (and AdaBoost?).
2. Theory:
    a. Study how to do regression using decision trees (Teach us tomorrow morning?)
3. Practical:
    a. Get familiar with decision tree, Random forest, Adaboost, xgboost libraries and use them on some datasets.