DATA 301:
Gradient Boosted Trees
(lightGBM, catboost)

Topics

Introduction
Bagging verses Boosting
Example
Boosting Benefits
Boosting Drawbacks
Packages
Summary

Introduction

Random forest are a collection of decision trees that are created using a technique called ‘bagging’
Which means create a bunch of independent decision trees and average (or majority vote) their results

Boosted decision trees are a collection of decision trees that are created using a technique called ‘boosting’
Which means create the trees one at a time, each new tree designed to improve upon previous trees estimates

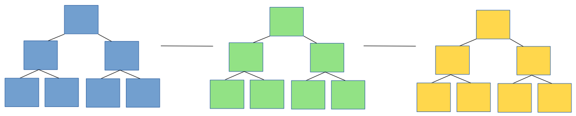
Bagging verses Boosting

Bagging

Bagging verses Boosting

Bagging

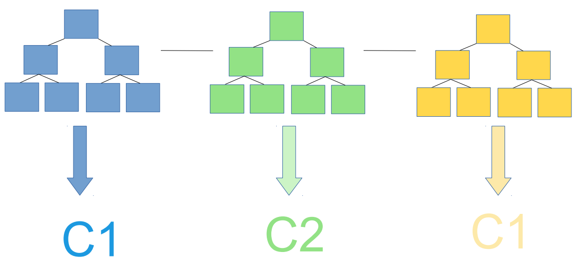
Multiple independent trees



Bagging verses Boosting

Bagging

Multiple independent trees



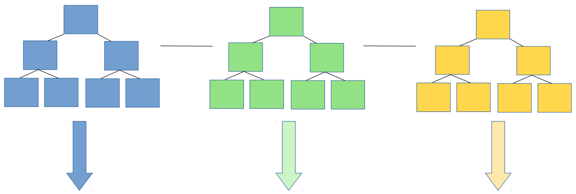
For Classification
Use majority vote

C1 C2 C1 = C1

Bagging verses Boosting

Bagging

Multiple independent trees



p1 p2 p3

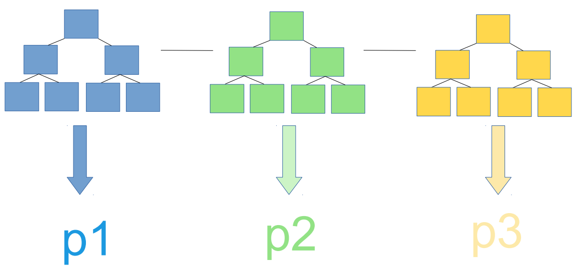
For Regression just Average results

(p1 + p2 + p3)/3 =val

Bagging verses Boosting

Bagging

Multiple independent trees



For Regression just Average results

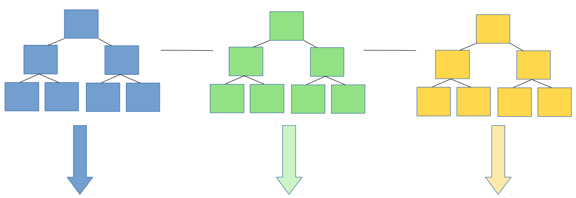
(p1 + p2 + p3)/3 =val

Build trees in parallel so very fast

Bagging verses Boosting

Bagging

Multiple independent trees



p1 p2 p3

For Regression just Average results

(p1 + p2 + p3)/3 =val

Build trees in parallel so very fast

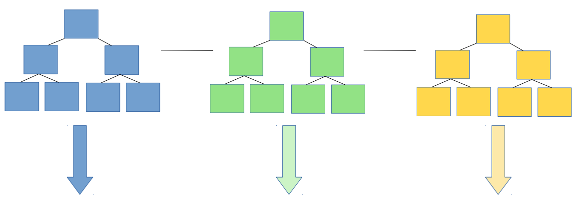
Boosting

Start with average target value

Bagging verses Boosting

Bagging

Multiple independent trees



p1 p2 p3

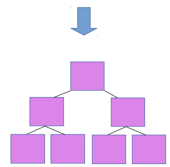
For Regression just Average results

$(P1 + p2 + p3)/3 = val$

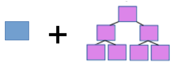
Build trees in parrallel so very fast

Boosting

Start with average target value



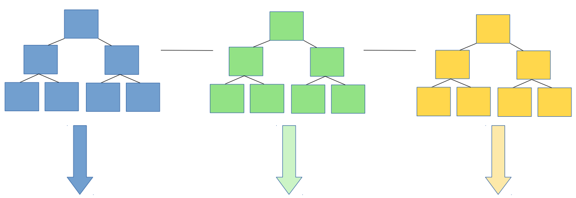
Create tree based on errors from previous ensemble and combine outputs



Bagging verses Boosting

Bagging

Multiple independent trees



p1 p2 p3

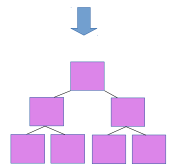
For Regression just Average results

(P1 + p2 + p3)/3 =val

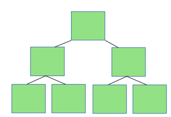
Build trees in parrallel so very fast

Boosting

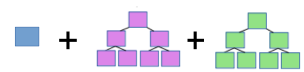
Start with average target value



Create tree based on errors from previous ensemble and combine outputs



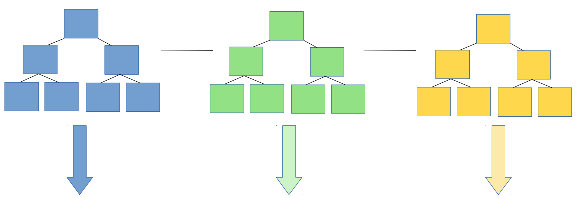
Create tree based on errors from previous ensemble and combine outputs



Bagging verses Boosting

Bagging

Multiple independent trees



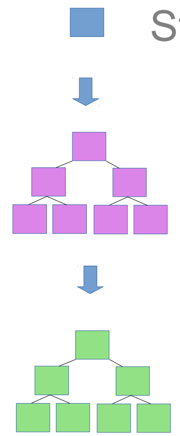
p1 p2 p3

For Regression just Average results

(P1 + p2 + p3)/3 =val

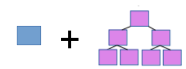
Build trees in parrallel so very fast

Boosting

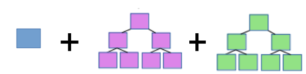


Start with average target value

Create tree based on errors from previous ensemble and combine outputs



Create tree based on errors from previous ensemble and combine outputs

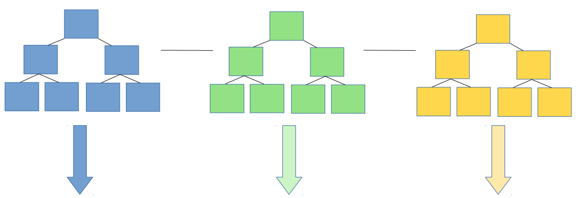


Continue until build number trees requested
Or additional trees fail to improve prediction

Bagging verses Boosting

Bagging

Multiple independent trees



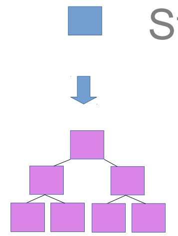
p1 p2 p3

For Regression just Average results

(P1 + p2 + p3)/3 =val

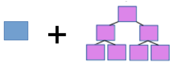
Build trees in parrallel so very fast

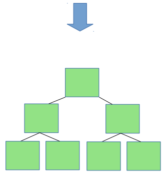
Boosting



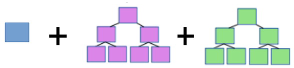
Start with average target value

Create tree based on errors from previous ensemble and combine outputs





Create tree based on errors from previous ensemble and combine outputs



Continue until build number trees requested
Or additional trees fail to improve prediction

Build trees sequentially so slow.
But more accurate than bagged methods like Random Forest

Example

Height	Color	Gender	Weight
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Average weight

71.2

Calculate average weight

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

Average weight

71.2

Calculate difference between average weight and Weight
Add as new column, Residuals
(1st row 88-71.2=16.8)

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Average weight

71.2

Calculate difference between average weight and Weight
Add as new column Residuals
(1st row 88-71.2=16.8)
Do this for All rows

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Average weight

71.2

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Now build a tree to predict the Residuals.
Use Height, Color and Gender.Trees have several tuning Parameters,

- max_depth= how many levels per tree
- max_leaf_nodes: number terminal leaf nodes

Set max_leaf_nodes = 4 for this problem

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Average weight

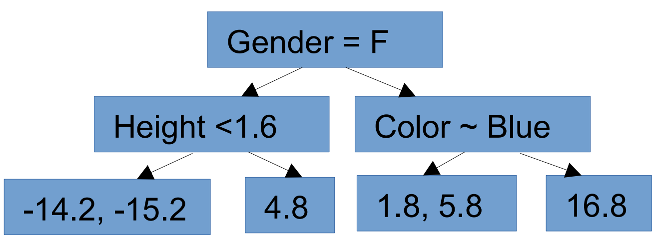
71.2

Now build a tree to predict the Residuals
Use Height, Color and Gender.Trees have several tuning Parameters,

max_depth= how many levels per tree

max_leaf_nodes: number terminal leaf nodes

Set max_leaf_nodes = 4 for this problem



Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Average weight

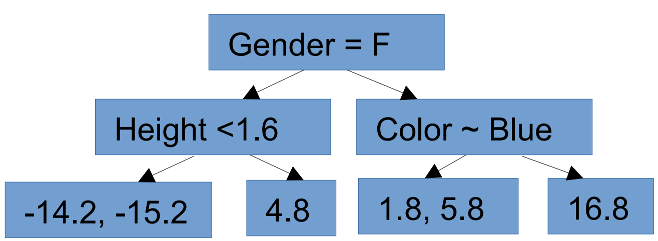
71.2

Now build a tree to predict the Residuals
Use Height, Color and Gender.Trees have several tuning Parameters,

max_depth= how many levels per tree

max_leaf_nodes: number terminal leaf nodes

Set max_leaf_nodes = 4 for this problem



But can have a max of only 4 leaf nodes

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Average weight

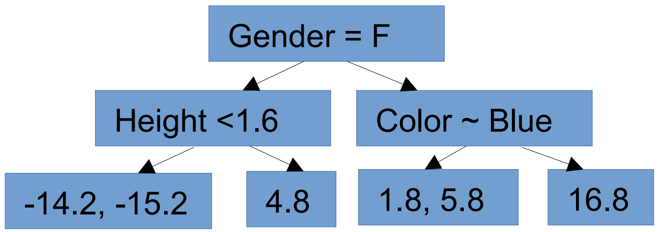
71.2

Now build a tree to predict the Residuals
Use Height, Color and Gender.Trees have several tuning Parameters,

max_depth= how many levels per tree

max_leaf_nodes: number terminal leaf nodes

Set max_leaf_nodes = 4 for this problem



So average the leaf nodes with more than 2 values
(-14.2+-15.2)/2=-14.7
(1.8 + 5.8)/2=3.8

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

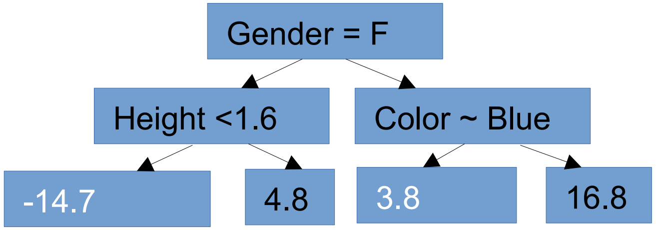
Example

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Average weight

71.2

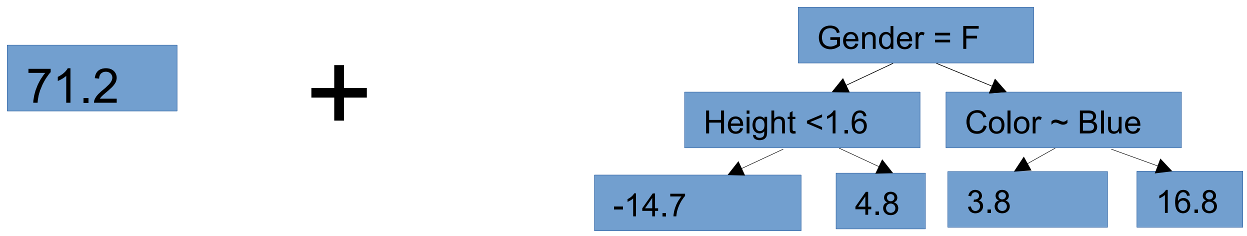
Now build a tree to predict the Residuals
Use Height, Color and Gender.Trees have several tuning Parameters,
  max_depth= how many levels per tree
  max_leaf_nodes: number terminal leaf nodes
Set max_leaf_nodes = 4 for this problem



So average the leaf nodes with more than 2 values
(-14.2+-15.2)/2=-14.7
(1.8 + 5.8)/2=3.8

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

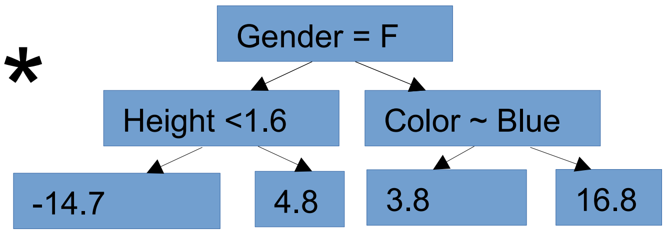


Combine new tree with
Original leaf and use to
calculate new residuals

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (lr)*

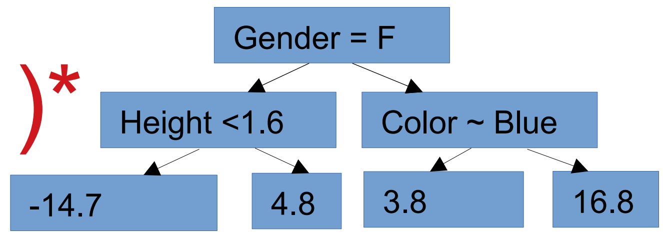


Use only part of the new trees
prediction to prevent overfitting (low bias, high variance) by
Multiplying it’s output by learning rate <1

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

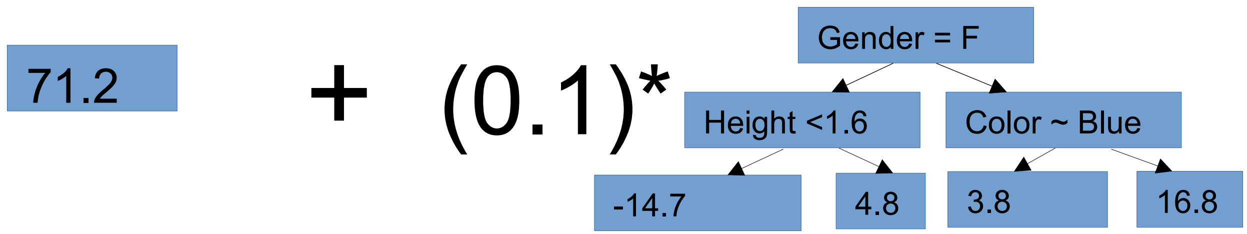
71.2 + (0.1)*



Use only part of the new trees
prediction to prevent overfitting (low bias, high variance) by
Multiplying it’s output by learning rate <1
Lr=0.1

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example



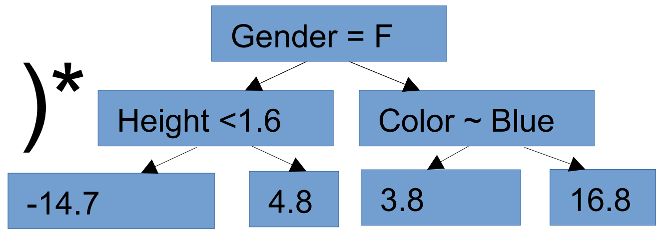
Calculate predicted weight (for row 0)
71.2 +0.1*16.8=72.9

Height	Color	Gender	Weight
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (0.1)*



Calculate predicted weight (for row 0)
71.2 +0.1*16.8=72.9

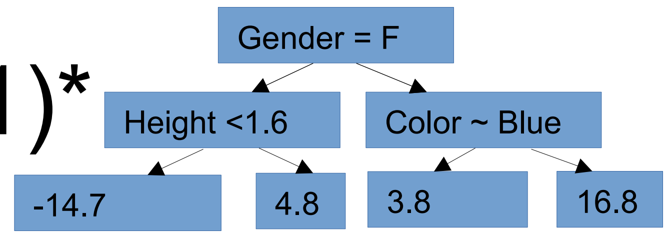
Which is a little better than 71.2 (the original average estimate)

Height	Color	Gender	Weight
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (0.1)*



Calculate predicted weight (for row 0)
71.2 +0.1*16.8=72.9

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

Which is a little better than 71.2

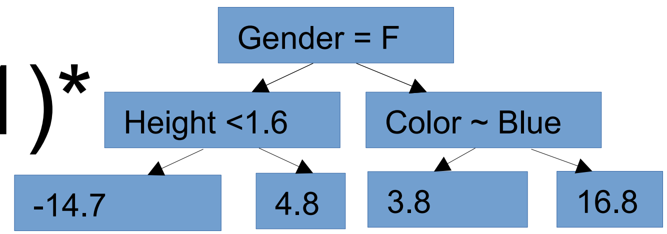
Calculate the new residuals (first row)
88-72.9=15.1

We are getting closer to the true weight

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (0.1)*



Calculate predicted weight (for row 0)
71.2 +0.1*16.8=72.9

Which is a little better than 71.2

Calculate the new residuals (first row)
88-72.9=15.1

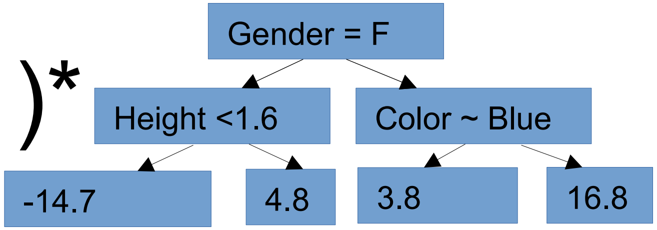
Do for all rows

Height	Color	Gender	Weight	Residuals
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	4.3
1.5	Blue	Female	56	-13.7
1.8	Red	Male	73	1.4
1.5	Green	Male	77	5.4
1.4	Blue	Female	57	-12.7

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (0.1)*



Original residuals

Residuals
16.8
4.8
-15.2
1.8
5.8
-14.2

New residuals

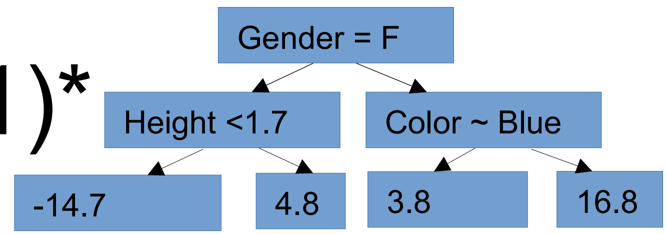
Residuals
15.1
4.3
-13.7
1.4
5.4
-12.7

Note that the New Residuals are lower Than the originals. We are reducing the Error as we add more trees.

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

71.2 + (0.1)*



Original residuals

Residuals
16.8
4.8
-15.2
1.8
5.8
-14.2

New residuals

Residuals
15.1
4.3
-13.7
1.4
5.4
-12.7

Note that the New Residuals are lower Than the originals. We are reducing the Error as we add more trees.

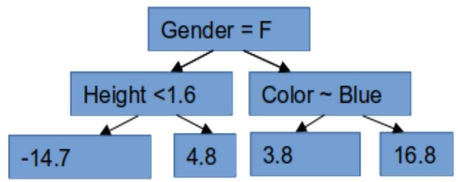
Repeat the process of calculating Residuals and building trees until Either max trees are reached or Residuals stop decreasing.

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

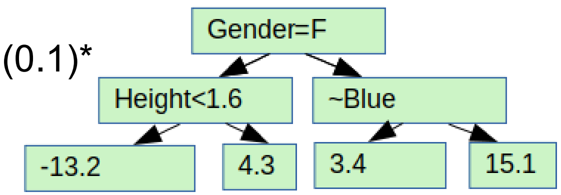
Example

When we have enough trees, we can predict weight

weight= 71.2 + (0.1)*



+ (0.1)*

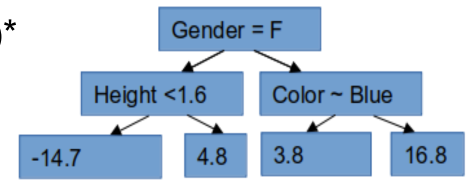


Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

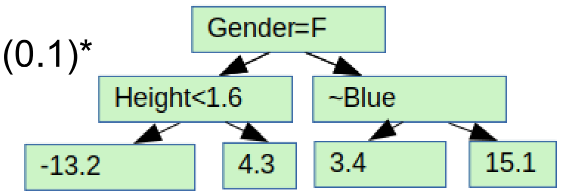
Example

When we have enough trees, we can predict weight

weight= 71.2 + (0.1)*



+ (0.1)*



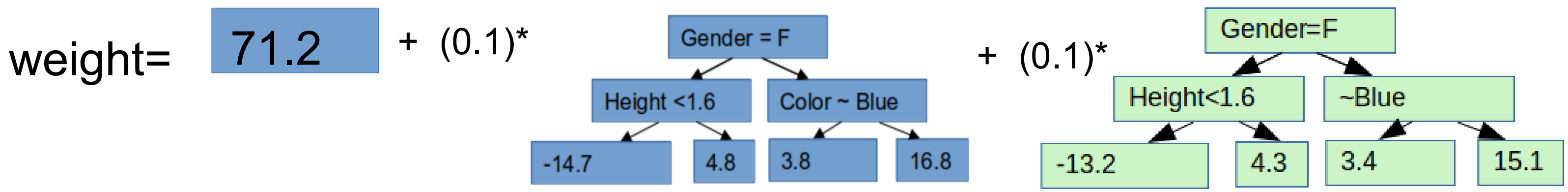
Height	Color	Gender	Weight
1.6	Blue	Male	88

Weight= 71.2 + 0.1*16.8 + 0.1*(15.1)
= 74.39

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Example

When we have enough trees, we can predict weight



Height	Color	Gender	Weight
1.6	Blue	Male	88

Weight= 71.2 + 0.1*16.8 + 0.1*(15.1)
= 74.39

The more trees you have the more accurate it gets (at the risk of overfitting)

Example from https://www.youtube.com/watch?v=3CC4N4z3GJc

Benefits

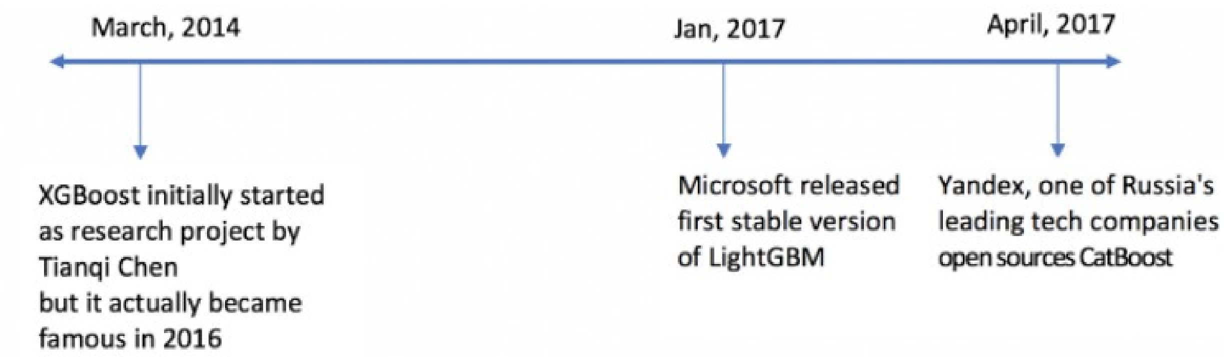
- Reducing residual approach lets trees push wrong answers in the ‘right’ direction.
- Each tree tries to improve the overall model by reducing residuals. Trees work together.
- More accurate than random forest, where each tree makes an independent estimate.

Drawbacks

- Trees calculated serially. Much slower than Random Forest which is calculated in parallel.
- More hyperparameters to tune (learning rate, max_tree_depth, max_leaf_nodes etc.)

Packages

- LightGBM: by Microsoft, gradient boosted trees
- Catboost: by Yandex, more gradient boosted trees
- Sklearn: GradientBoostingClassifier and GradientBoostingRegressor, not covered here since LightGBM and Catboost are faster, more accurate and support sklearns default model training procedure
- XGBoost: still more gradient boosted trees, not covered here because they take MUCH longer to train than catboost or LightGBM



Tunable Parameters

Function	CatBoost	Light GBM
Important parameters which control overfitting	1. **Learning_rate** 2. **Depth** - value can be any integer up to 16. Recommended - [1 to 10] 3. No such feature like min_child_weight 4. **l2-leaf-reg**: L2 regularization coefficient. Used for leaf value calculation (any positive integer allowed)	1. **learning_rate** 2. **max_depth**: default is 20. Important to note that tree still grows leaf-wise. Hence it is important to tune **num_leaves** (number of leaves in a tree) which should be smaller than 2^(max_depth). It is a very important parameter for LGBM 3. **min_data_in_leaf**: default=20, alias= min_data, min_child_samples
Parameters for categorical values	1. **cat_features**: It denotes the index of categorical features 2. **one_hot_max_size**: Use one-hot encoding for all features with number of different values less than or equal to the given parameter value (max – 255)	1. **categorical_feature:** specify the categorical features we want to use for training our model
Parameters for controlling speed	1. **rsm:** Random subspace method. The percentage of features to use at each split selection 2. No such parameter to subset data 3. **iterations**: maximum number of trees that can be built; high value can lead to overfitting	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data to be used for each iteration and is generally used to speed up the training and avoid overfitting 3. **num_iterations:** number of boosting iterations to be performed; default=100

Original figure: https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

Summary

- Gradient Boosted trees are the preferred tree ensemble given it's increase in accuracy (or F1, or R^2 or whatever performance metric of choice)
- Work with regression and classification
- Built into scikitlearn
- Harder to tune (more hyperparameters)
- Longer to train