Statistics and Machine Learning

Tree-based algorithm: Improvements
Bagging, Random Forest, Boosting

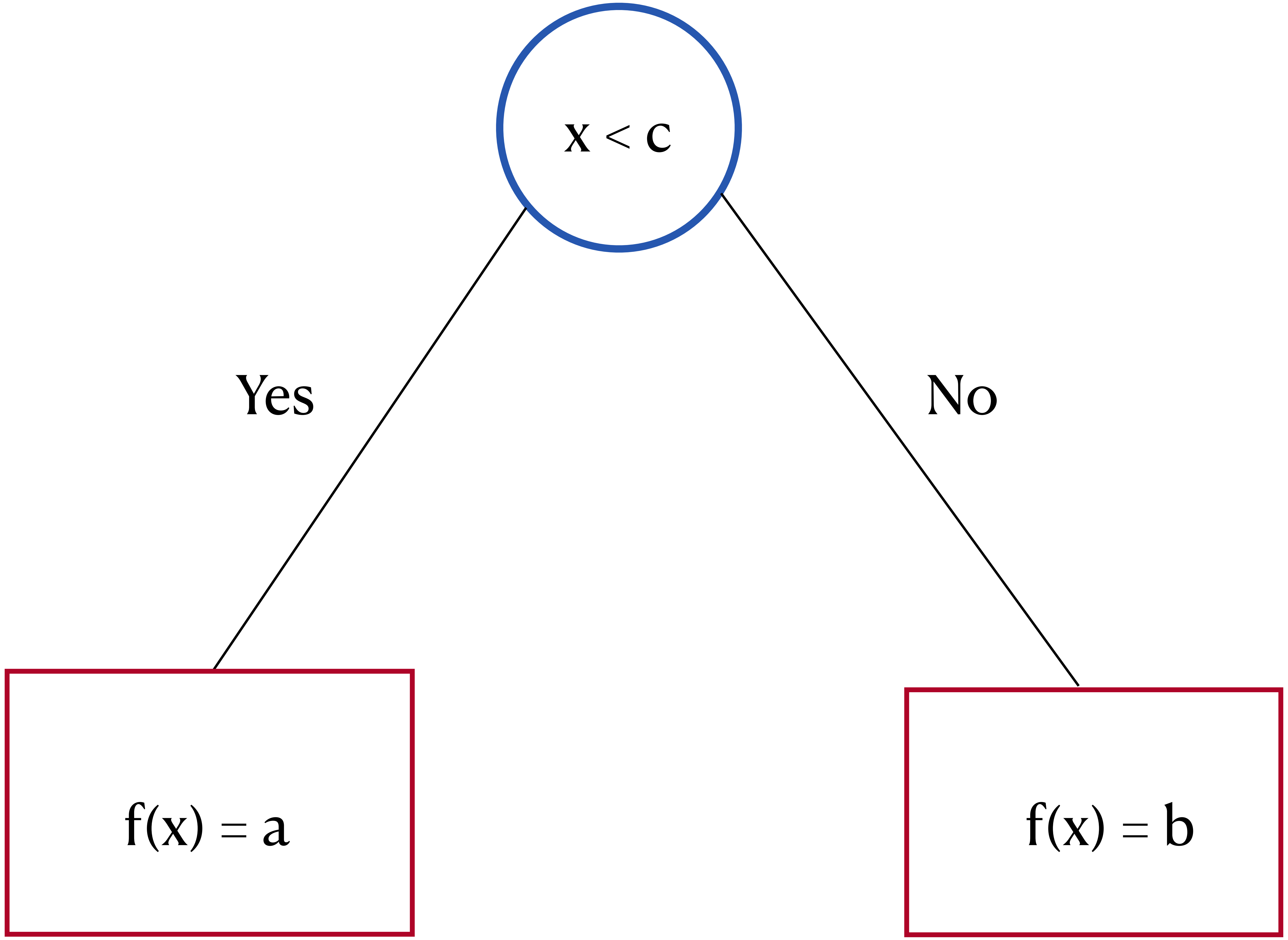
Week 12 Slide 04/05 — 04/09 2021

Contents

Week 12

- Decision trees are simple, but not good enough; bigger trees better but overfit
- 1st improvement: from one tree to one forest — bagging
- 2nd improvement: inserting randomness into tree branching — random forest
- 3rd improvement: make next tree greater than previous tree — boosting
- Ada-boosting and gradient-boosting
- Homework

Simple trees (stump) usually under-fit



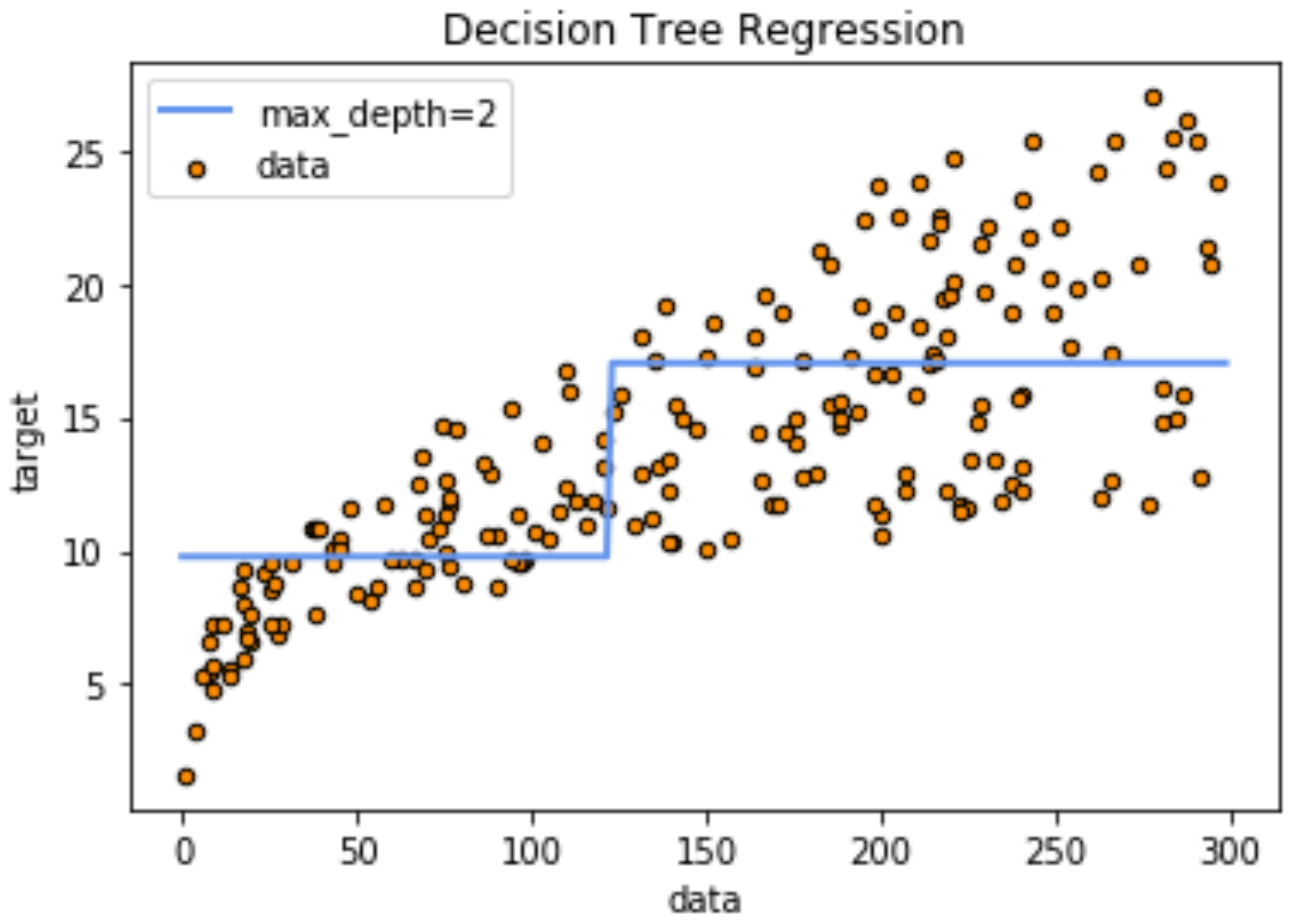
x < c

Yes

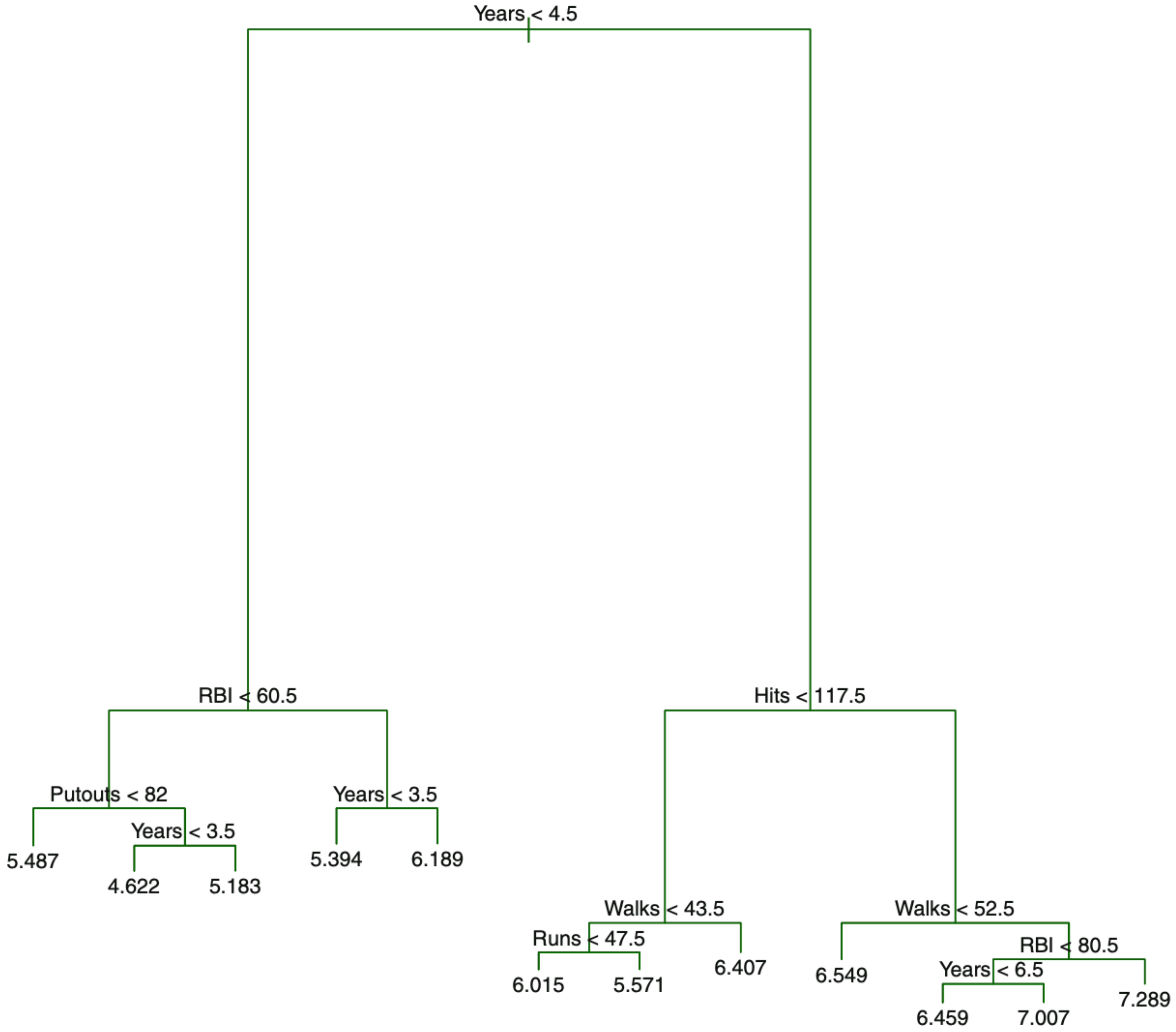
No

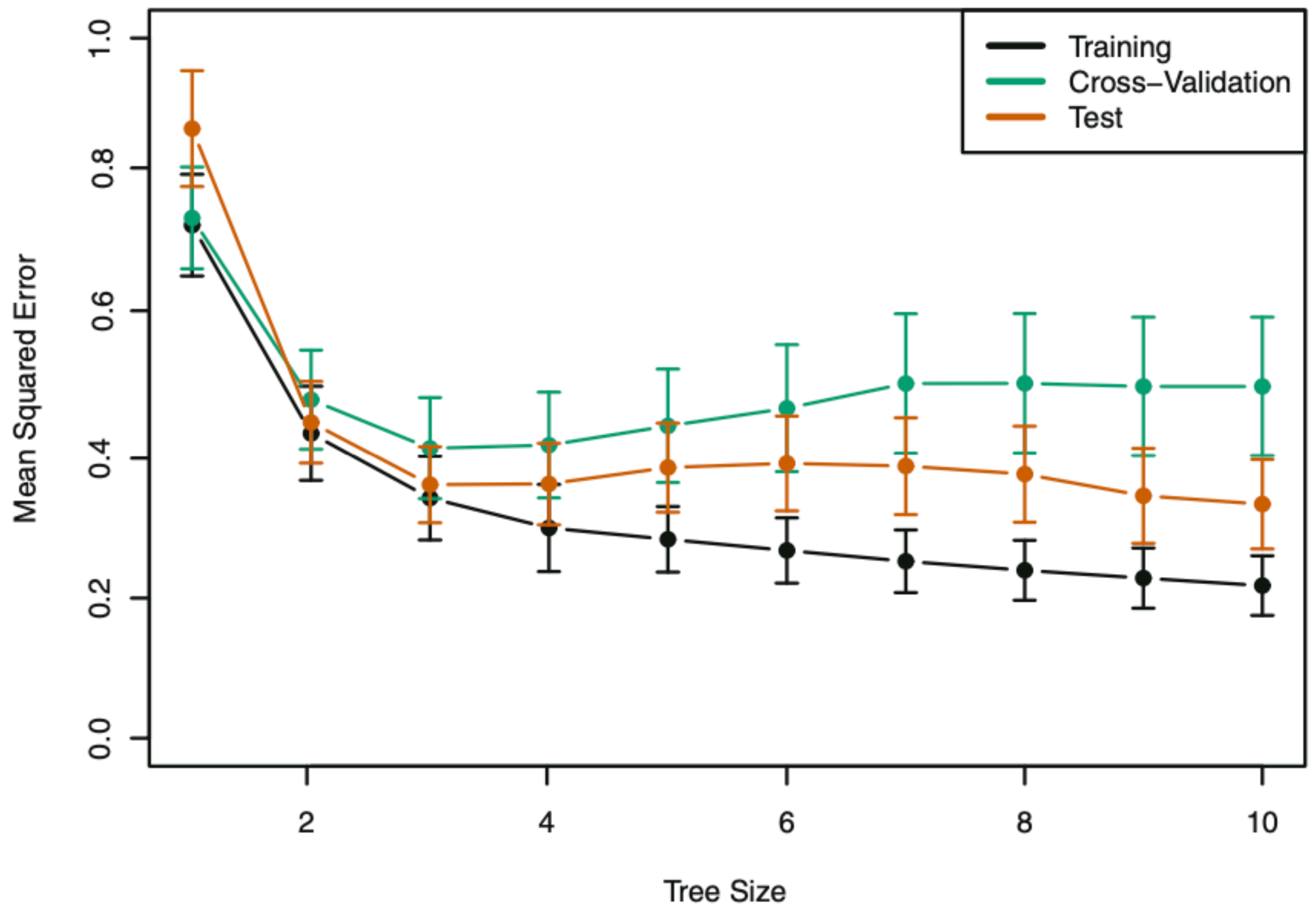
f(x) = a

f(x) = b



Deeper trees often over-fit





A forest of stumps is better than a big tree

Power of AVERAGE

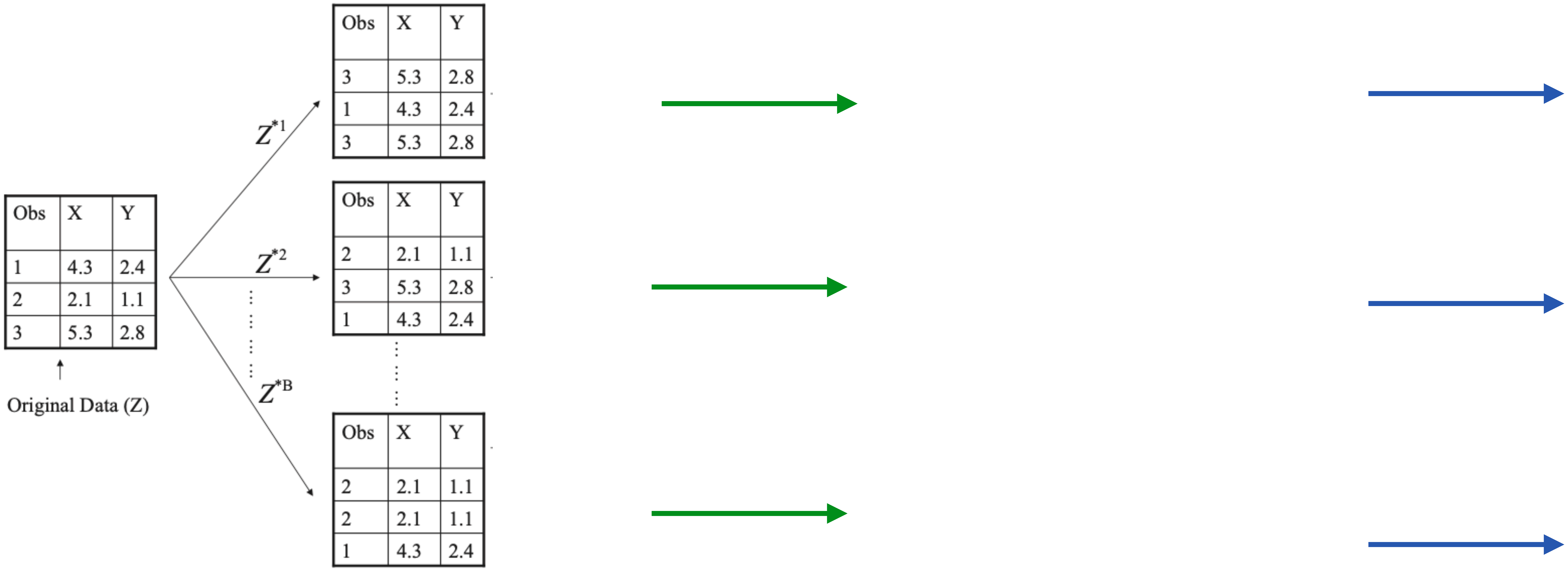
Think about playing the following game: find 10 people and let each one can generate a random number from Numpy.random.normal(0,1). The person whose number is closest to zero wins the contest. But who will win is unpredictable because everyone has an equal chance. However, if 5 participants decide to group together and average their random numbers, then the group has much bigger chance of winning over the rest individuals. Why?

Assume $E[X_1] = E[X_2] = \cdots E[X_{10}] = 0$ And $E[X_1^2] = E[X_2^2] = \cdots E[X_{10}^2] = 1$

$$\longrightarrow \quad E[\frac{X_1 + \cdots + X_5}{5}] = 0 \qquad E[(\frac{X_1 + \cdots + X_5}{5})^2] = \frac{1}{5}$$

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

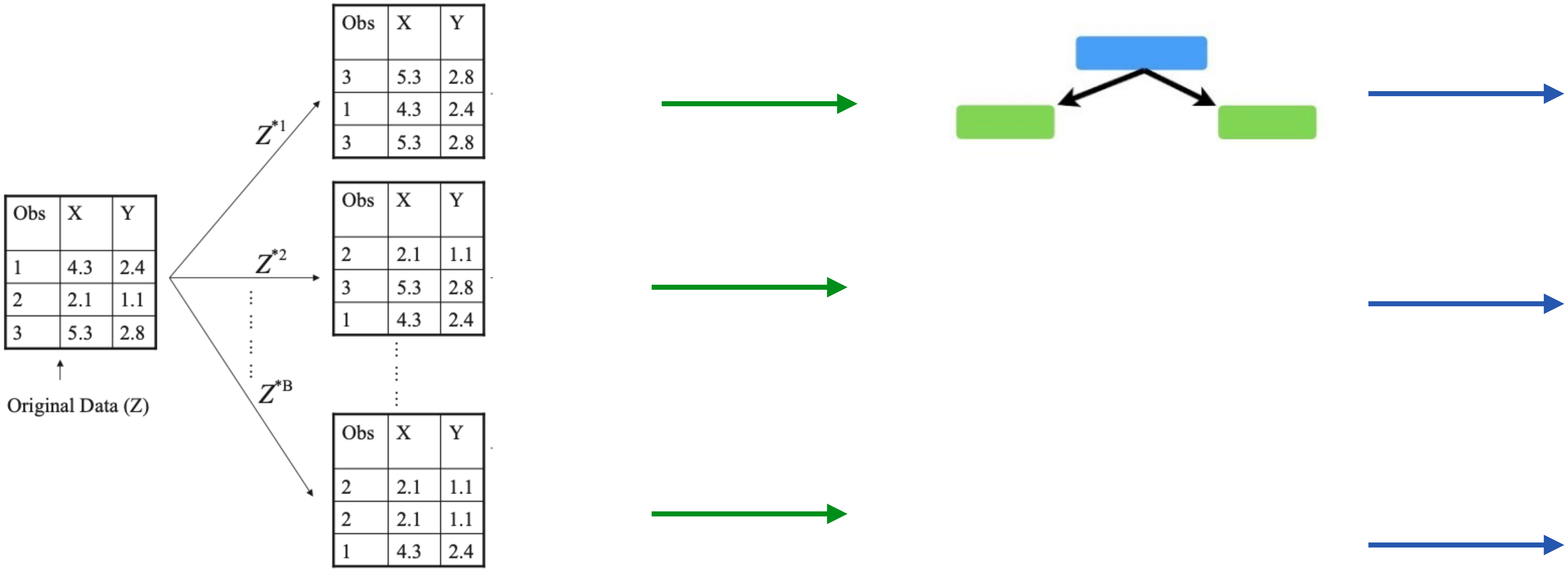
Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

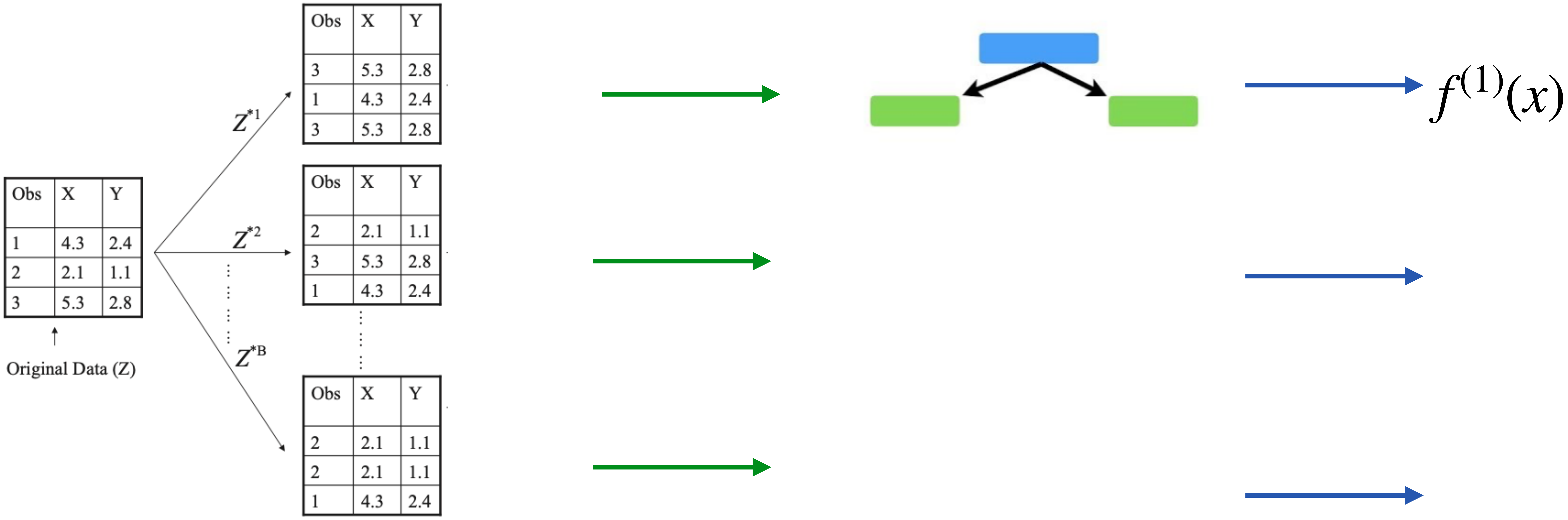
Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

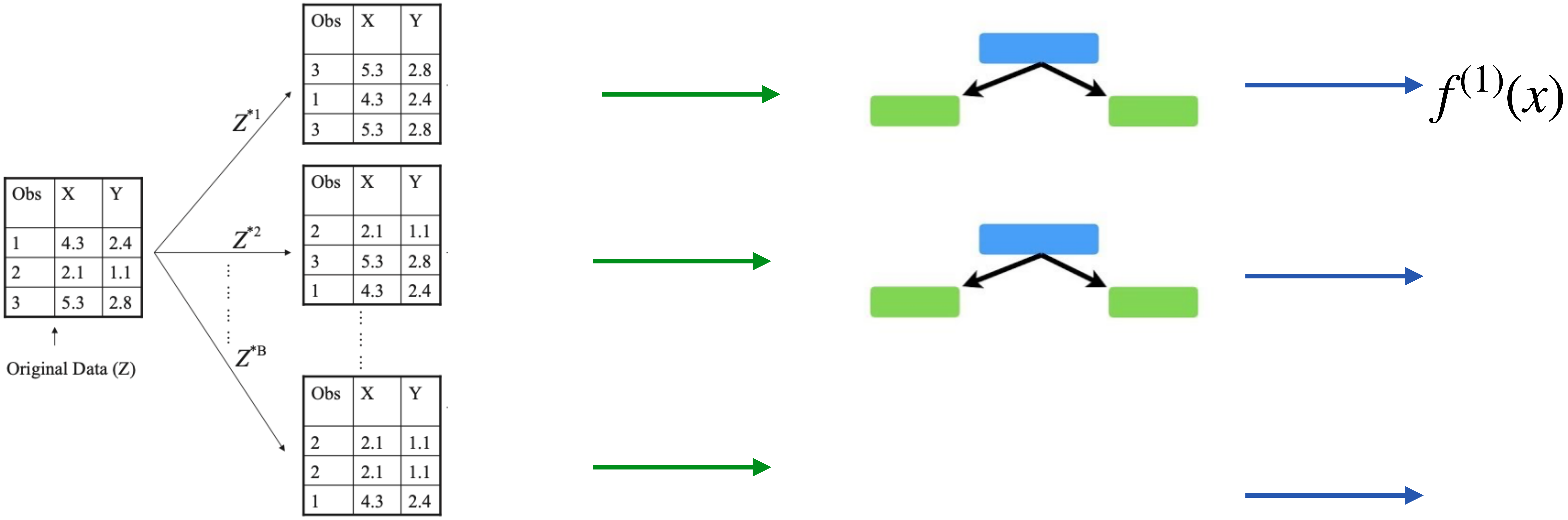
Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Original Data (Z)

Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Z^{*2}

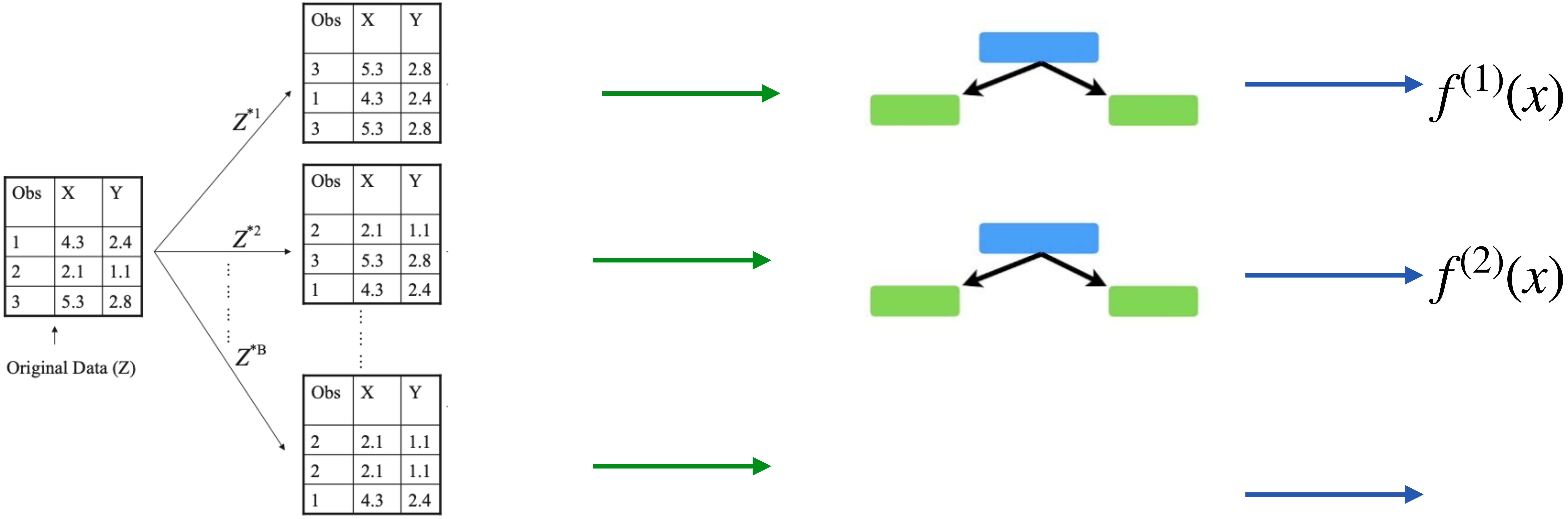
Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Z^{*2}

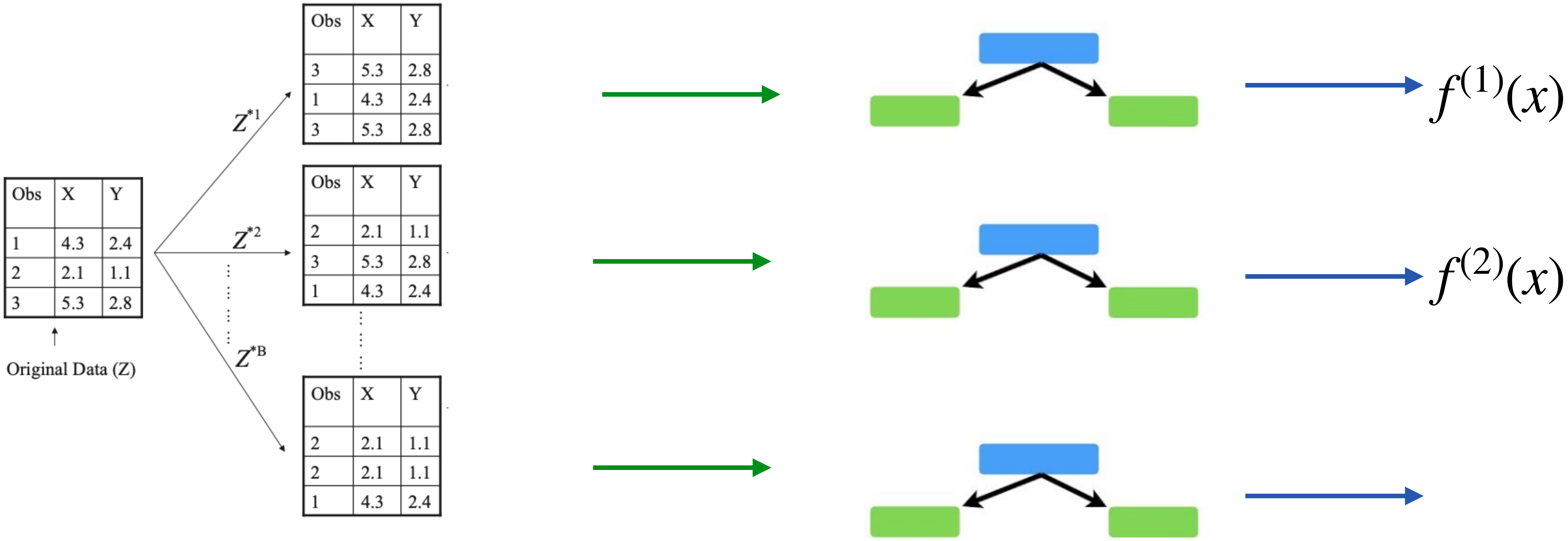
Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

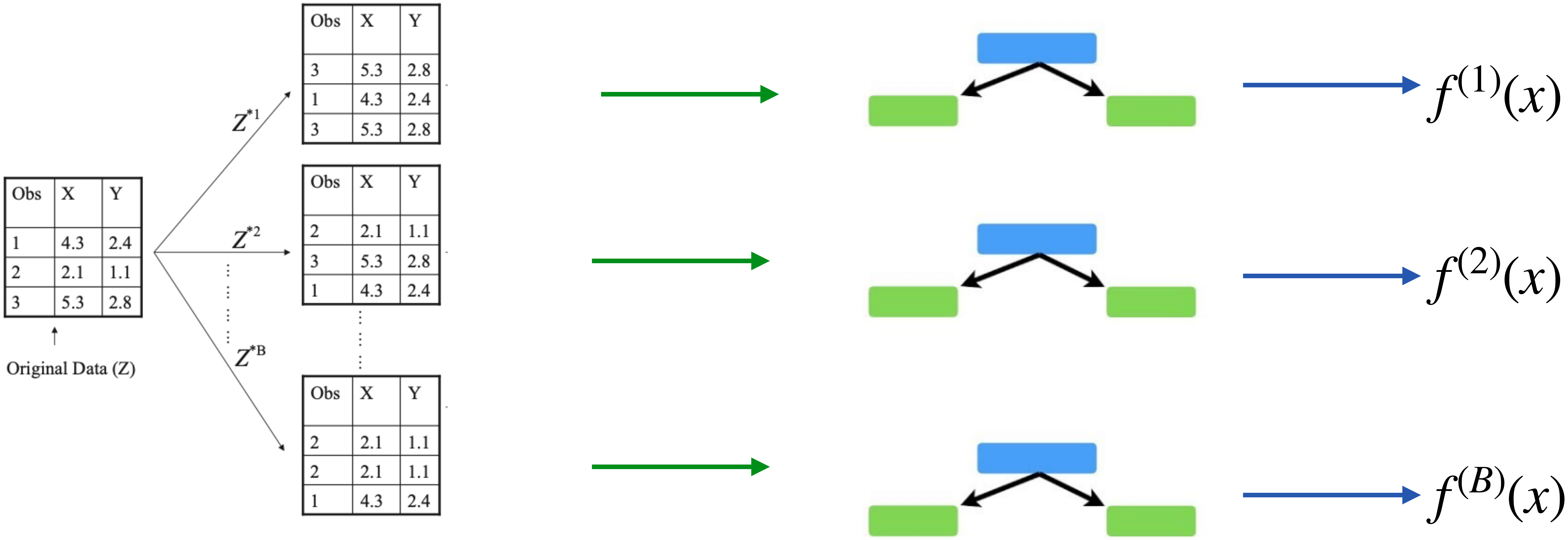
Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

How to grow different trees with only one data set?

Bootstrapping data set is to generate copies of different versions of one data set



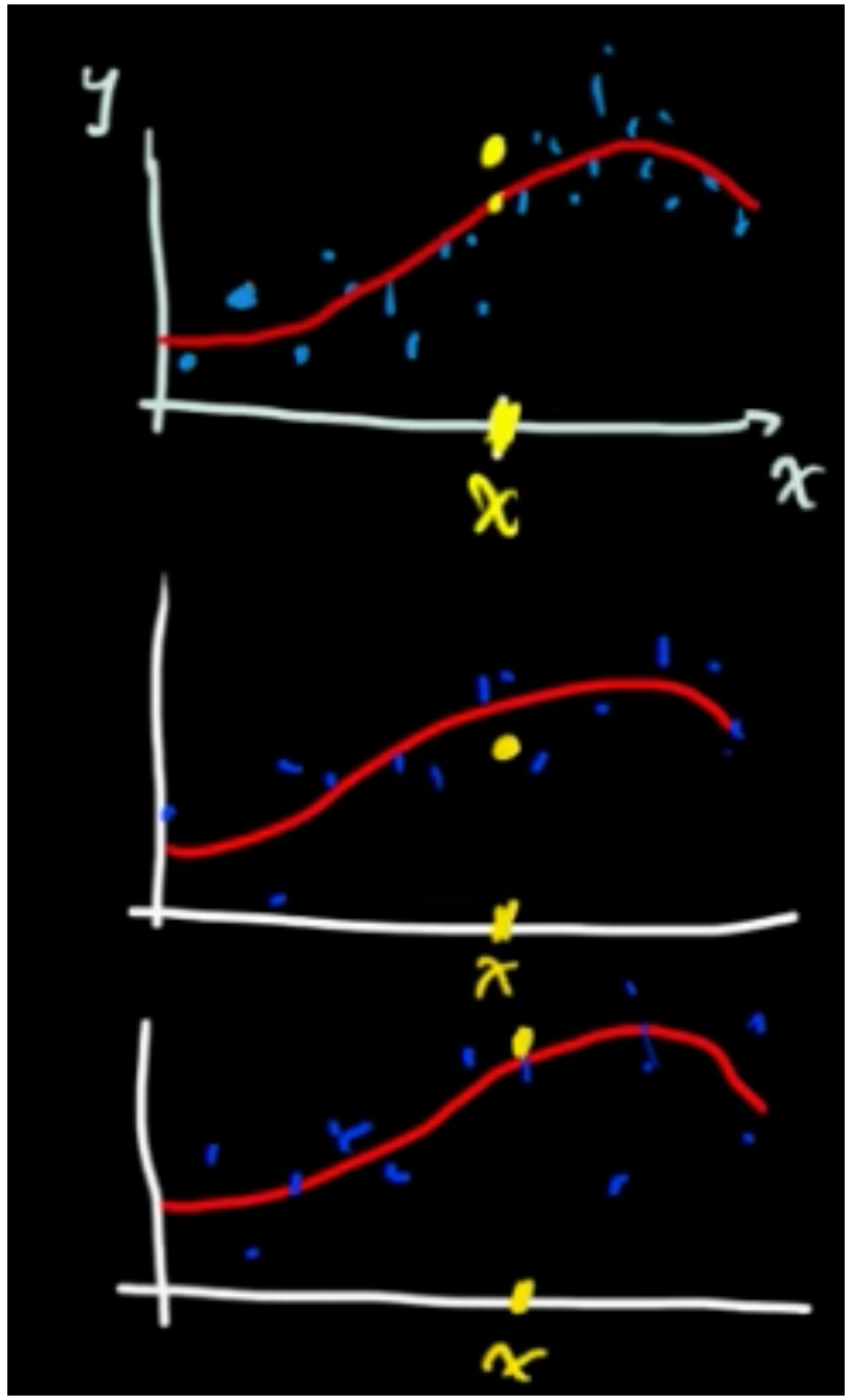
Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4



<—- Using all data at once.

<— Using bootstrapped dataset #1

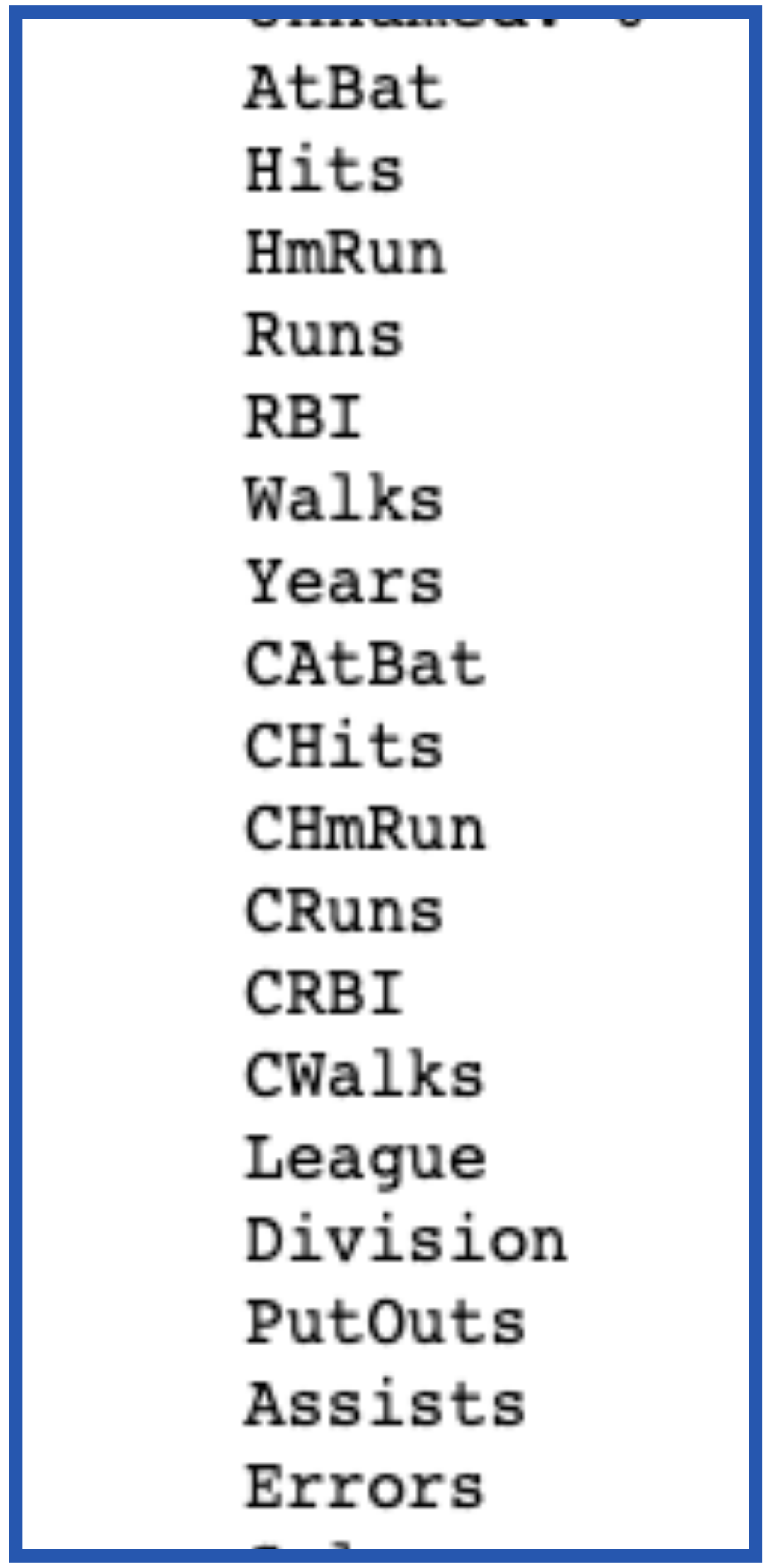
<— Using bootstrapped dataset #2

In the end, we average all the prediction.

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} \hat{f}(x)$$

Random Forest

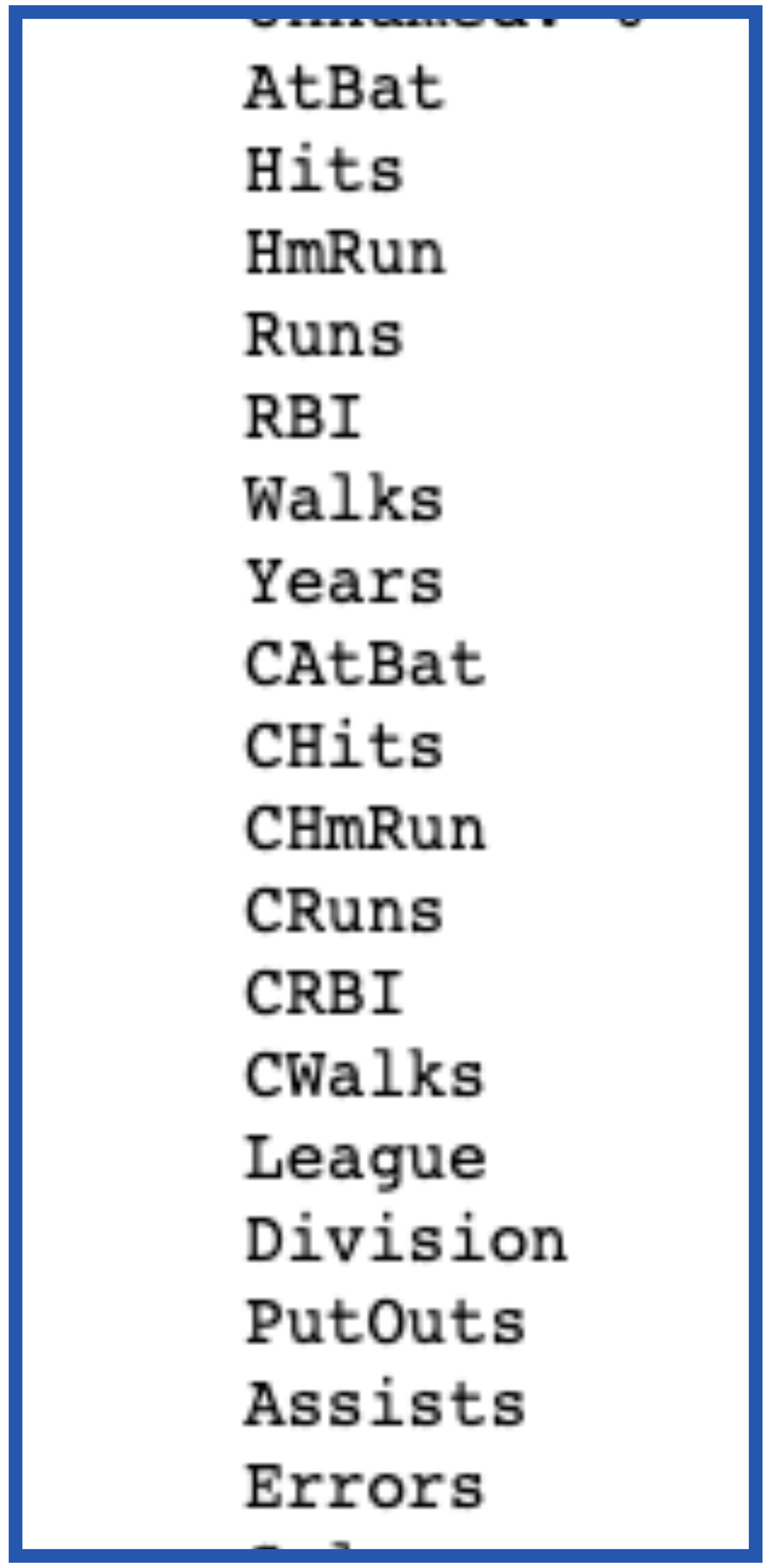
The point: randomly select features when branching



```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

Random Forest

The point: randomly select features when branching

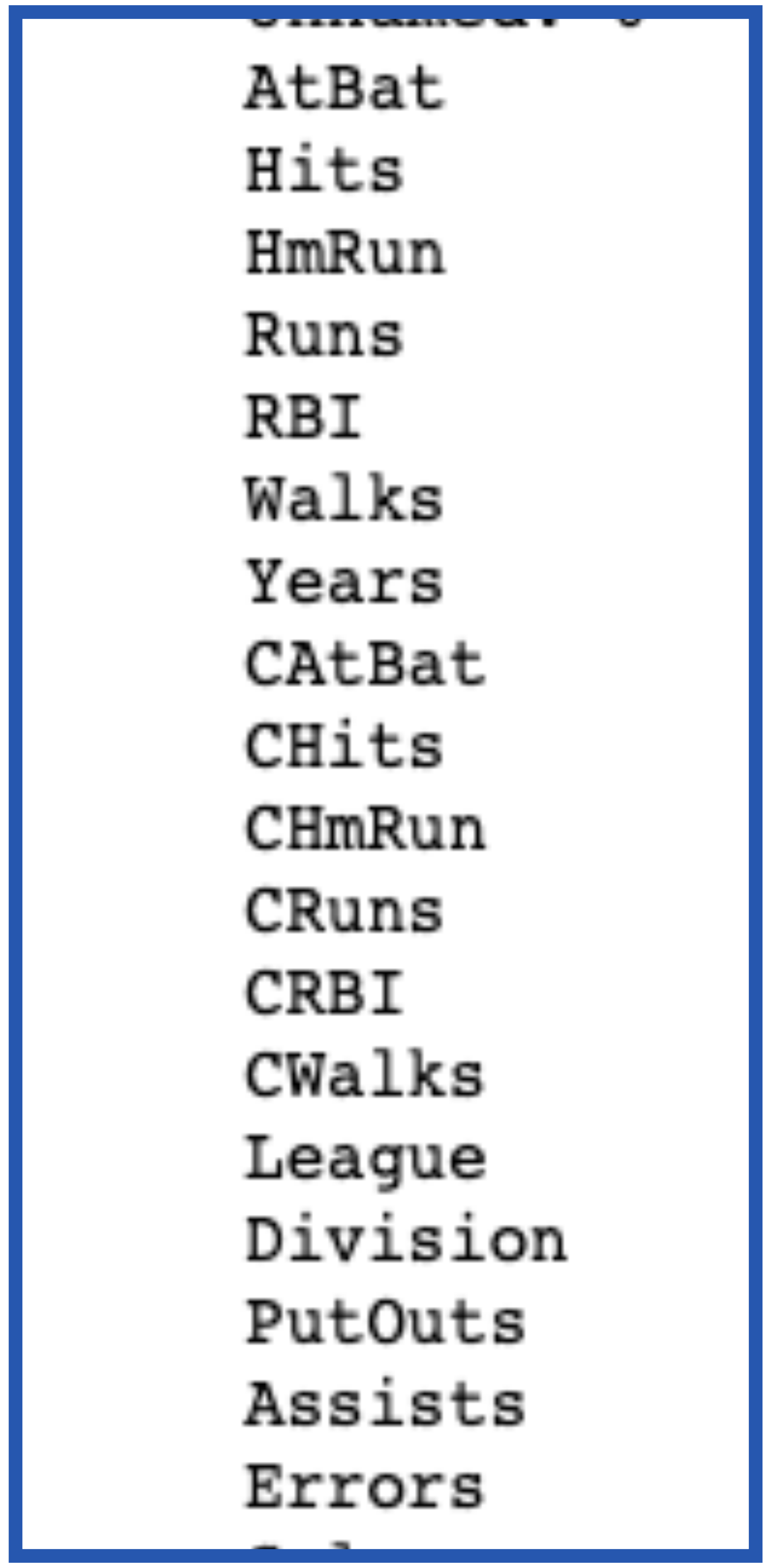


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets

Random Forest

The point: randomly select features when branching

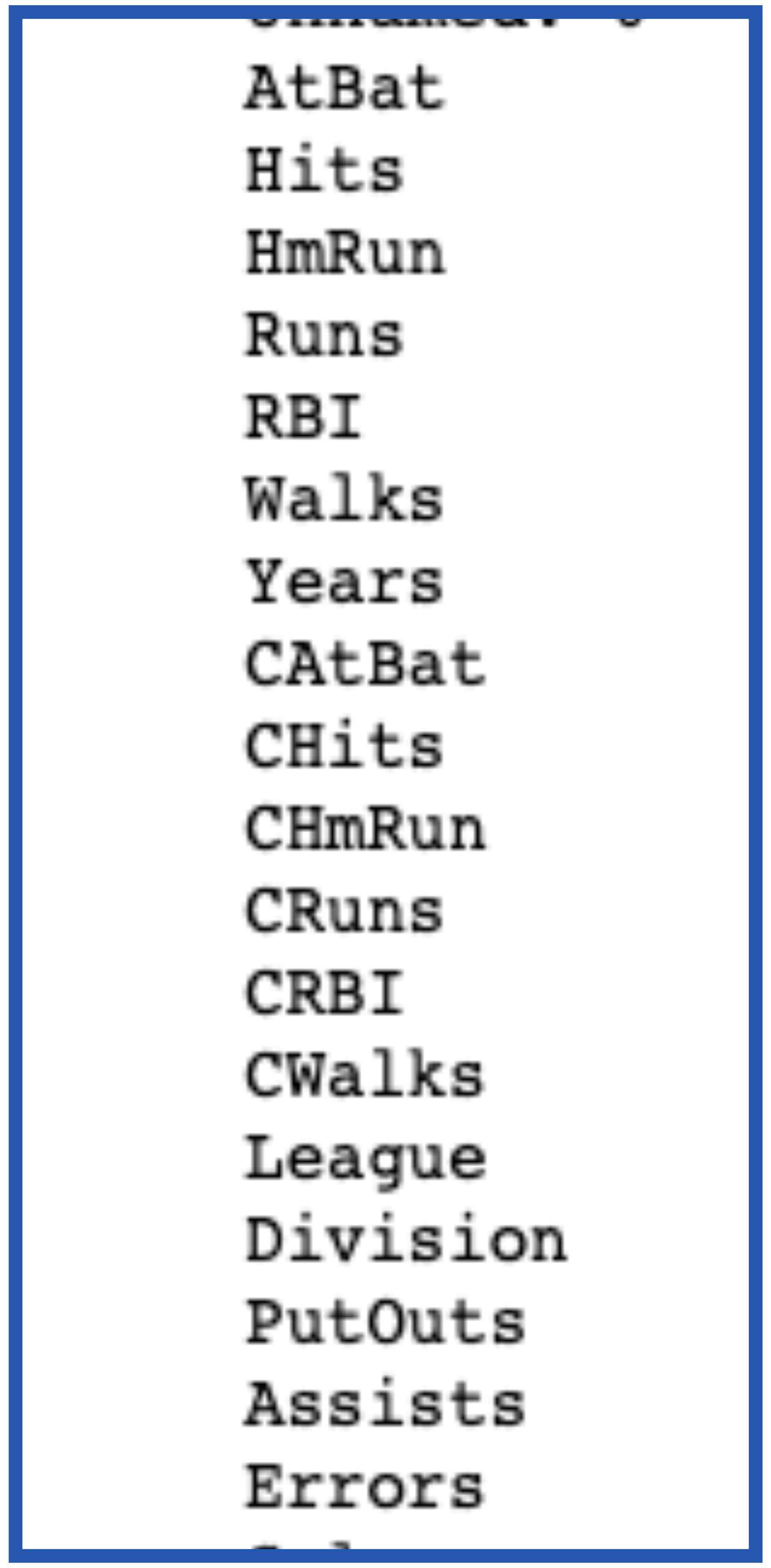


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set

Random Forest

The point: randomly select features when branching

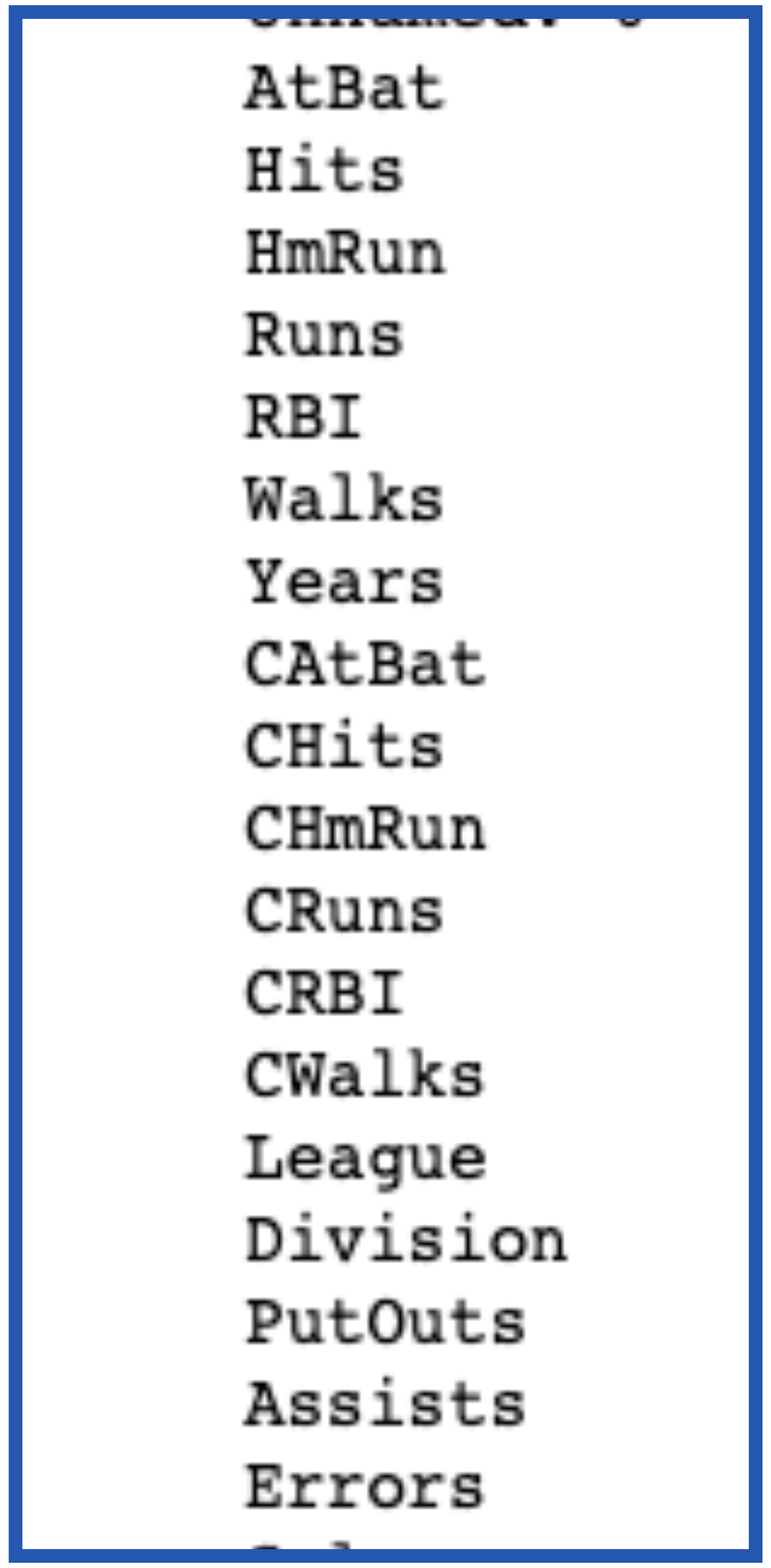


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set
- Step 3: Parameter $p \approx \sqrt{M}$ is determined

Random Forest

The point: randomly select features when branching

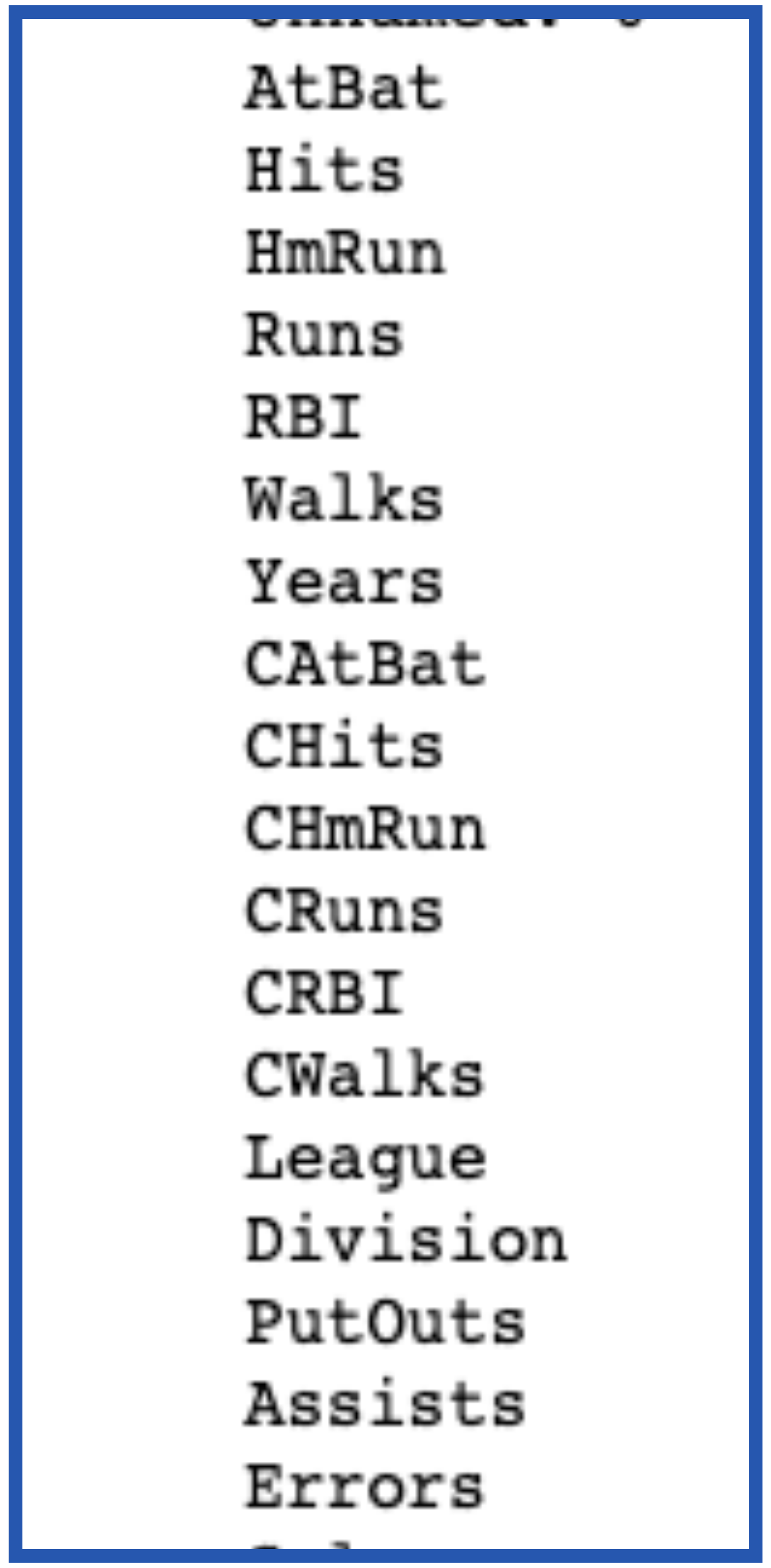


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set
- Step 3: Parameter $p \approx \sqrt{M}$ is determined
- Step 4: For data set *i*, randomly select p features.

Random Forest

The point: randomly select features when branching

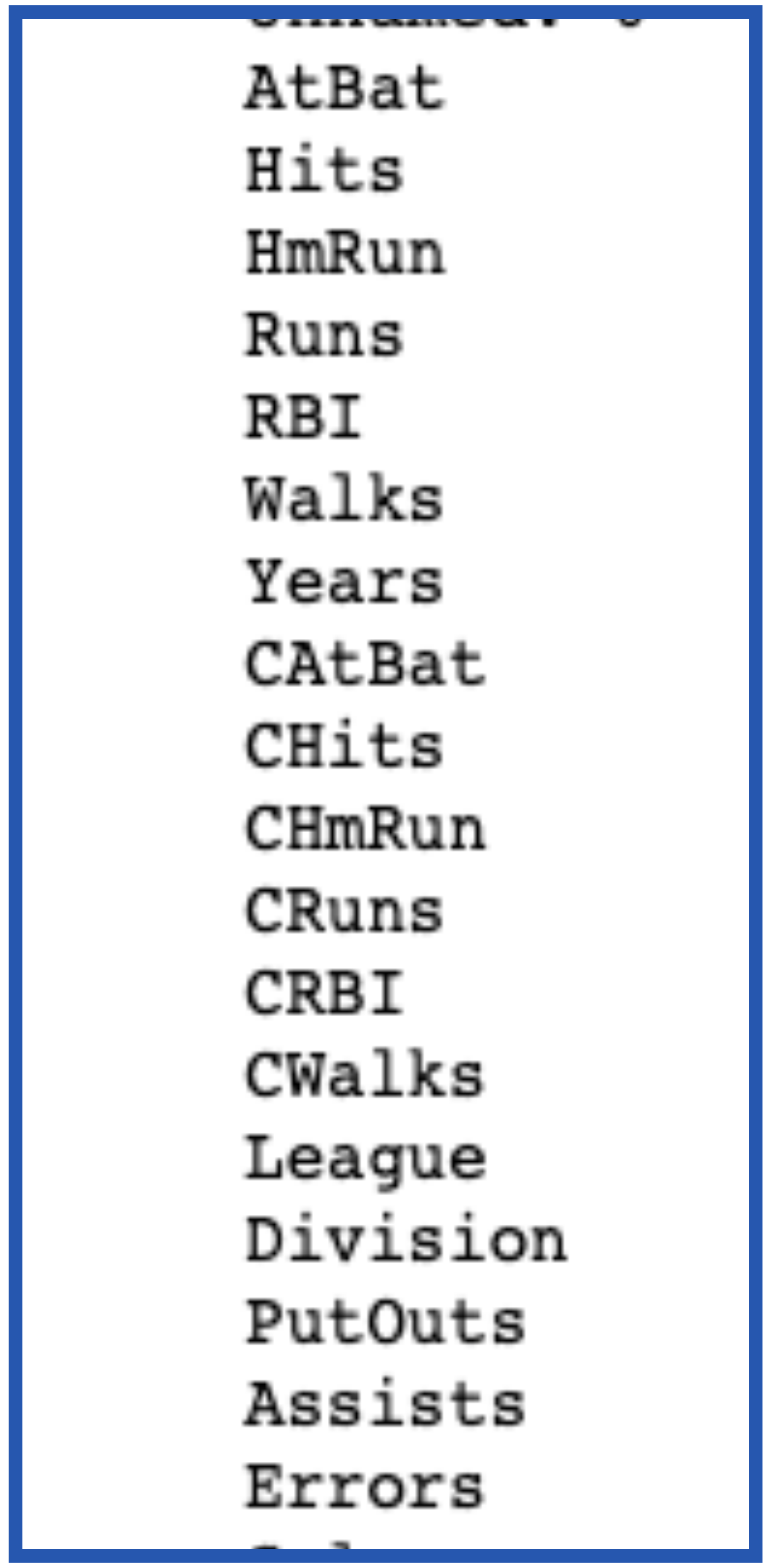


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set
- Step 3: Parameter $p \approx \sqrt{M}$ is determined
- Step 4: For data set i, randomly select p features.
- Step 5: Grow a tree and can only use the p features to branch

Random Forest

The point: randomly select features when branching

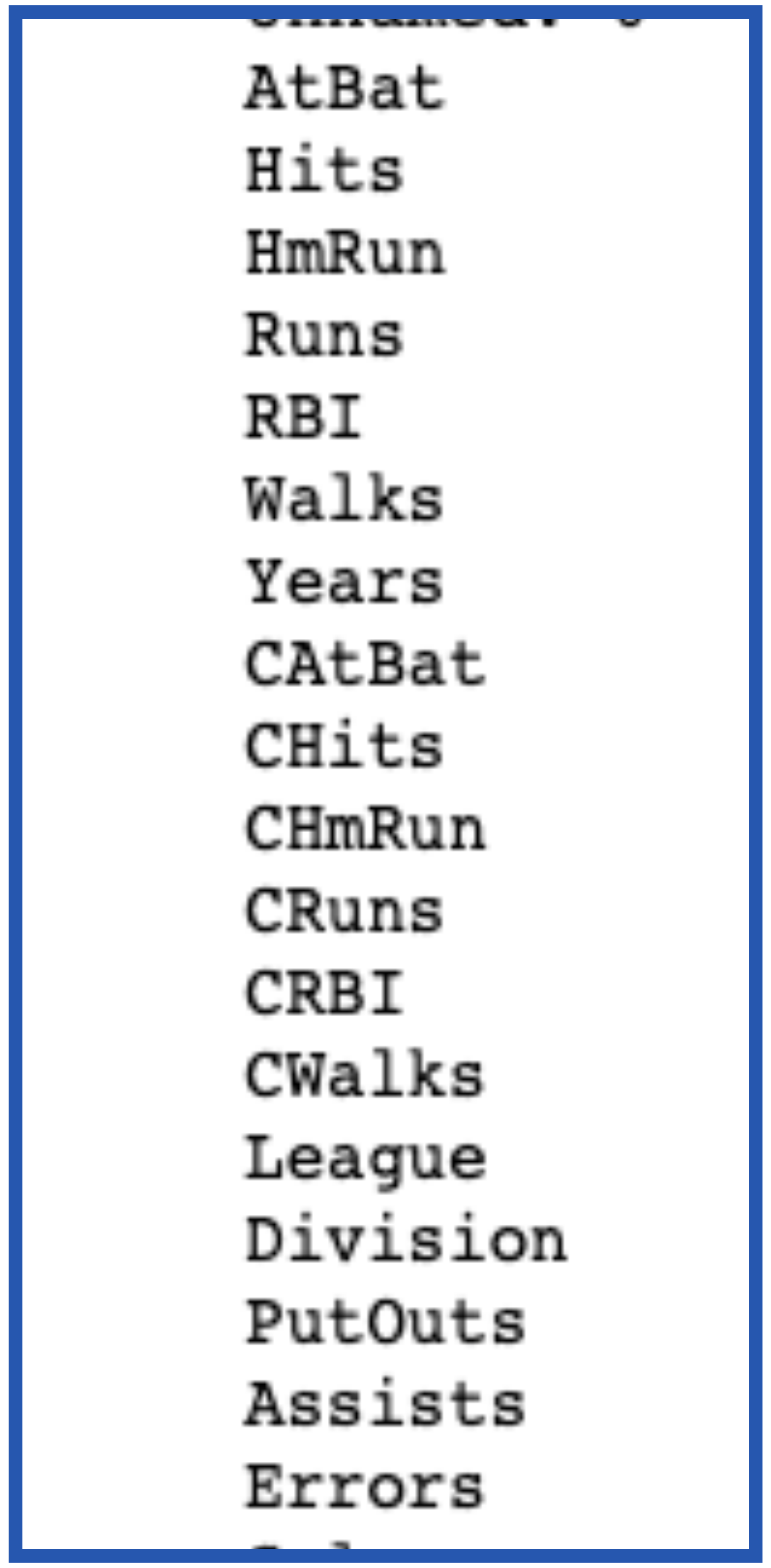


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set
- Step 3: Parameter $p \approx \sqrt{M}$ is determined
- Step 4: For data set *i*, randomly select p features.
- Step 5: Grow a tree and can only use the p features to branch
- Step 6: Back to Step 4 until B trees are built.

Random Forest

The point: randomly select features when branching

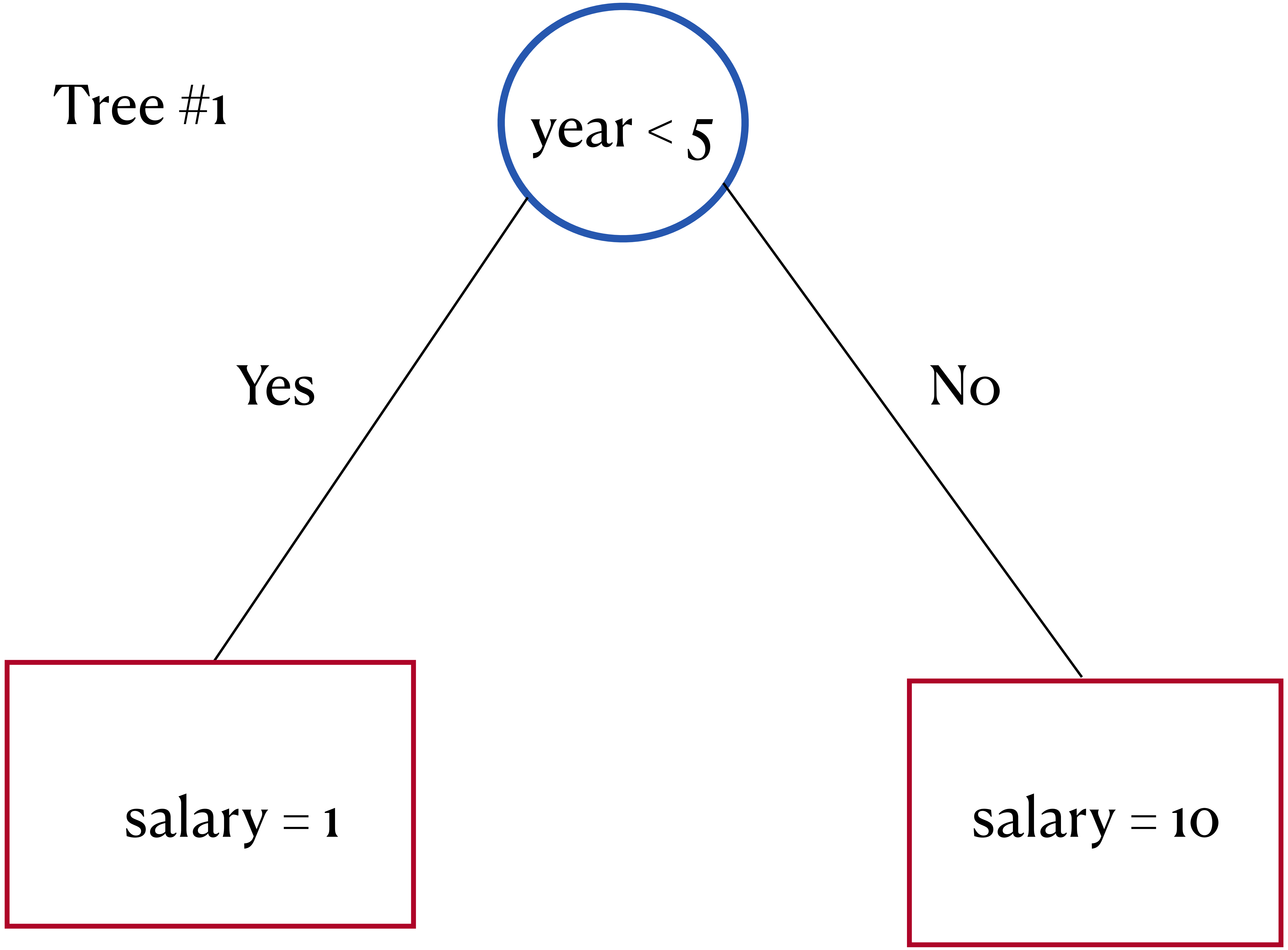


```
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
League
Division
PutOuts
Assists
Errors
```

- Step 1: generate N bootstrapped data sets
- Step 2: Observe there are M features in the data set
- Step 3: Parameter $p \approx \sqrt{M}$ is determined
- Step 4: For data set *i*, randomly select p features.
- Step 5: Grow a tree and can only use the p features to branch
- Step 6: Back to Step 4 until B trees are built.
- Step 7: For test sample x, the prediction is the average of B trees.

Built two simple trees as example with the baseball player data set

Tree #1



year < 5

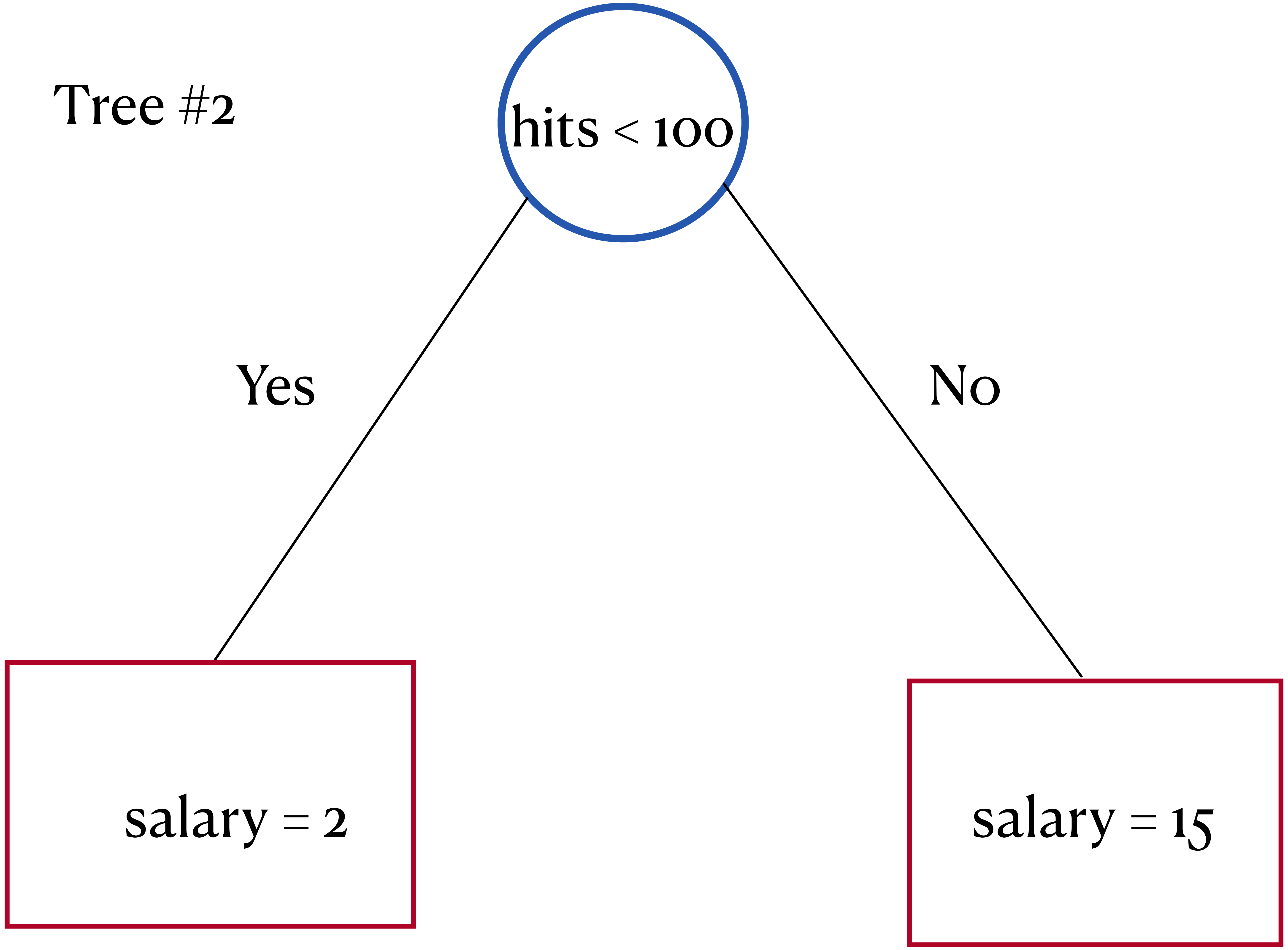
Yes

No

salary = 1

salary = 10

Tree #2



hits < 100

Yes

No

salary = 2

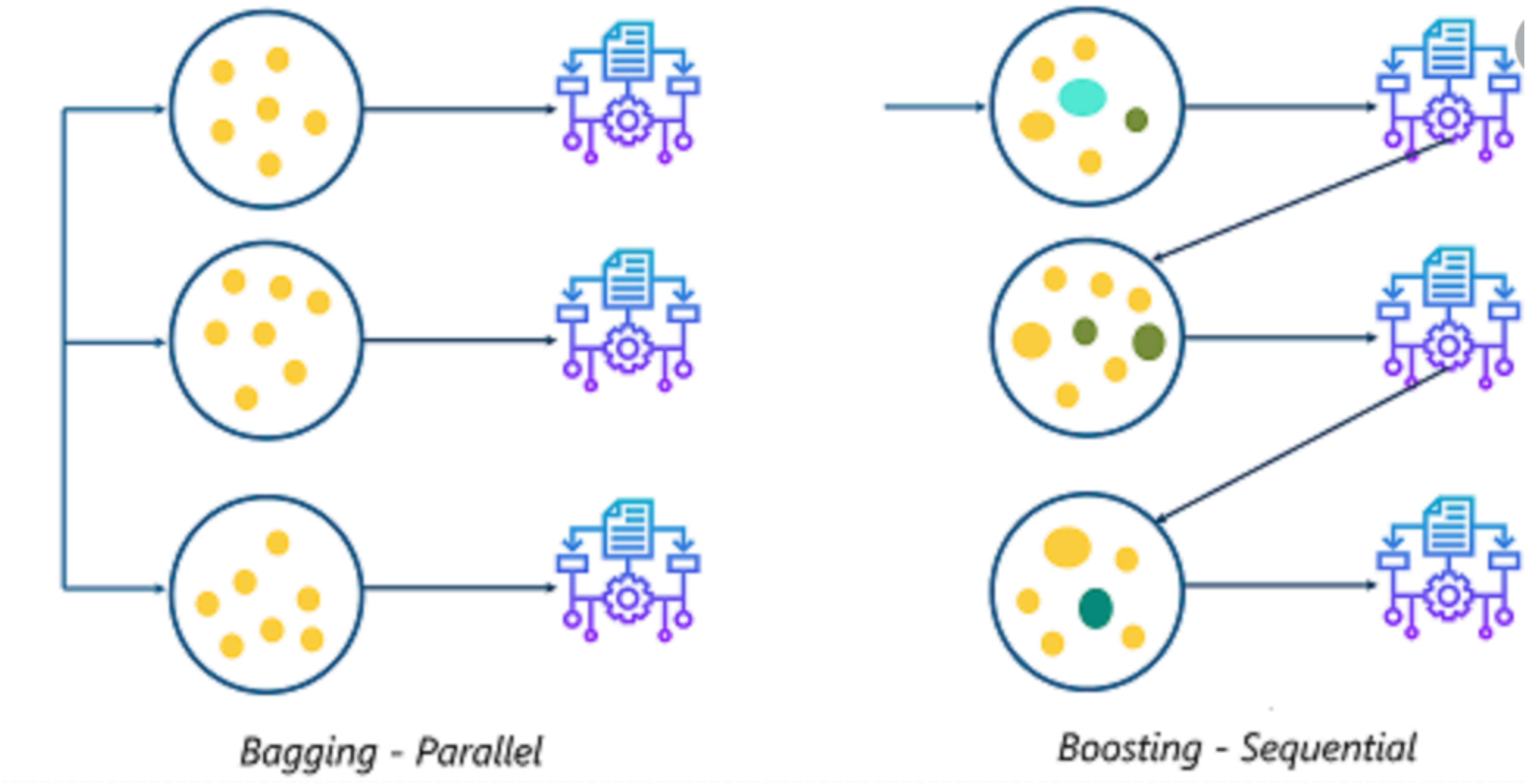
salary = 15

Suppose we have the following player's data, we can predict their salary based on the two trees.

Player Name	year	hits	salary
A	2	20	(1+2)/2 = 1.5
B	4	120	(1+15)/2 = 8
C	6	10	(10+2)/2 = 6
D	7	150	(10+15)/2 = 12.5

Boosting: A sequence of trees

Collection of weak learners become a strong learner

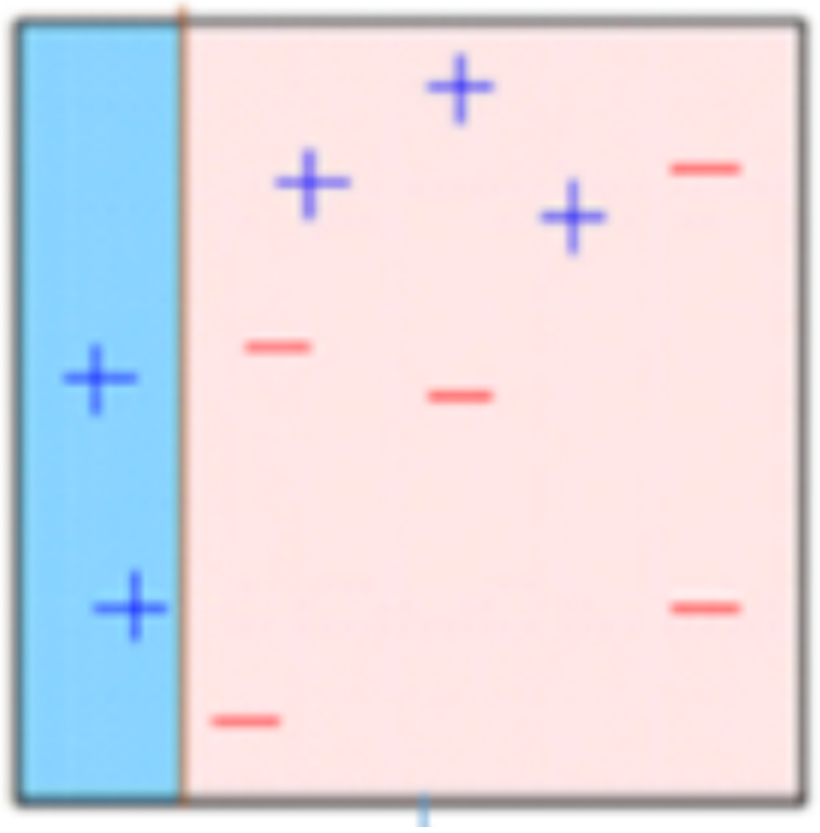


Classification

Tree 1

Tree 2

Tree 3

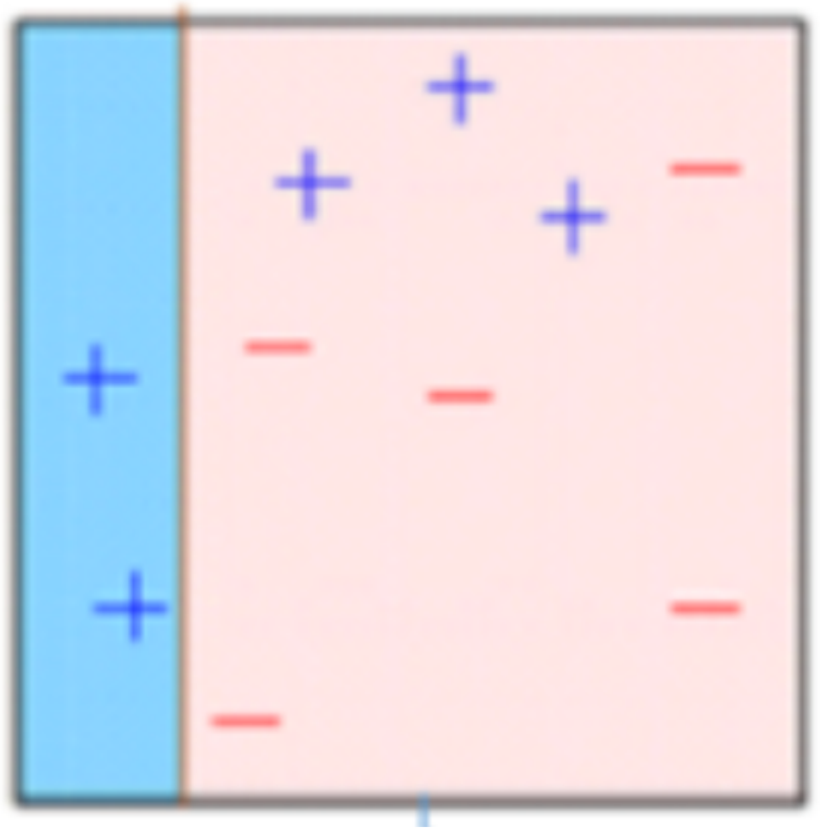


Final vote

https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Classification

Tree 1



# of incorrect: 3

Tree 2

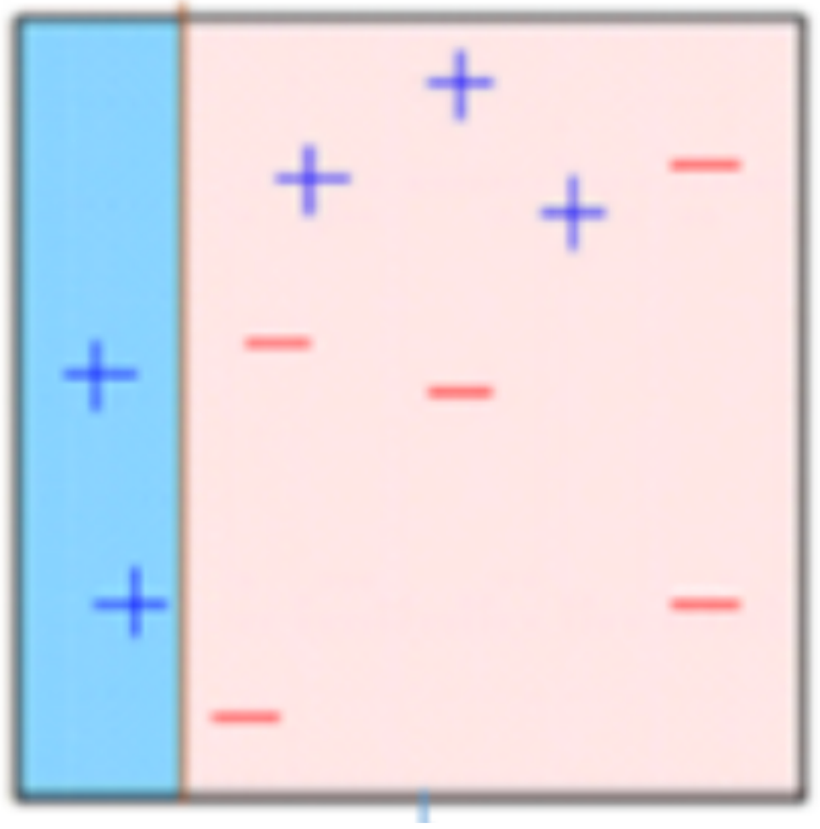
Tree 3

Final vote

https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

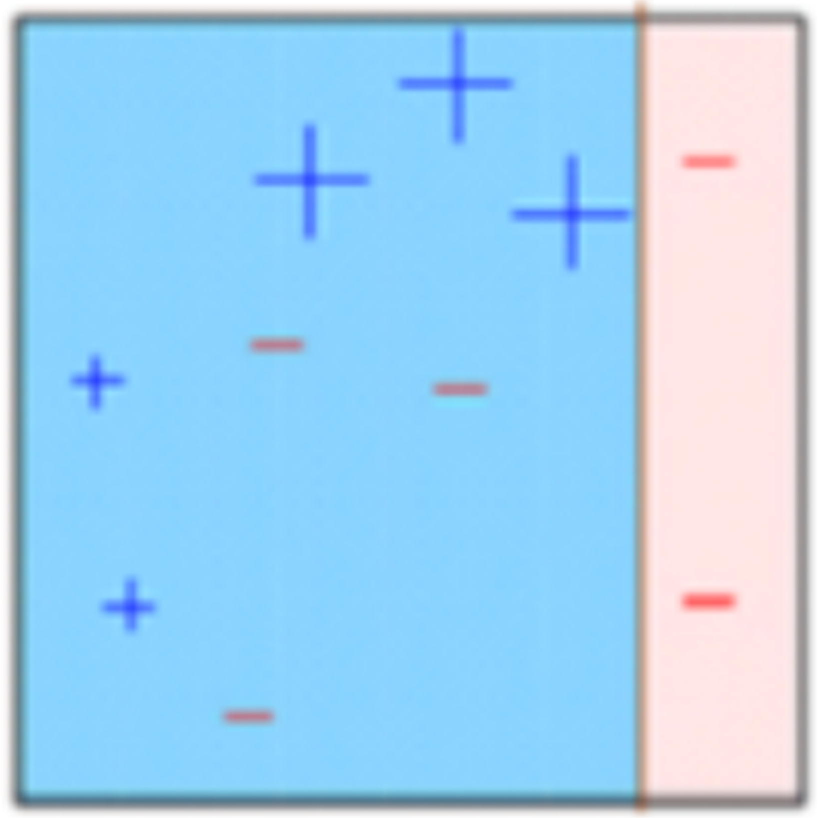
Classification

Tree 1



# of incorrect: 3

Tree 2



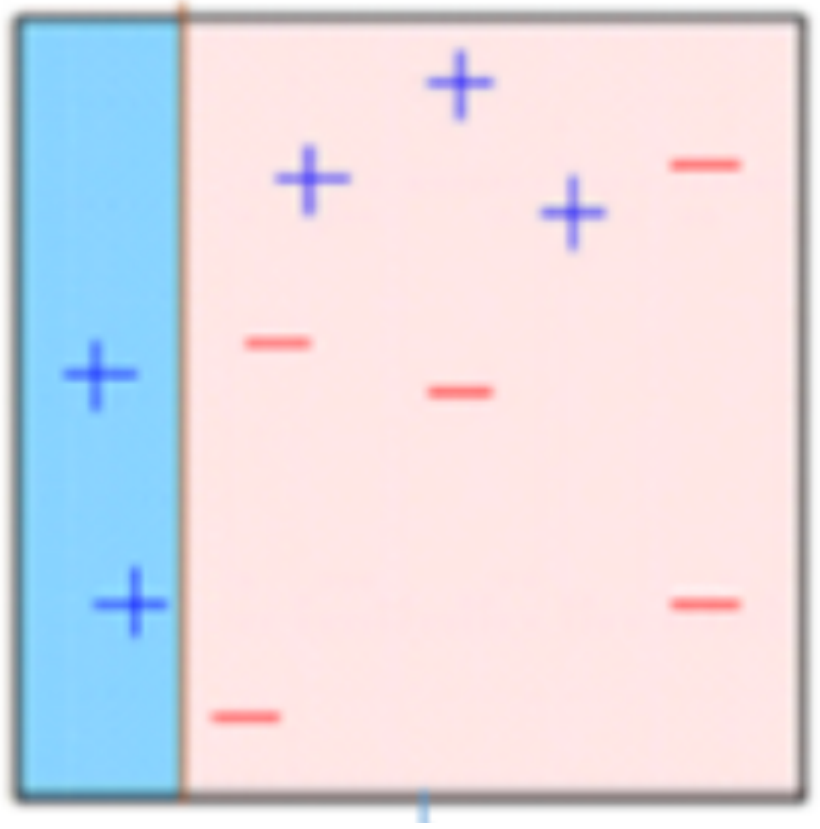
Tree 3

Final vote

https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

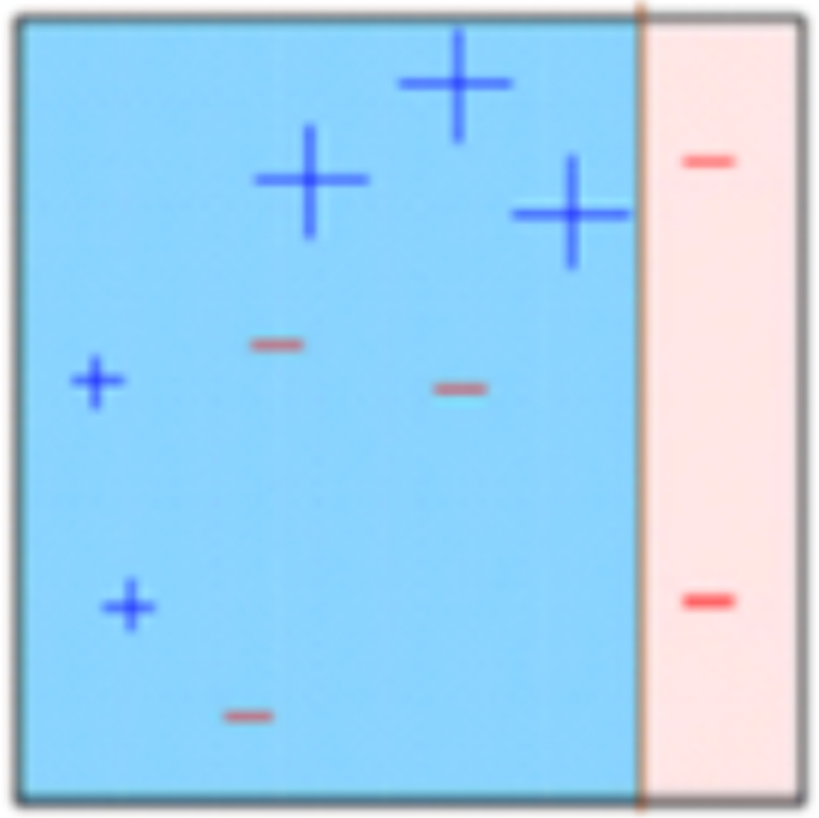
Classification

Tree 1



# of incorrect: 3

Tree 2



# of incorrect: 3

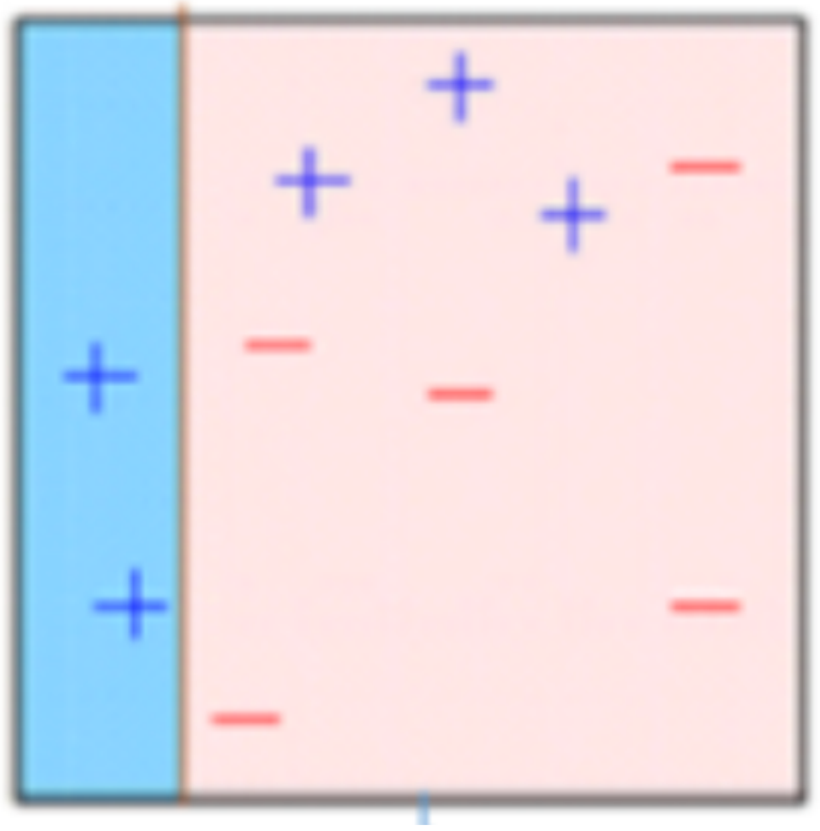
Tree 3

Final vote

https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

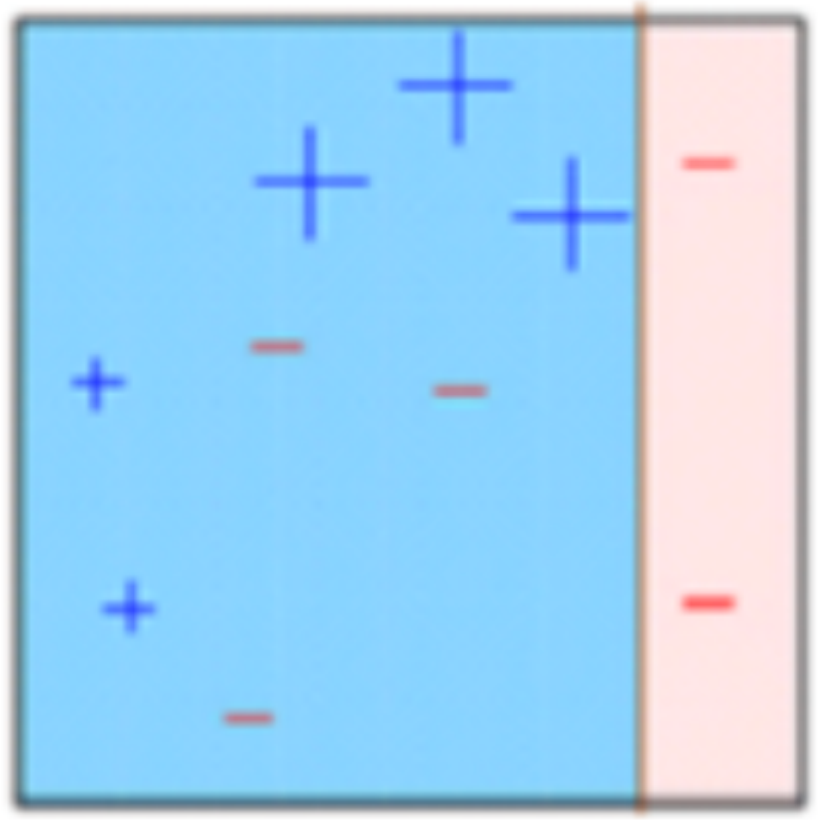
Classification

Tree 1



# of incorrect: 3

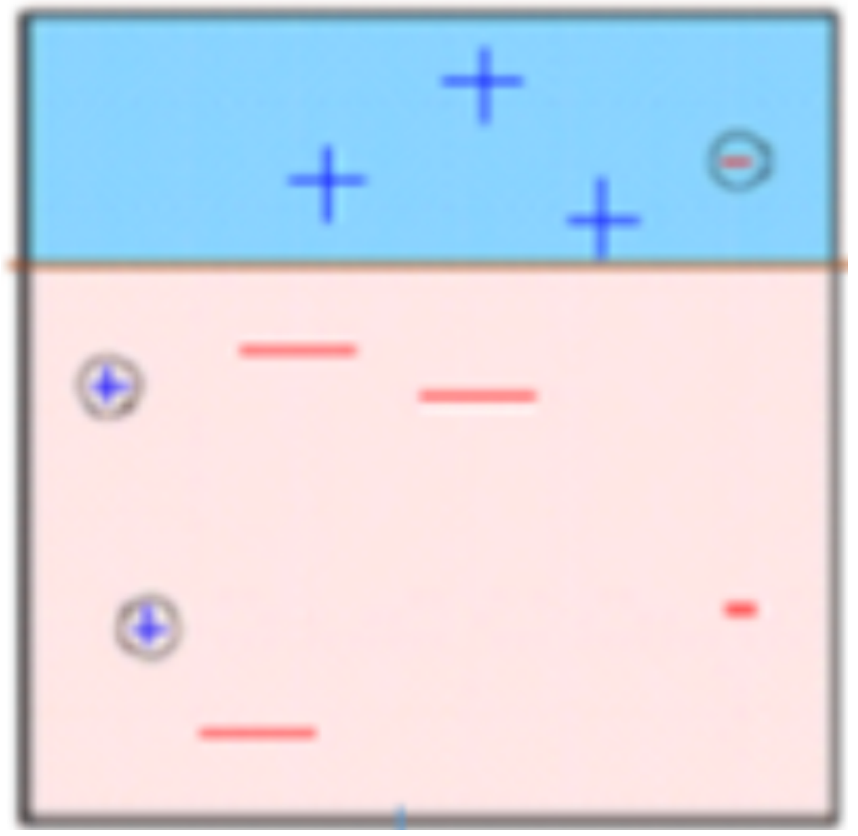
Tree 2



# of incorrect: 3

Final vote

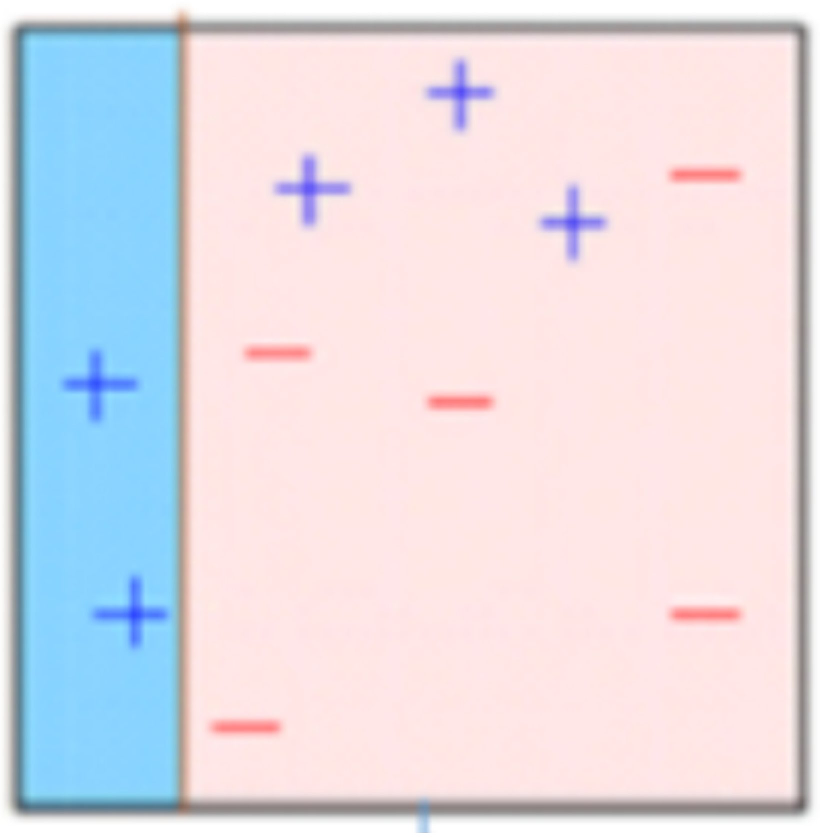
Tree 3



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

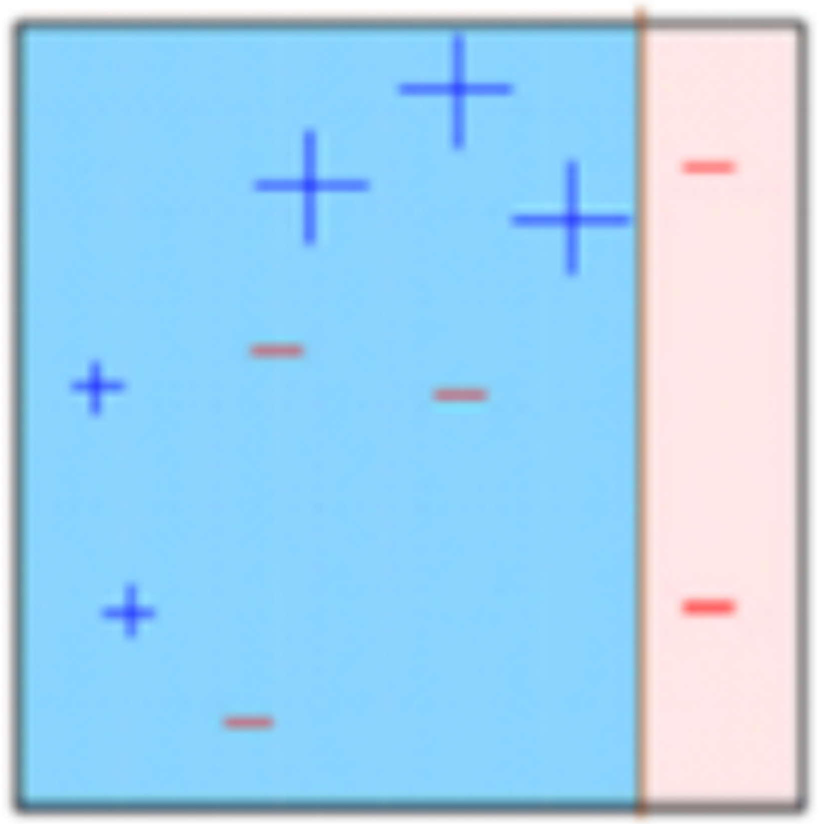
Classification

Tree 1



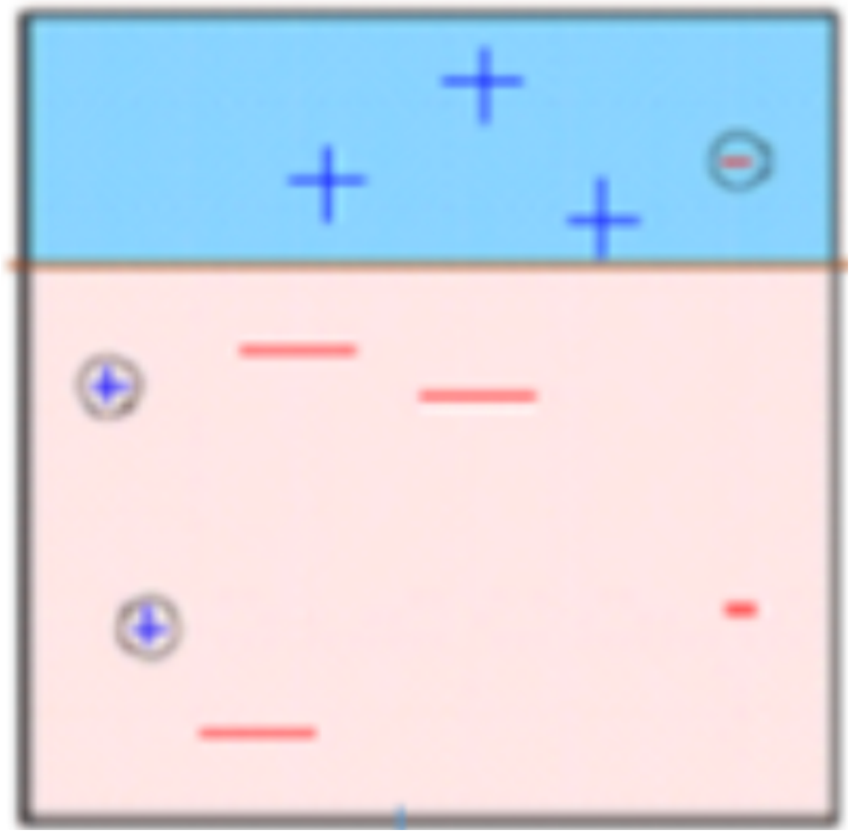
# of incorrect: 3

Tree 2



# of incorrect: 3

Tree 3



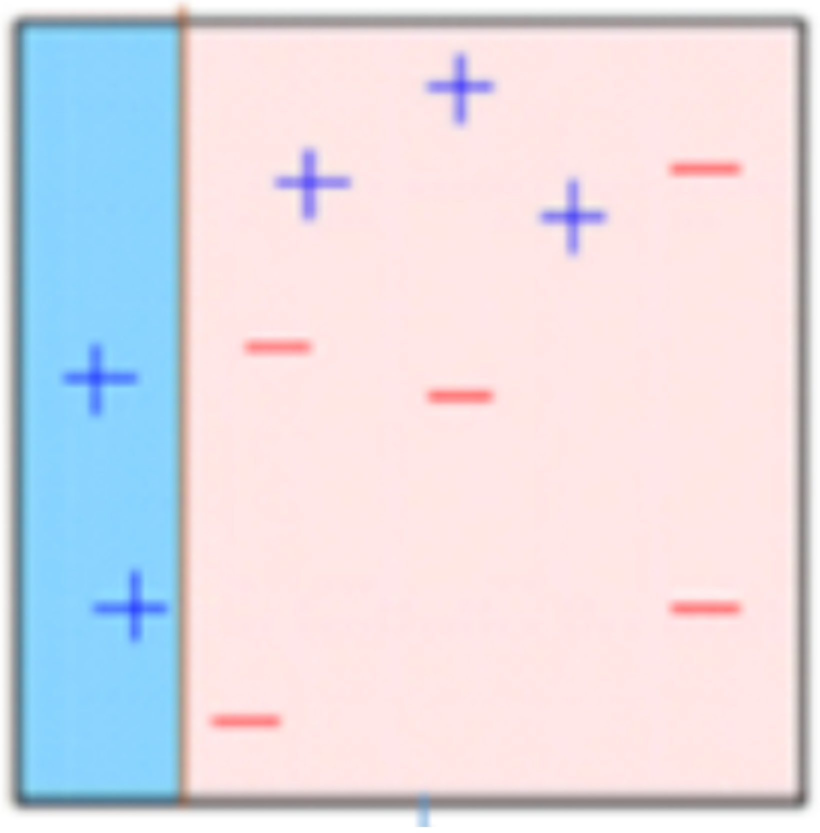
# of incorrect: 3

Final vote

https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

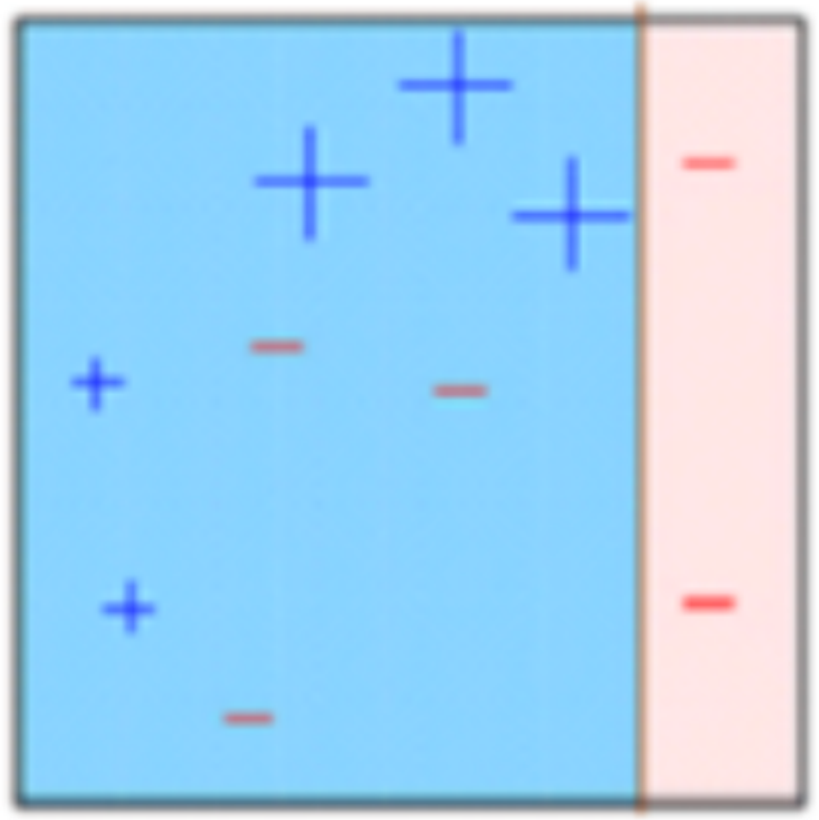
Classification

Tree 1



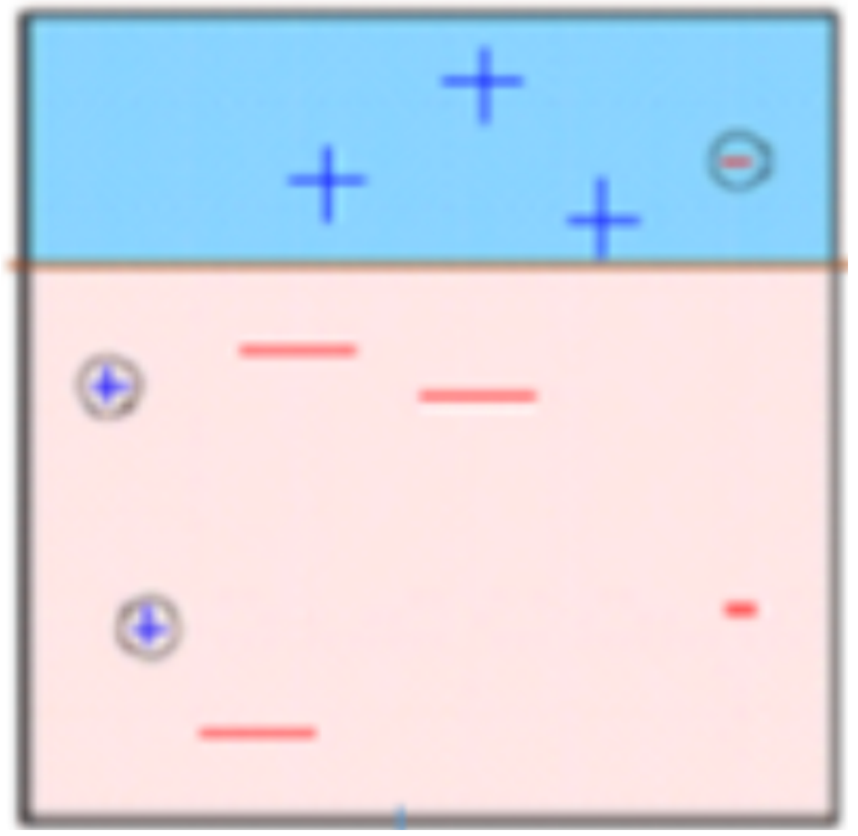
# of incorrect: 3

Tree 2



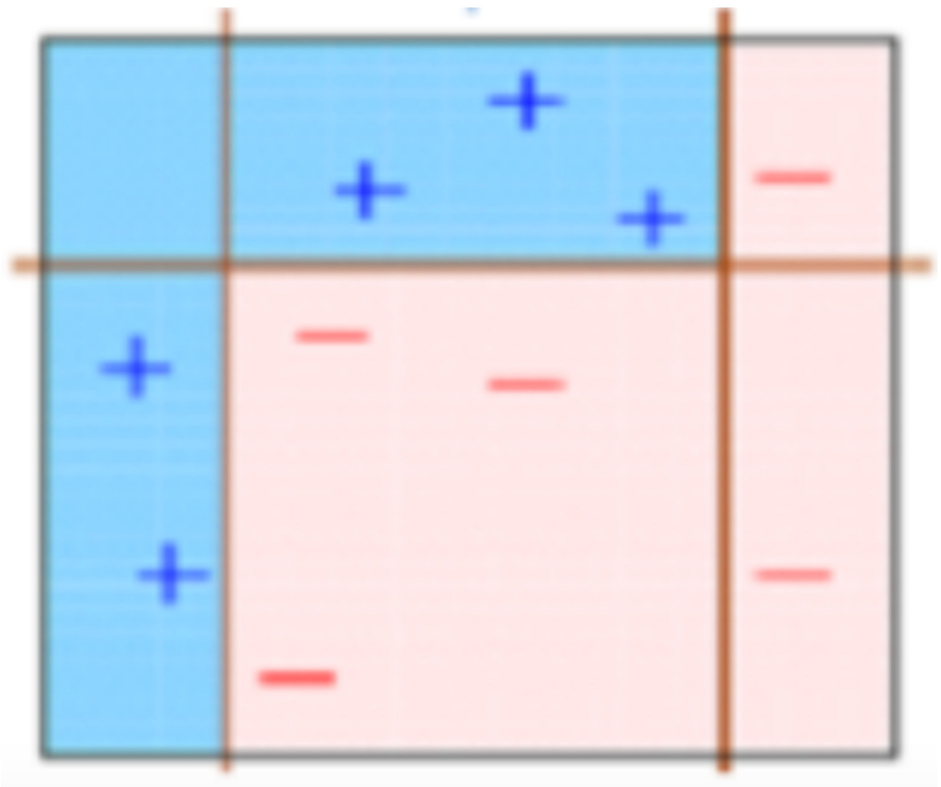
# of incorrect: 3

Tree 3



# of incorrect: 3

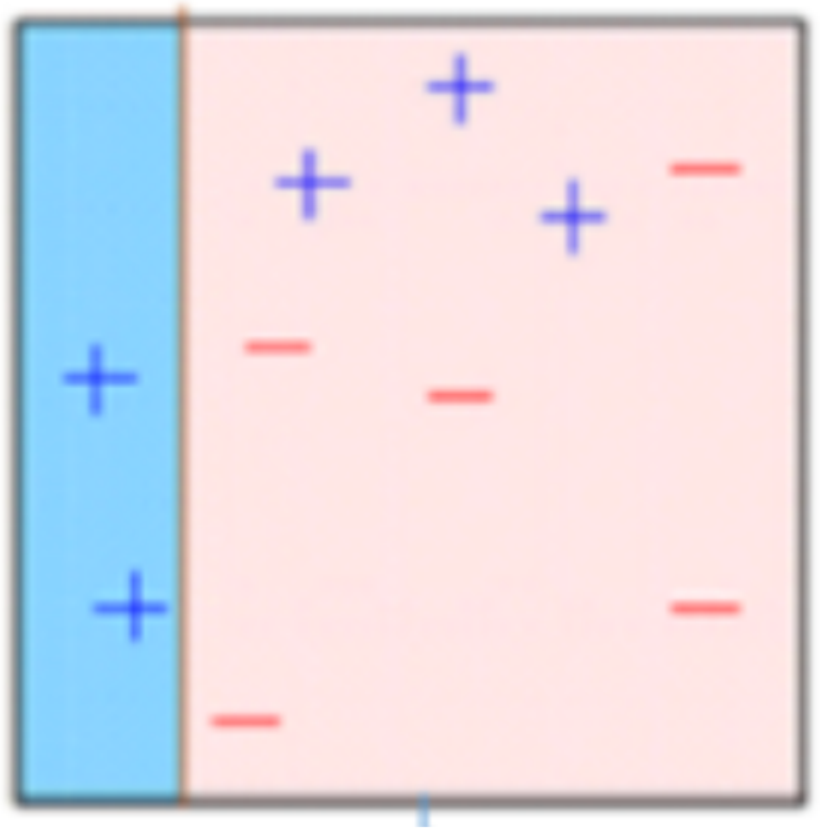
Final vote



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

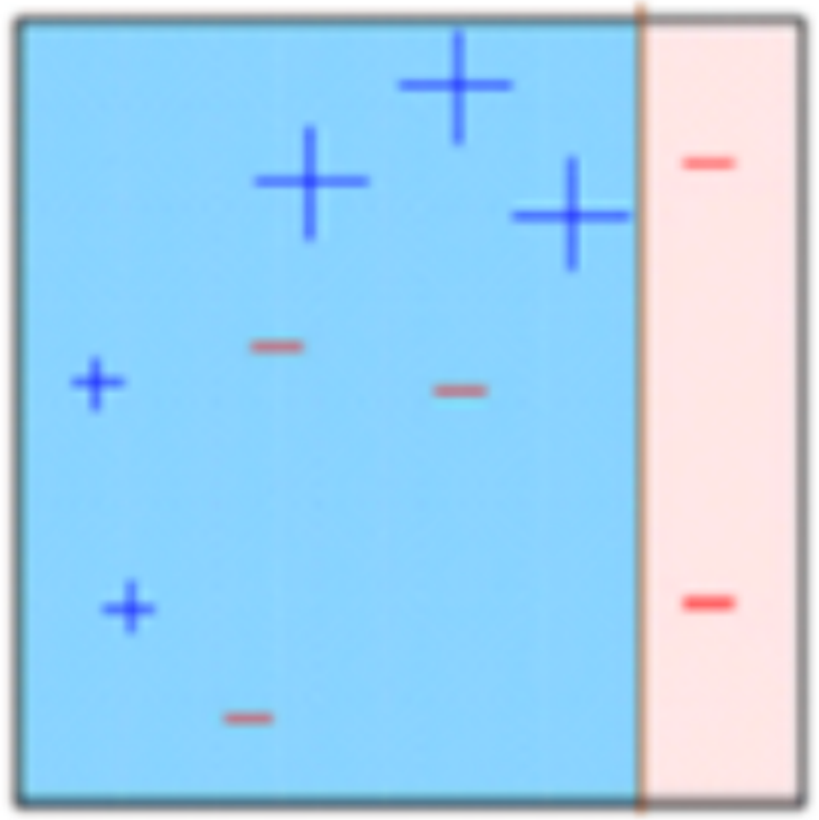
Classification

Tree 1



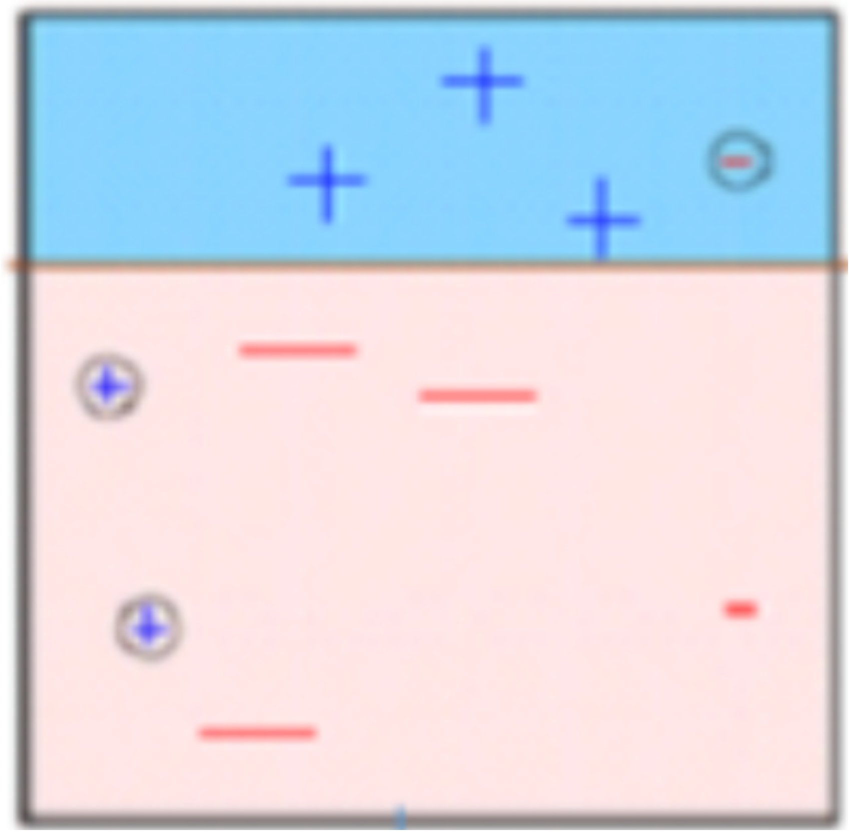
# of incorrect: 3

Tree 2



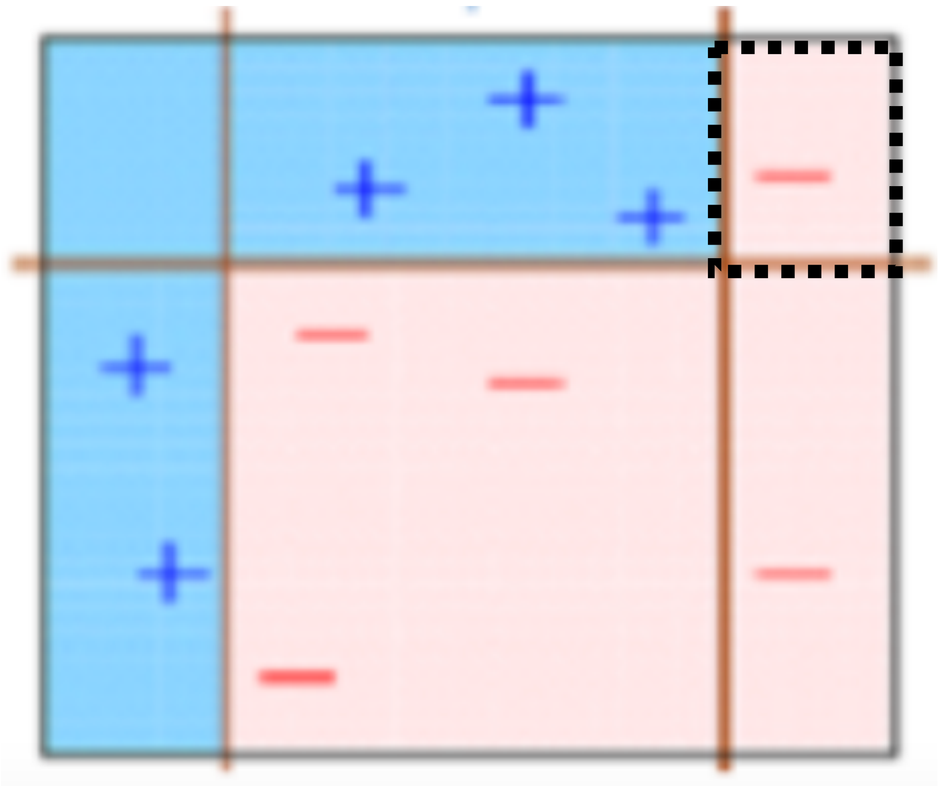
# of incorrect: 3

Tree 3



# of incorrect: 3

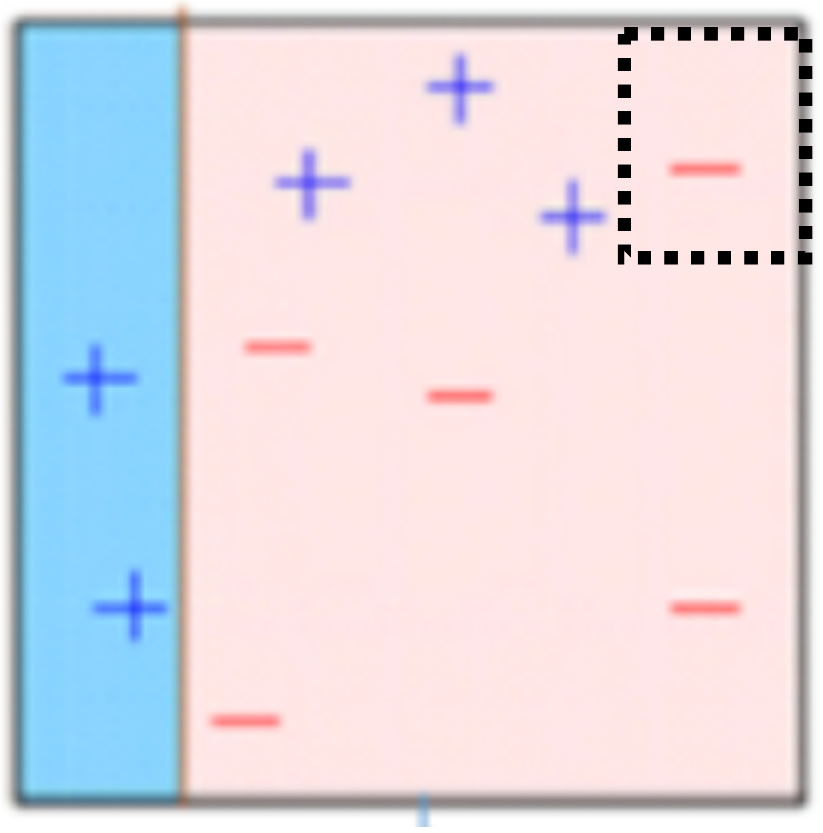
Final vote



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

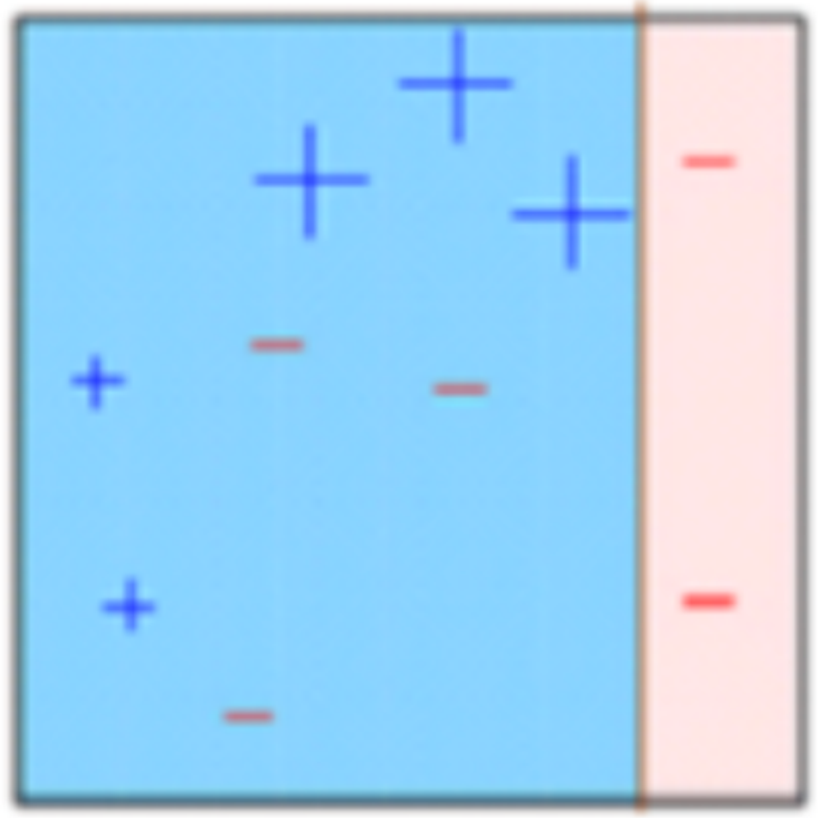
Classification

Tree 1



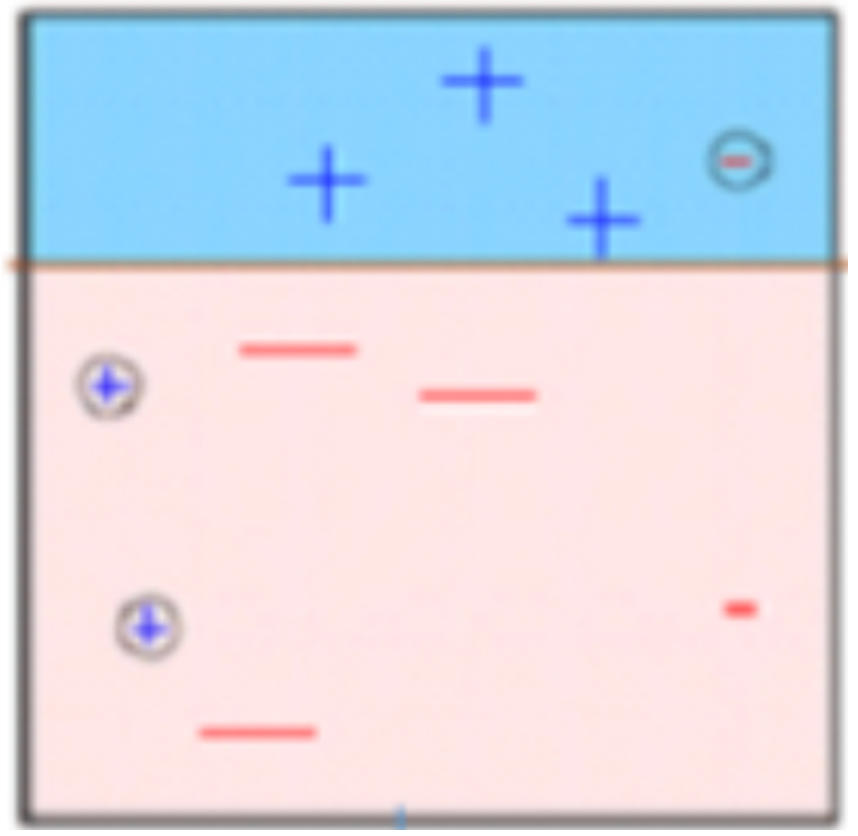
# of incorrect: 3

Tree 2



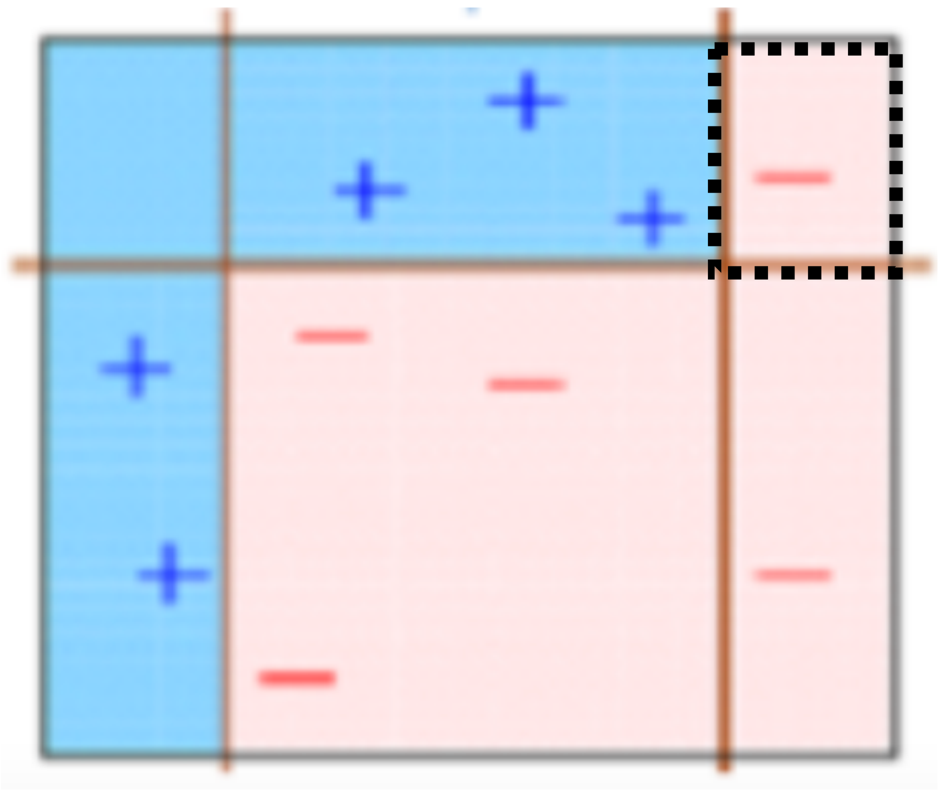
# of incorrect: 3

Tree 3



# of incorrect: 3

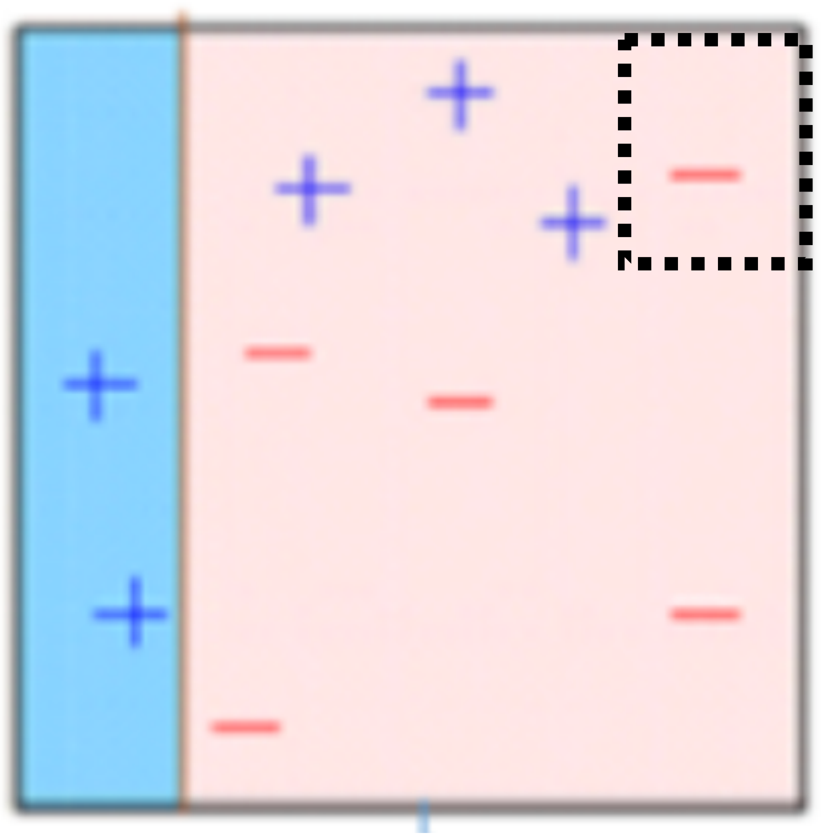
Final vote



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

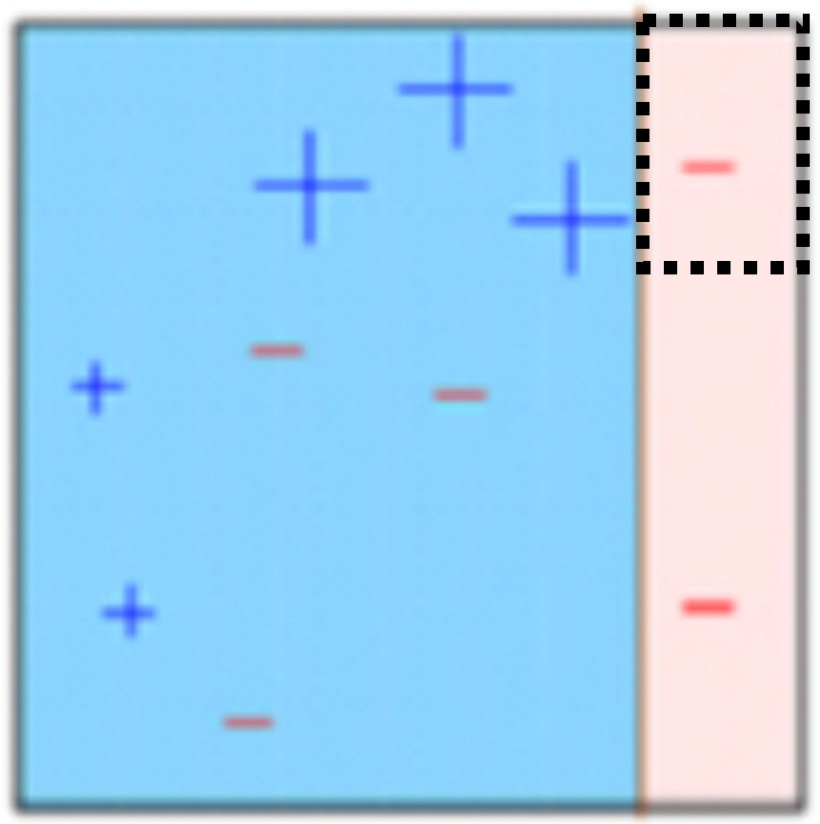
Classification

Tree 1



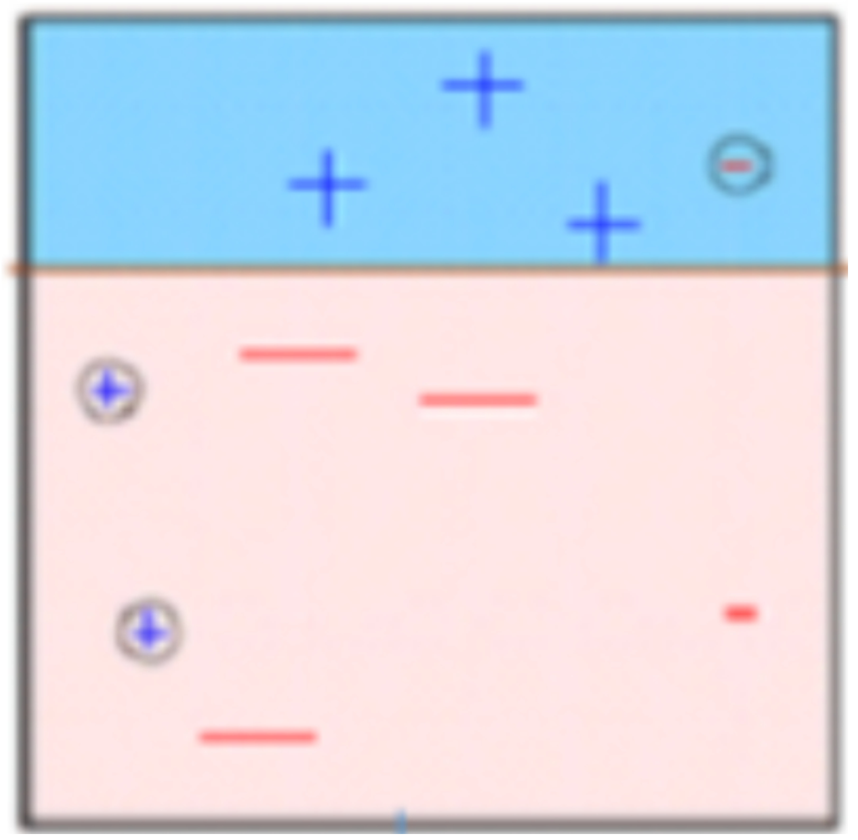
# of incorrect: 3

Tree 2



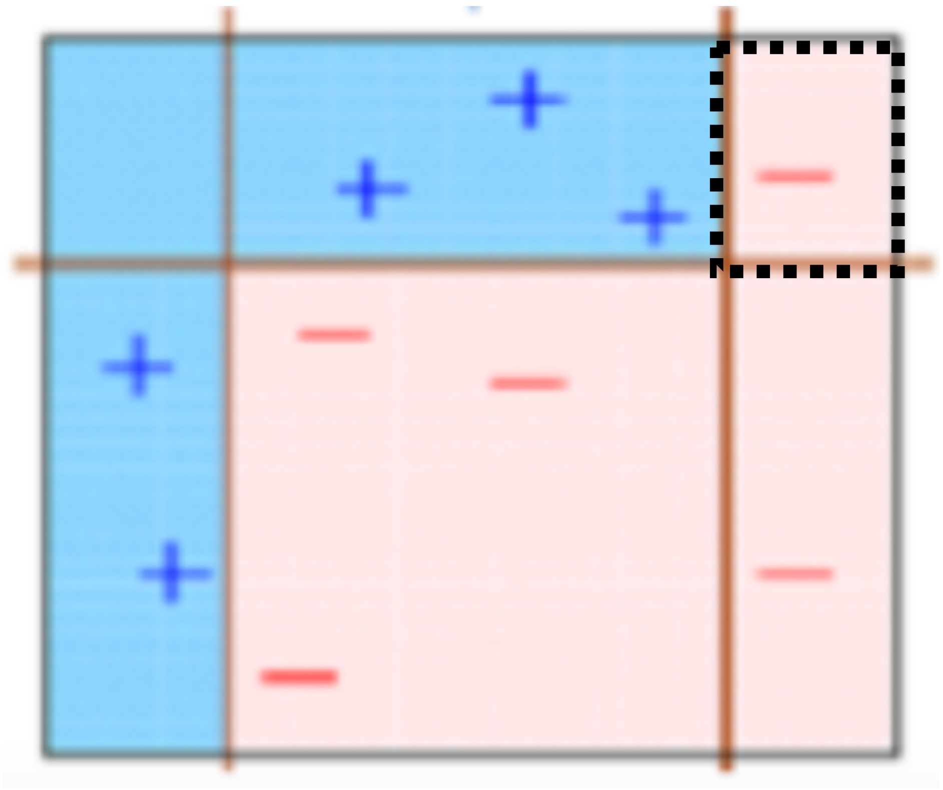
# of incorrect: 3

Tree 3



# of incorrect: 3

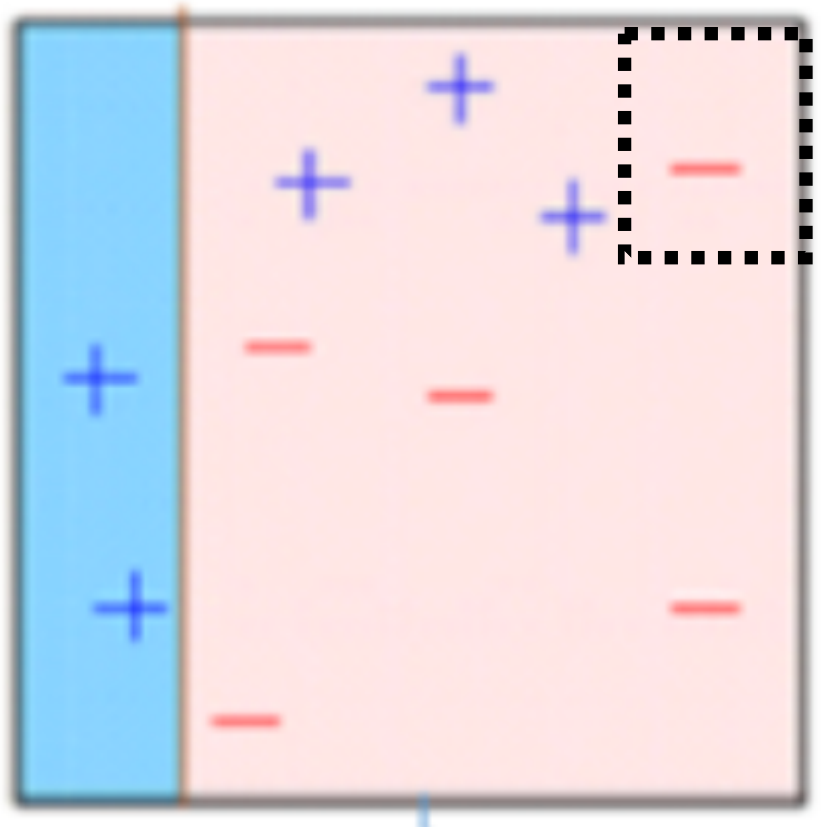
Final vote



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

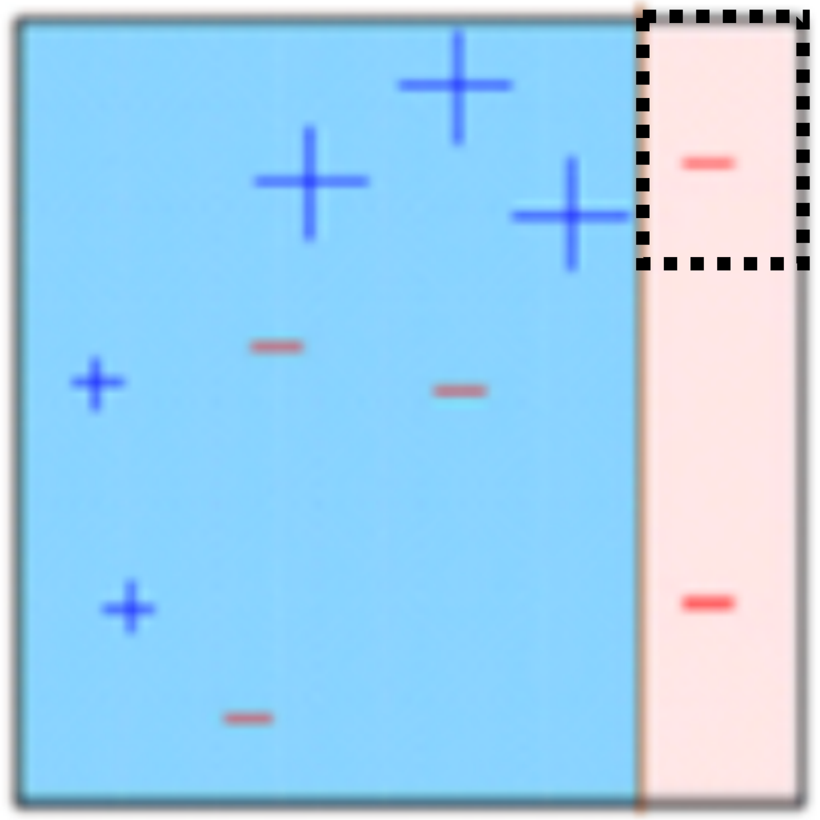
Classification

Tree 1



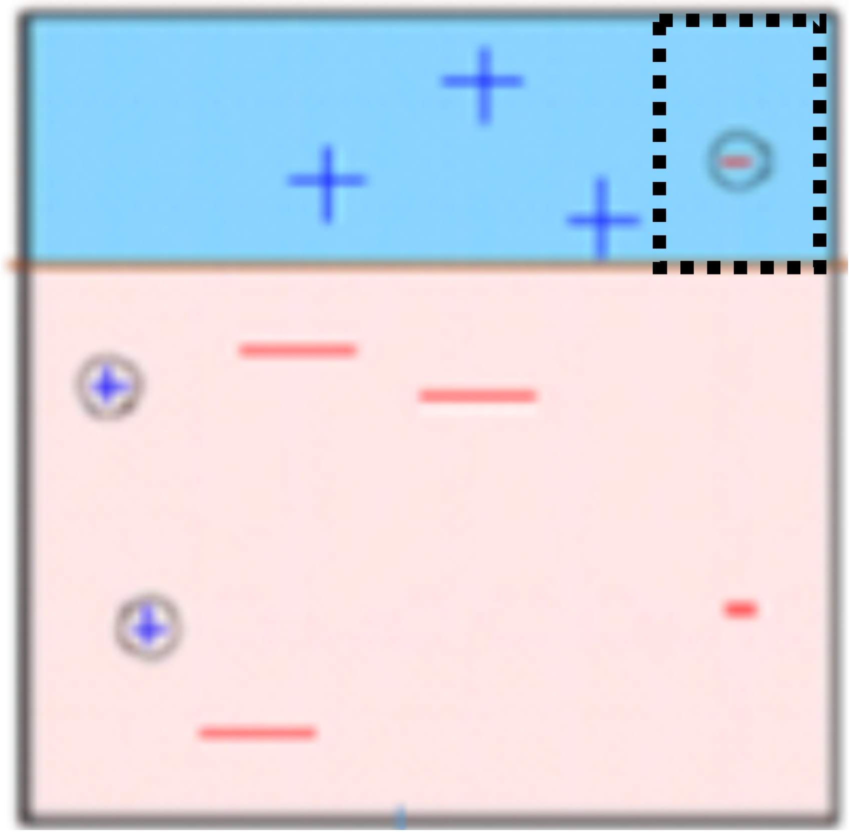
# of incorrect: 3

Tree 2



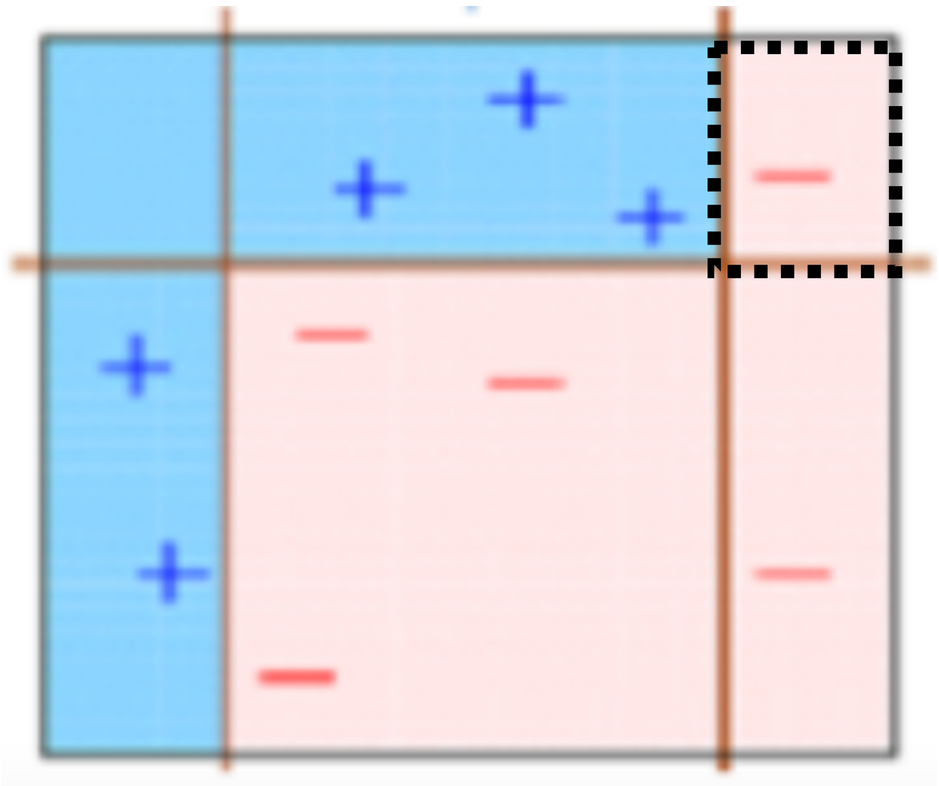
# of incorrect: 3

Tree 3



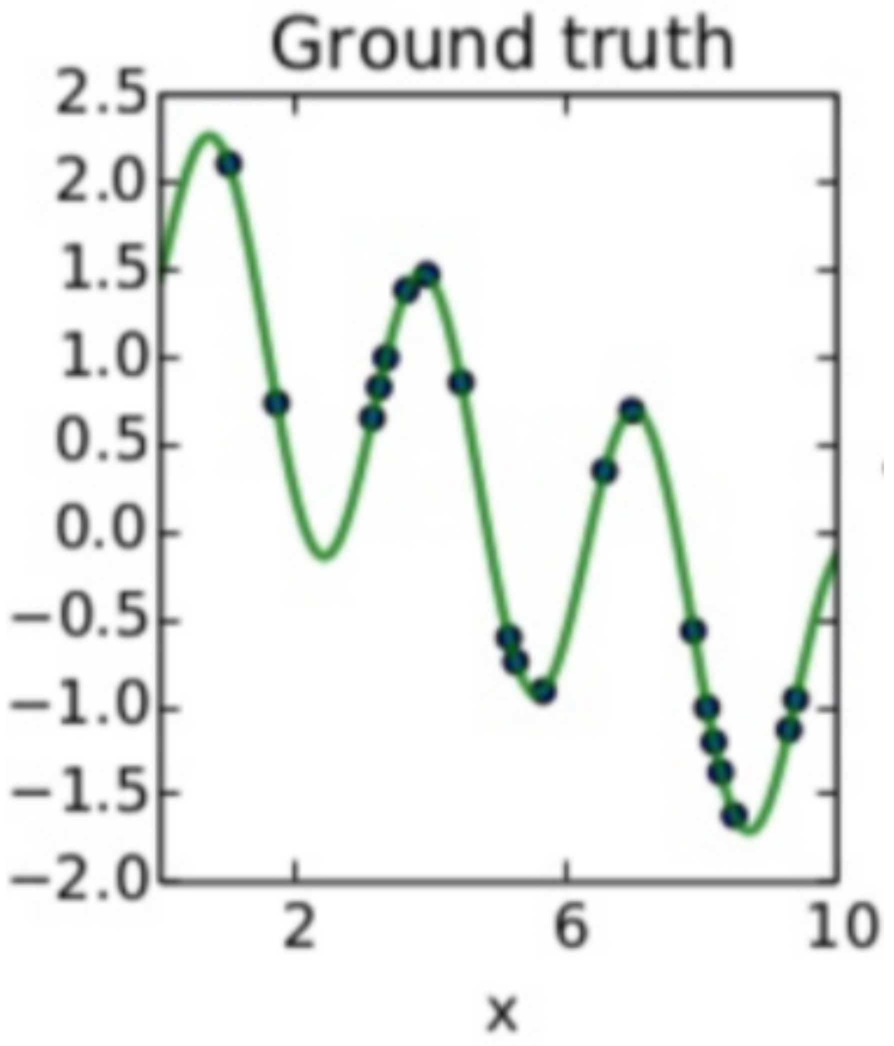
# of incorrect: 3

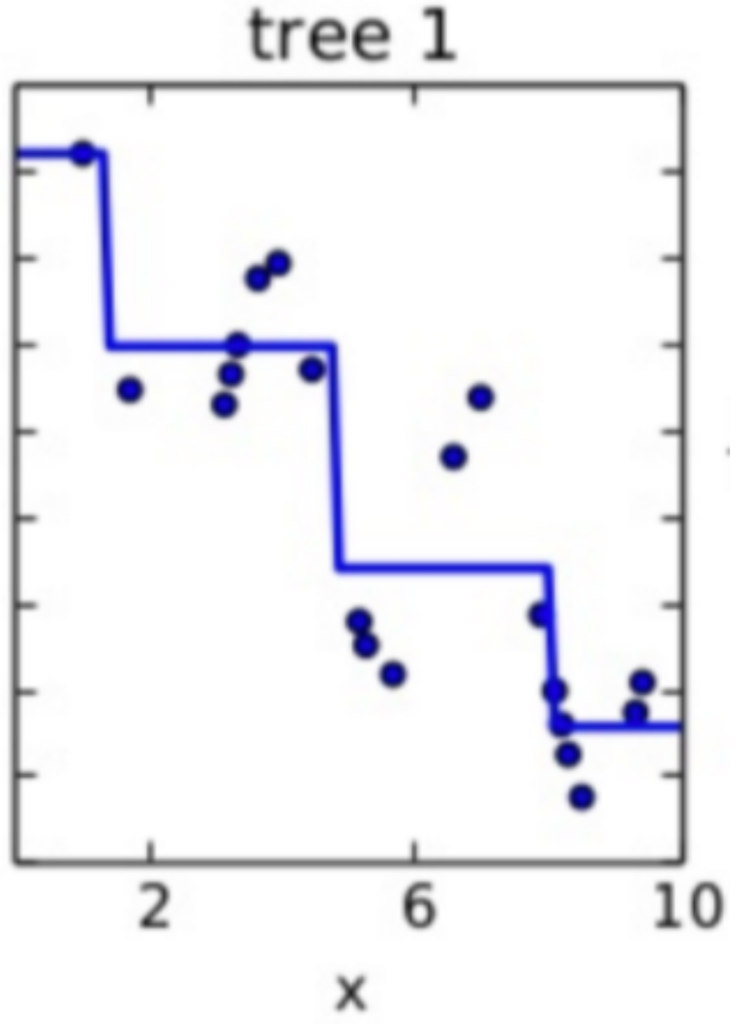
Final vote



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

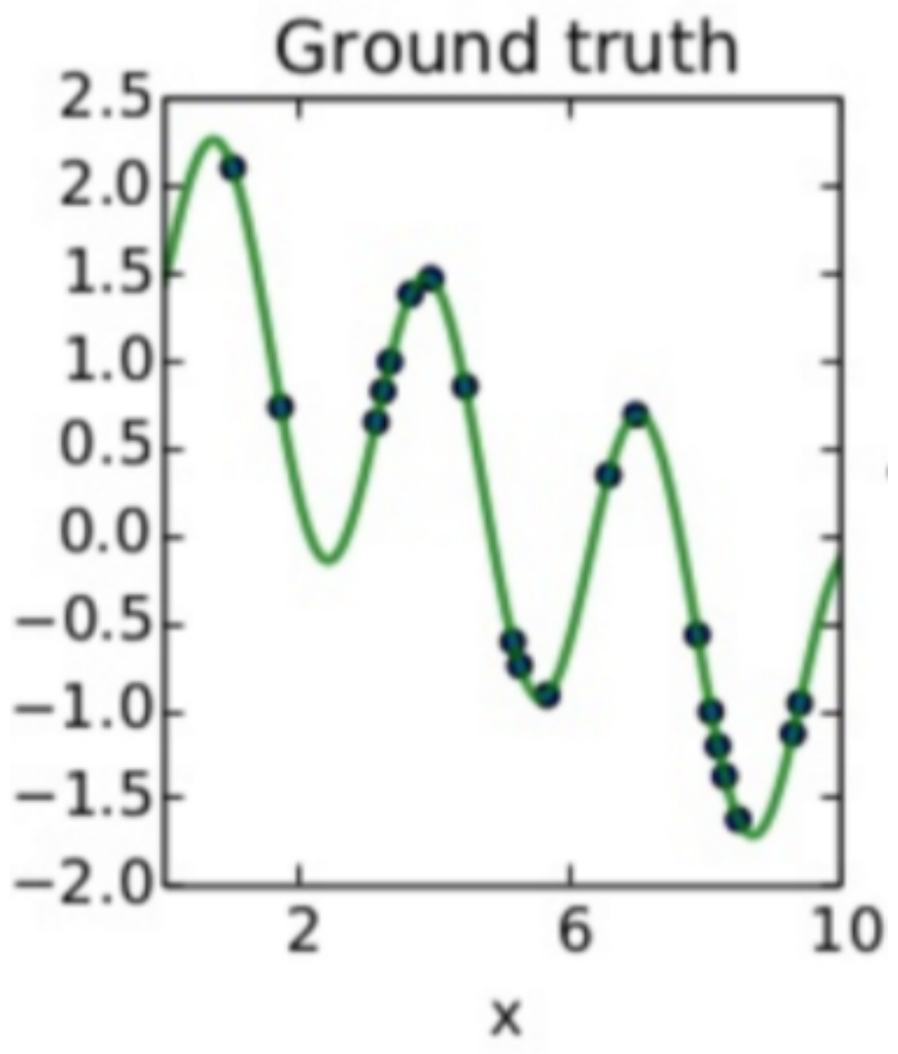
Regression

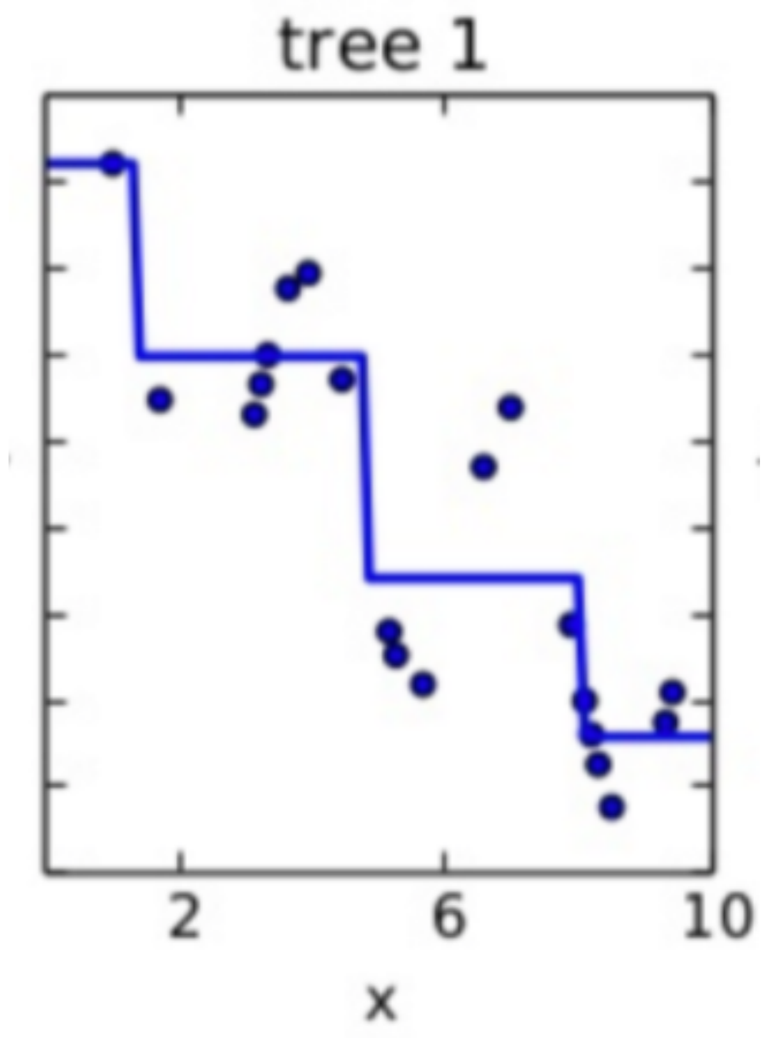


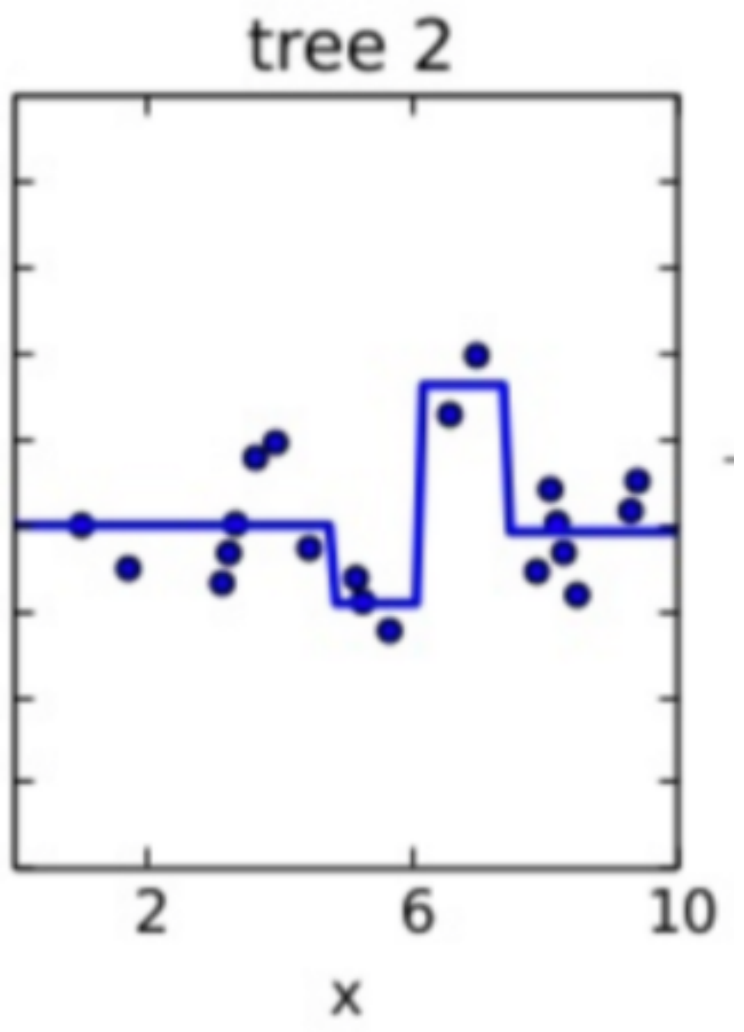


https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression

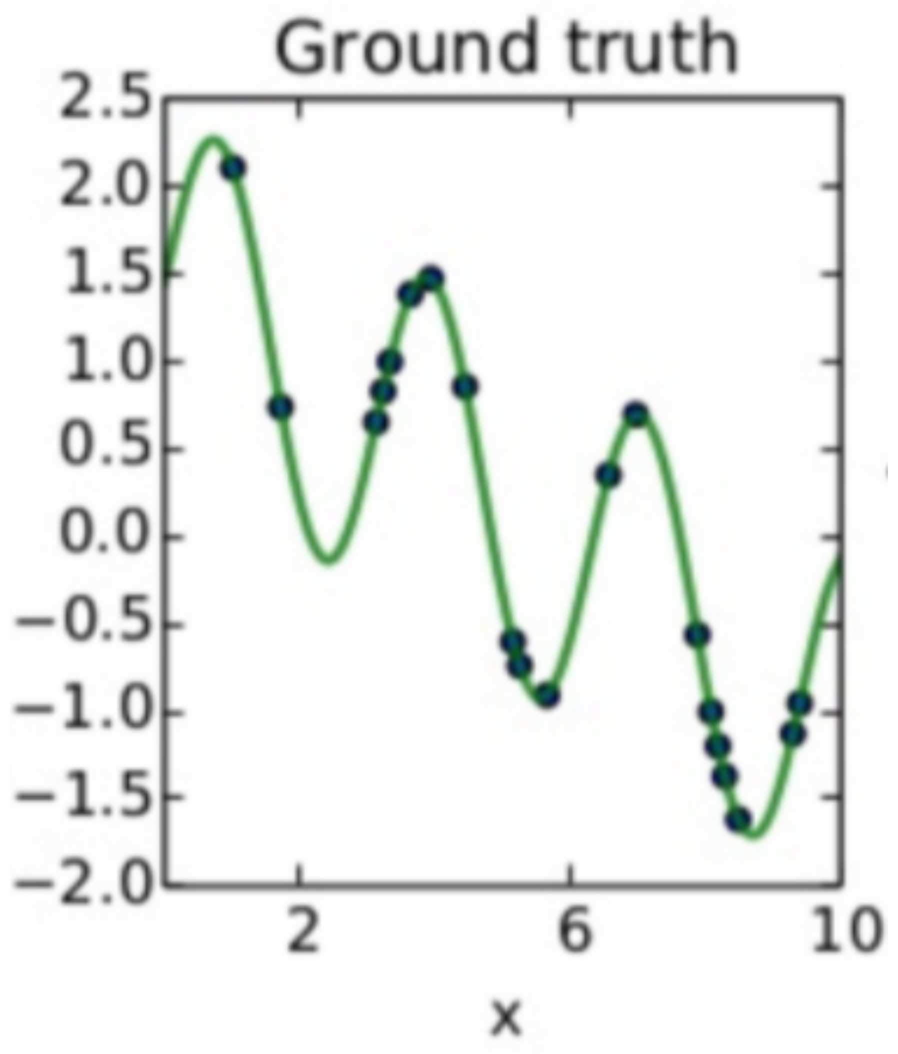


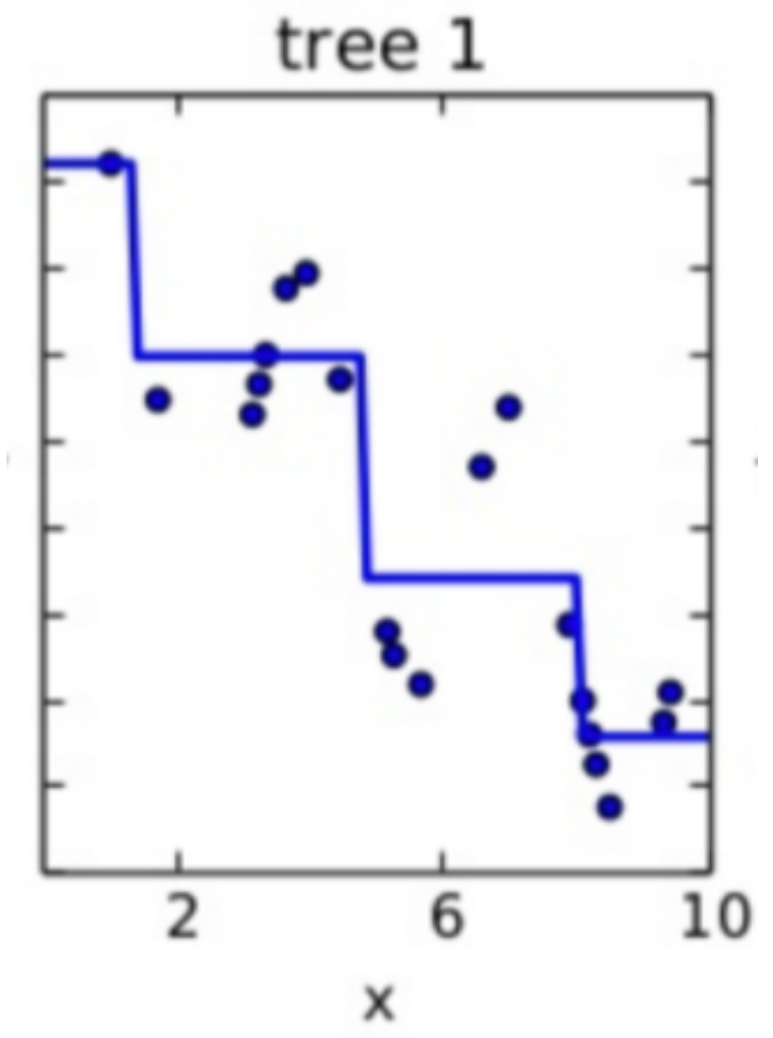


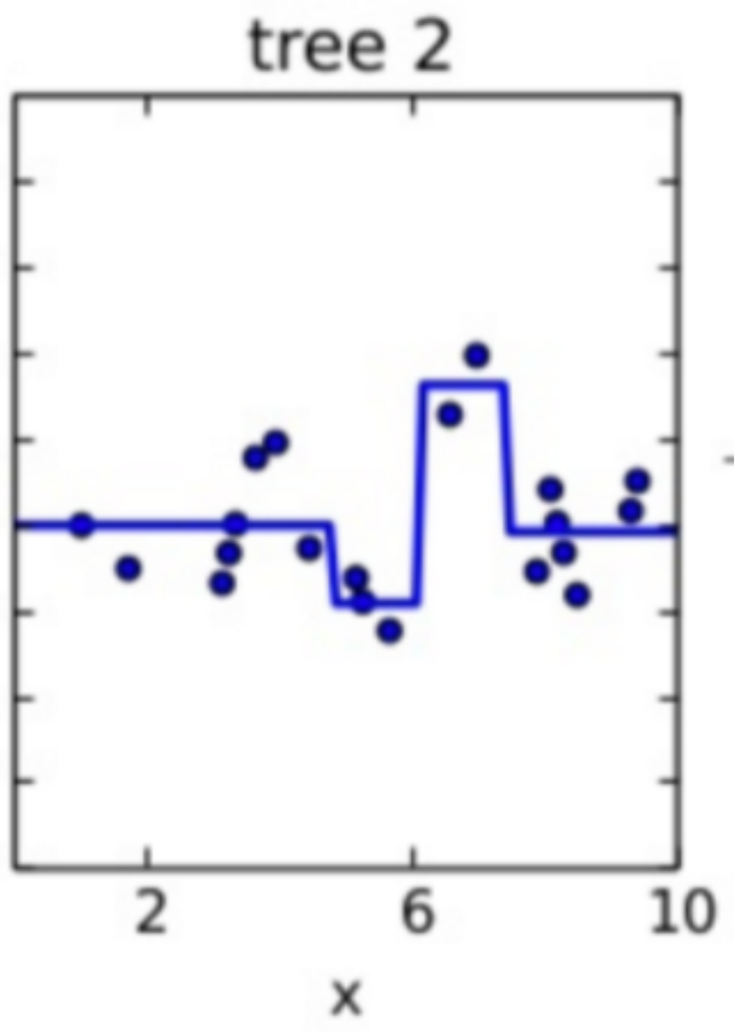


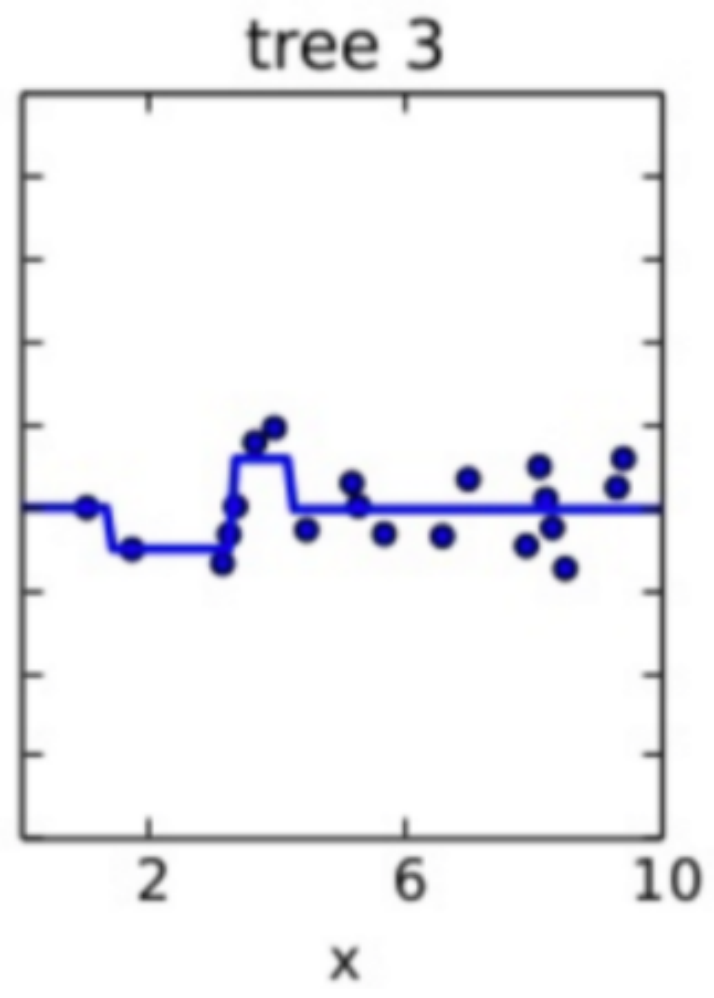
https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression



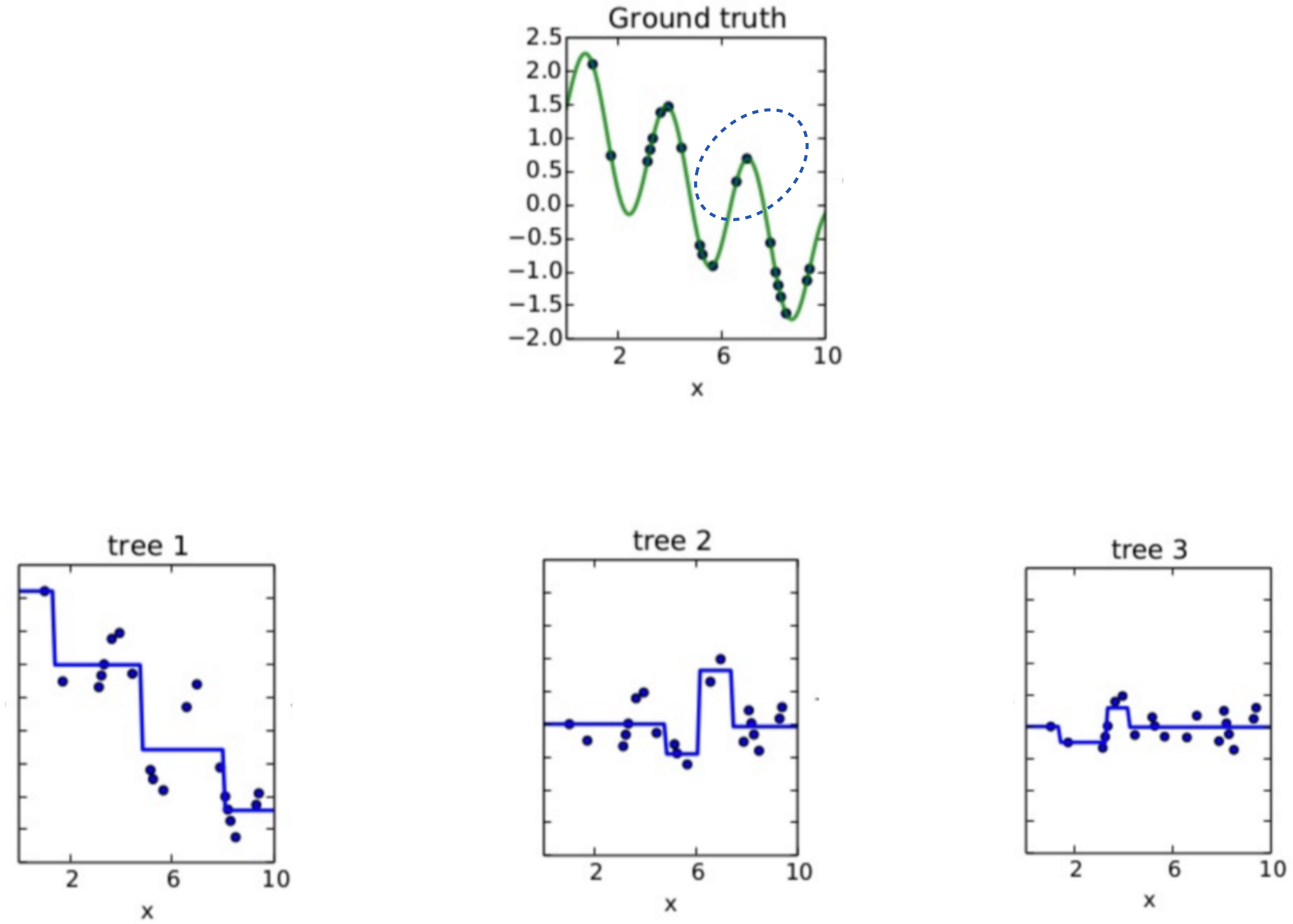






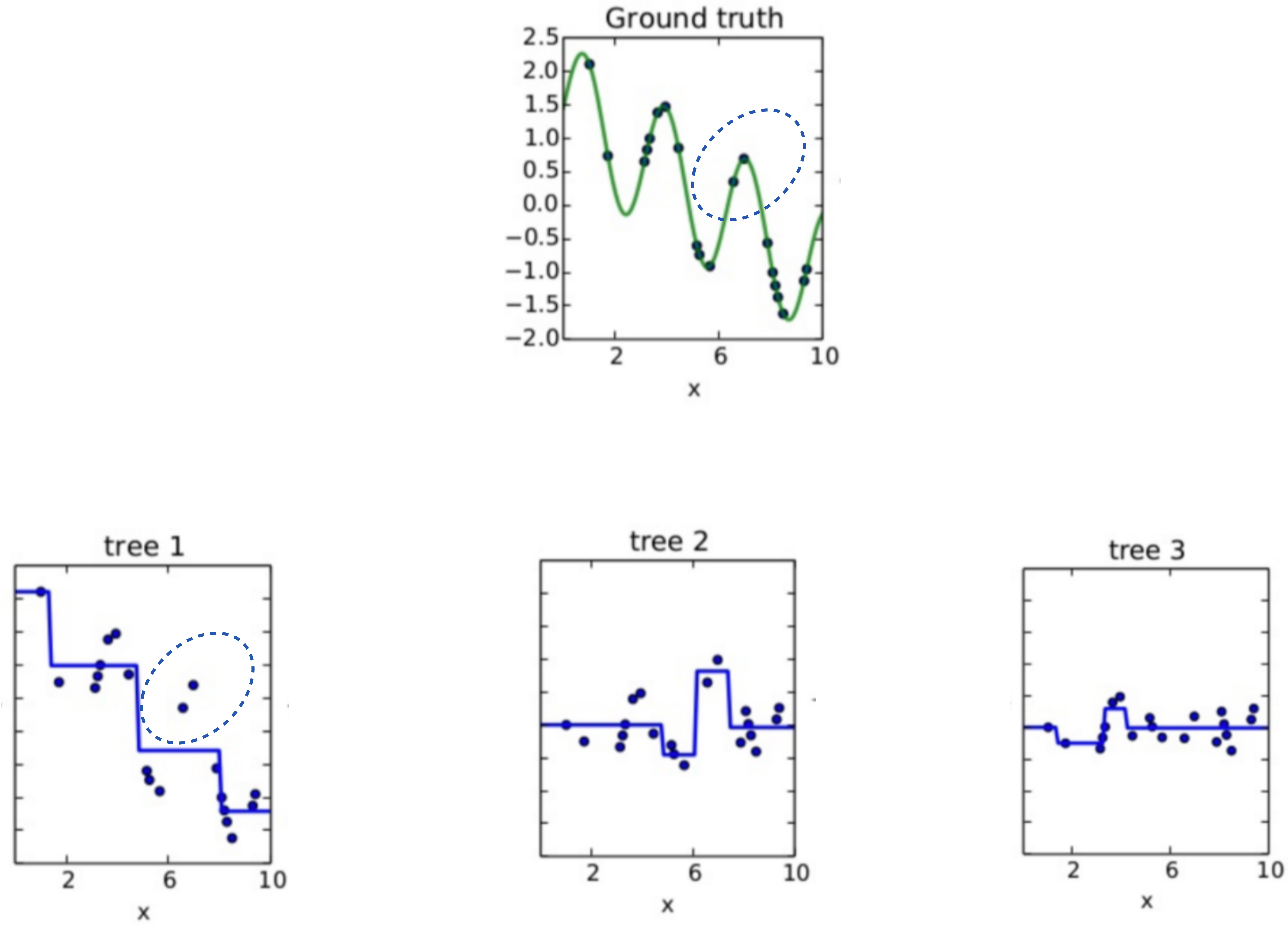
https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression



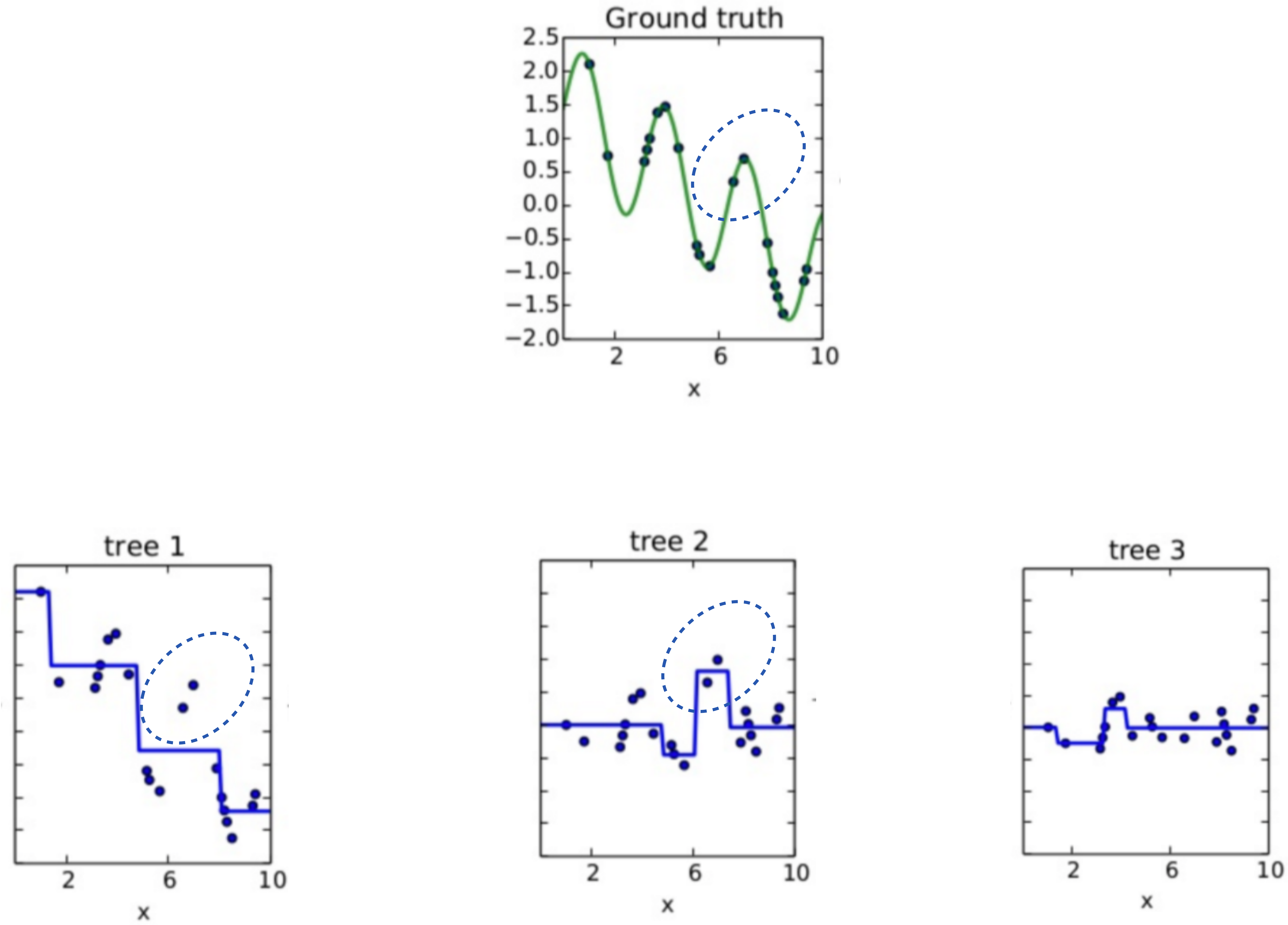
https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression



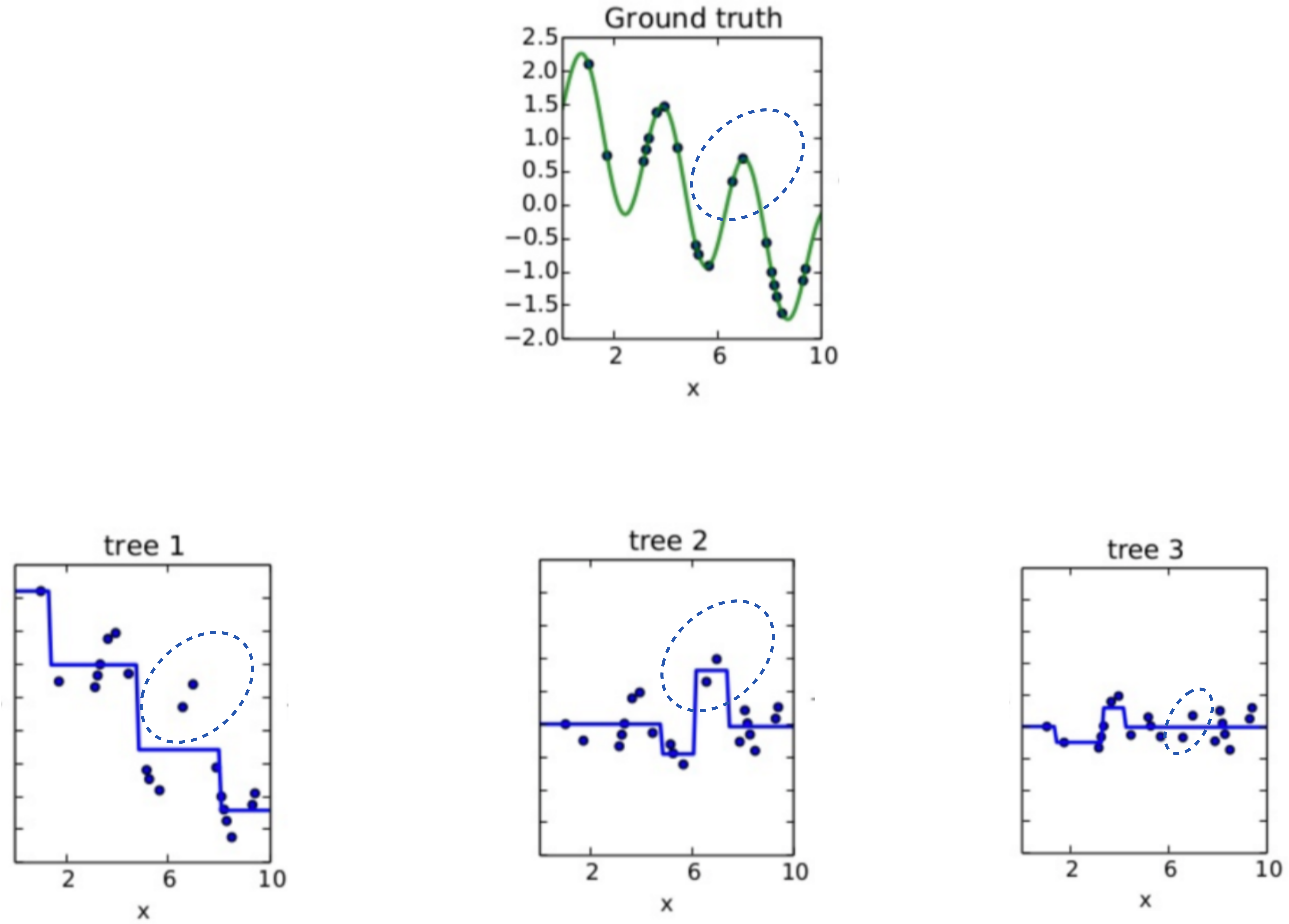
https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Regression



https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

Homework

Boosting in sklearn

Complete Exercise 11 in page 335 of textbook

Data set Caravan.csv is available in Blackboard

Boosting method in R is different than in python. Find out what are the corresponding parameters (number of trees, shrinkage value) in python sklearn library.

There are ada-boosting and gradient-boosting methods. You just need to choose one to perform the homework.