Boosting Algorithms

CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai

Anthony Goldbloom gives you the secret to winning Kaggle competitions

January 13, 2016 Andrew Fogg Big Data

Kaggle has become the premier Data Science competition where the best and the brightest turn out in droves – Kaggle has more than 400,000 users – to try and claim the glory. With so many Data Scientists vying to win each competition (around 100,000 entries/month), prospective entrants can use all the tips they can get.

And who better than Kaggle CEO and Founder, Anthony Goldbloom, to dish out that advice? We caught up with him at Extract SF 2015 in October to pick his brain about how best to approach a Kaggle competition.

Anthony Goldbloom gives you the secret to winning Kaggle competitions

As long as Kaggle has been around, Anthony says, it has almost always been ensembles of decision trees that have won competitions.

It used to be random forest that was the big winner, but over the last six months a new algorithm called XGboost has cropped up, and it's winning practically every competition in the structured data category.

What is Boosting ?

How does it work ?

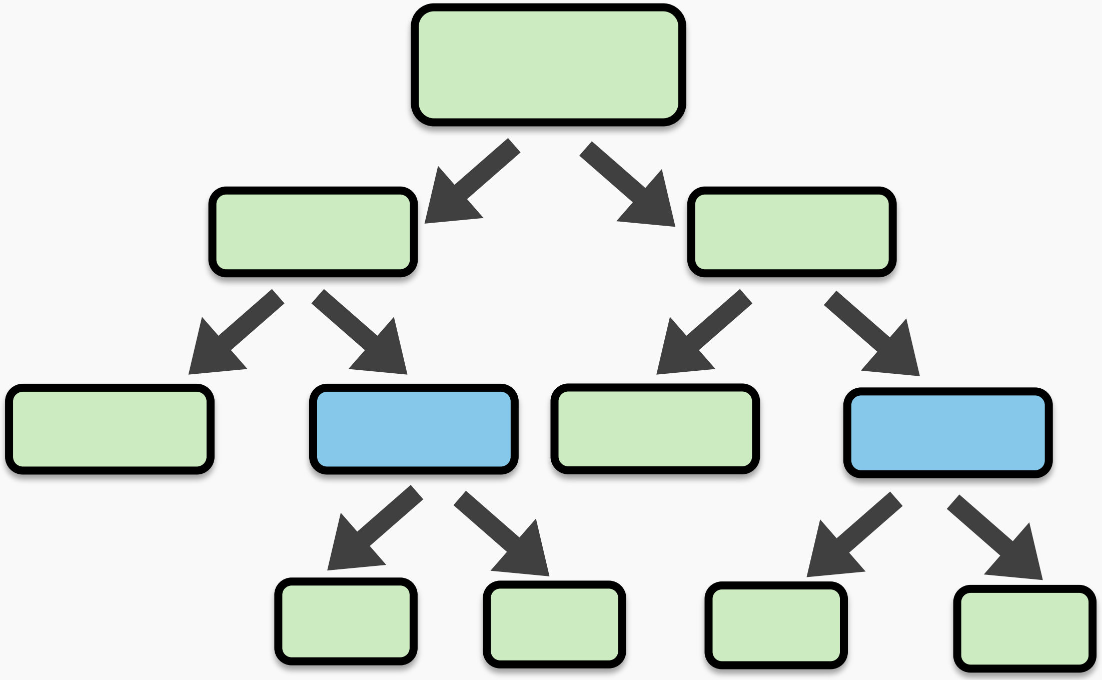
Why is it so good ?

What is Boosting ?

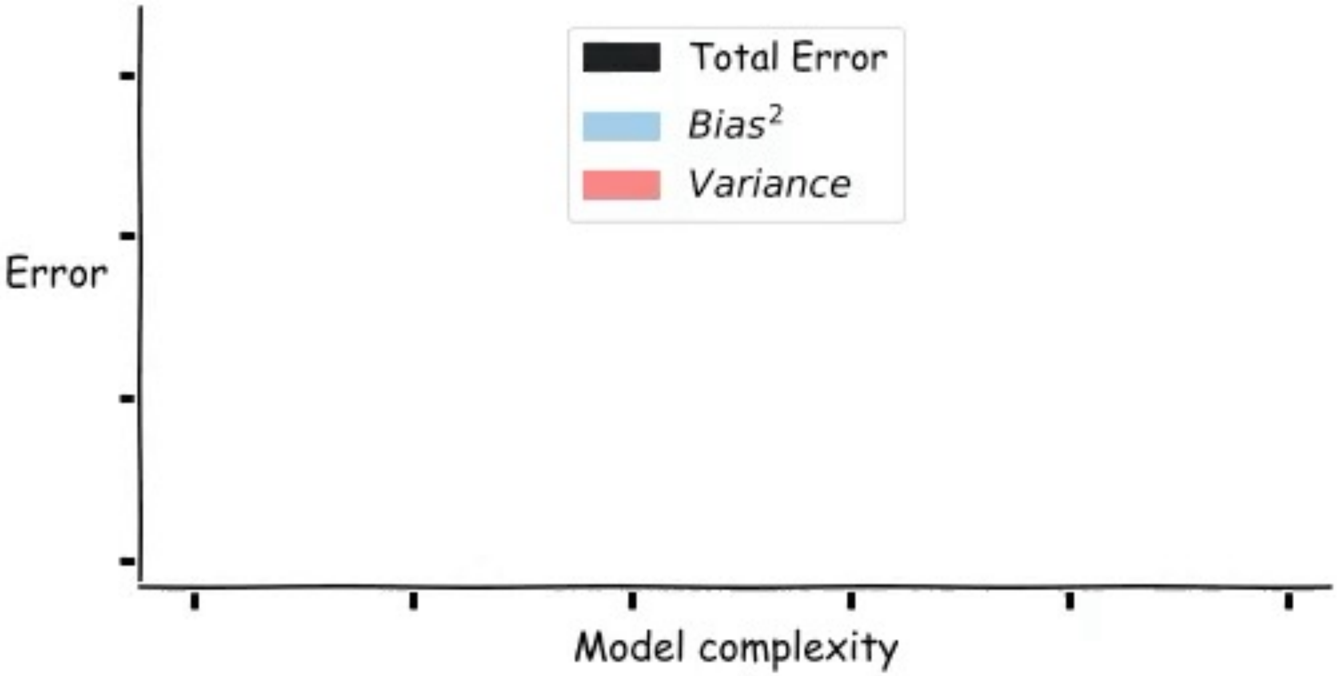
RECAP: Decision Trees

DECSION TREE ISSUES?

- Shallow trees:
  - Shallow trees (with very few leaves) suffer from high bias and do not train well.
- Deep Trees:
  - Deep trees (with large number of nodes and leaves) have low bias, but suffer from high variance leading to very low generalization error.



RECAP: Decision Trees



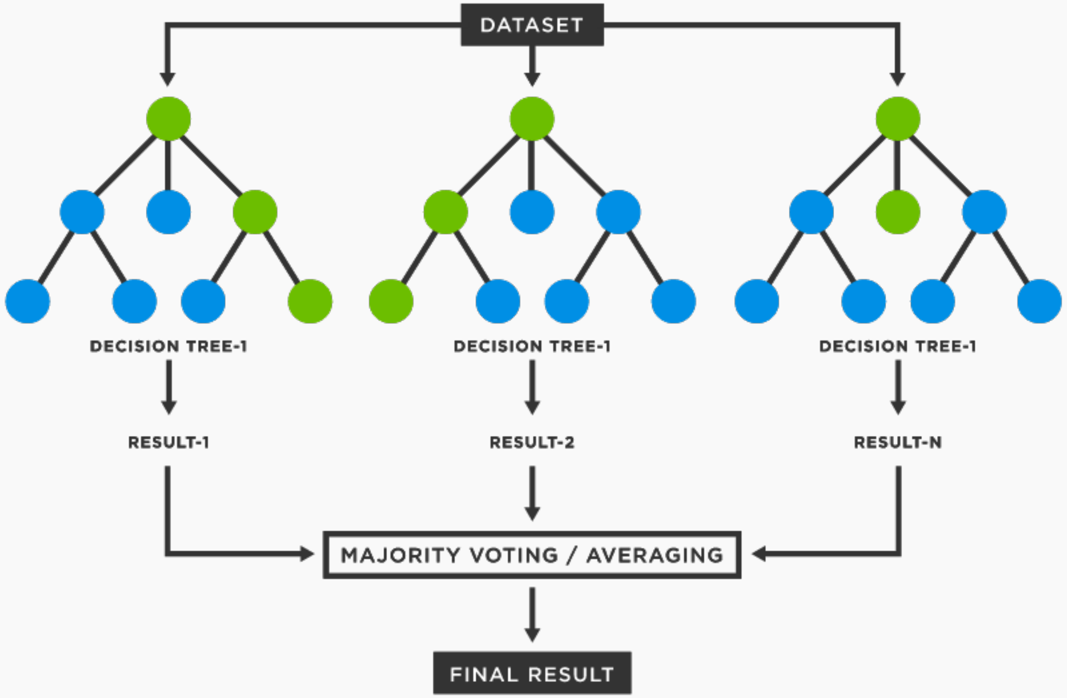
Decision Tree growth

Bias-Variance Tradeoff

Random Forest – The only solution ?

RF ISSUES?

- Variance:
  - Although variance reduction is better than bagging, the generalization error is still high
- Speed:
  - Large number of trees can make the algorithm very slow and ineffective for real-time predictions



Motivation for Boosting

Question: Could we address the shortcomings of single decision trees models in some other way?

For example, rather than performing variance reduction on complex trees, can we decrease the bias of simple trees - make them more expressive?

Can we learn from our mistakes?

A solution to this problem, making an expressive model from simple trees, is another class of ensemble methods called ***boosting***.

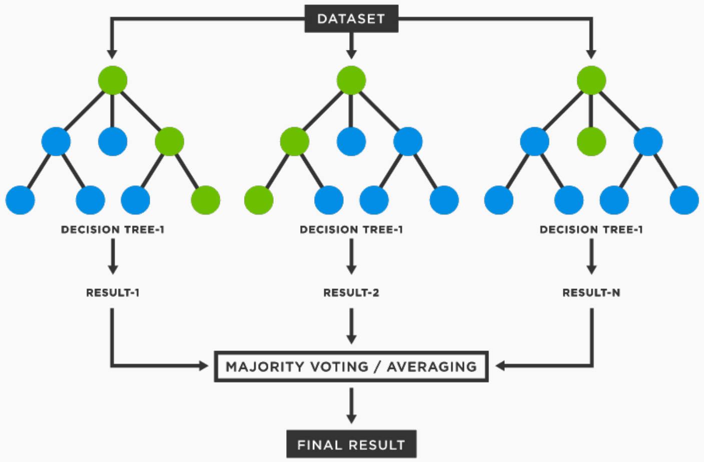
Random Forest – The only solution ?

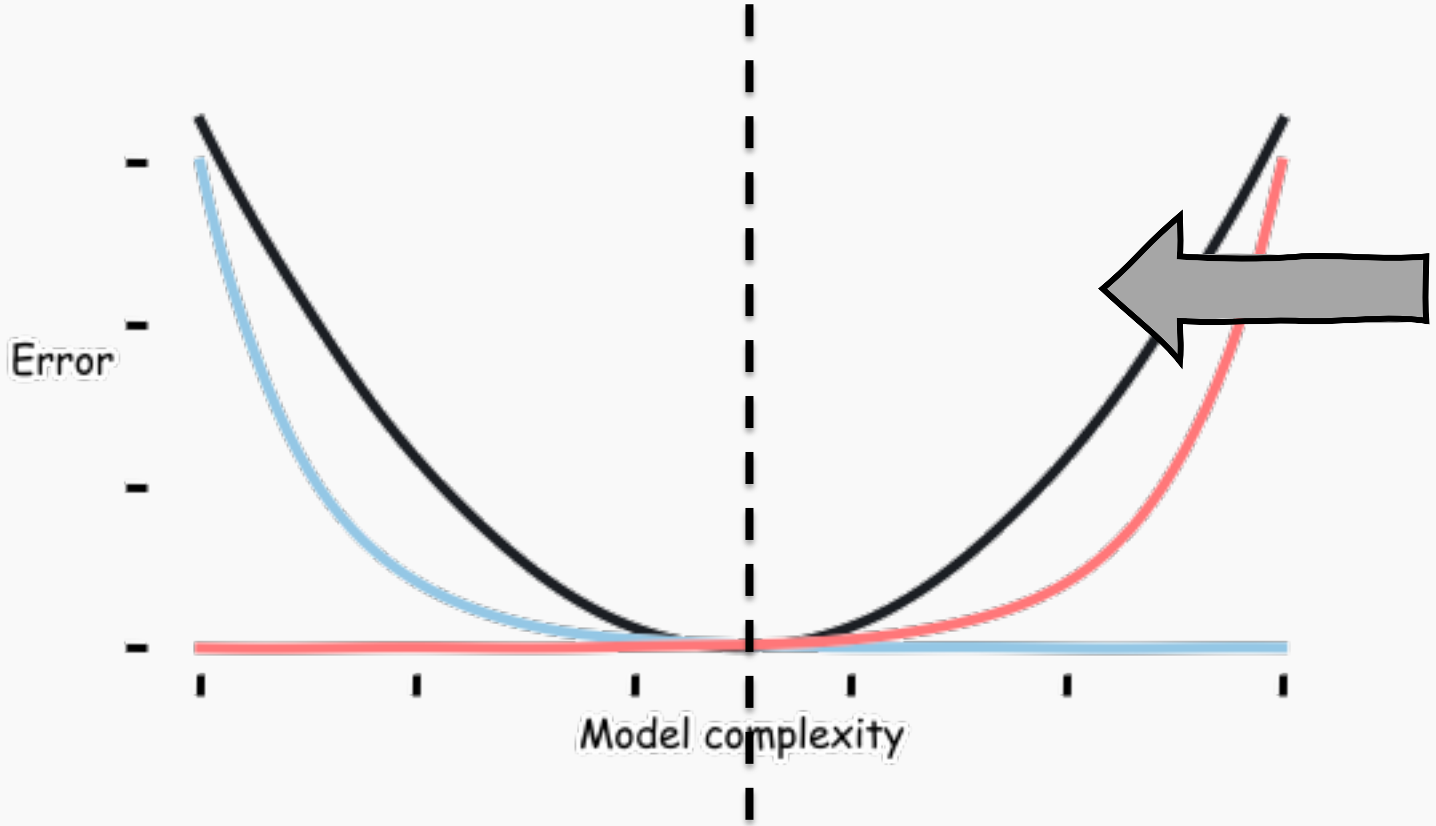
OPTION #1

Reduce variance

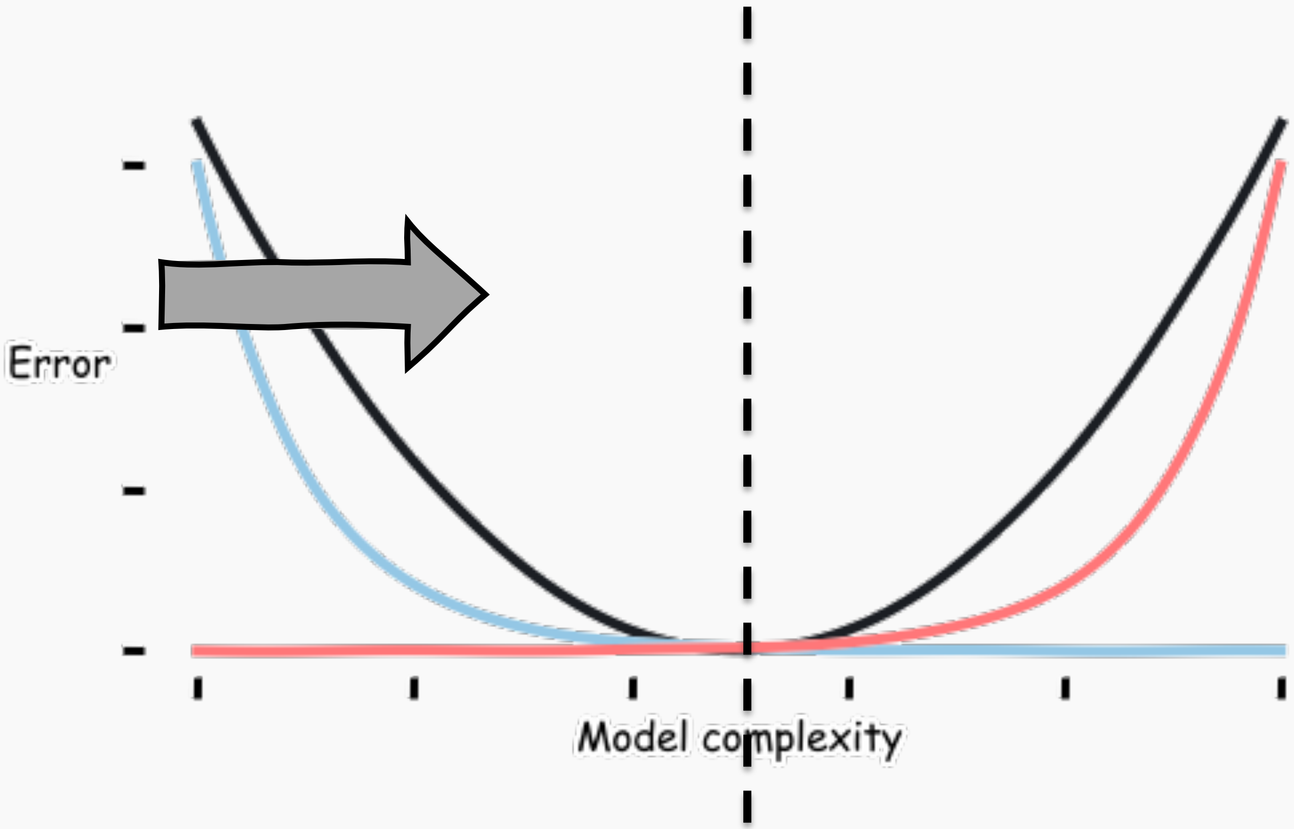
$$d_{trees} \rightarrow \infty$$

$$var \rightarrow 0$$





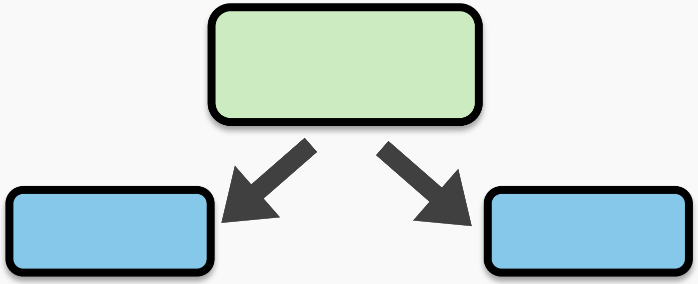
Random Forest – The only solution ?



OPTION #2

Reduce bias

$$d_{trees} \to \infty$$
$$bias \to 0$$



Boosting

NEW IDEA 💡

- **Boosting** methods are general algorithms which combine several "**weak learners**" to produce a strong rule.

- The first implementation of Boosting was '**Adaboost**' invented by Robert Schapire and Yoav Freund in 1996.

- Boosting algorithms are fast, easy to compute and very accurate and are the de-facto optimization tree algorithms.

Rob Schapire & Yoav Freund

HOW DOES IT WORK ?

(FUN EDITION)

CS 109A FINALS

CS 109A FINALS

TOPIC: BOOSTING

DATE: DEC-14-2021

Q1:

Q2:

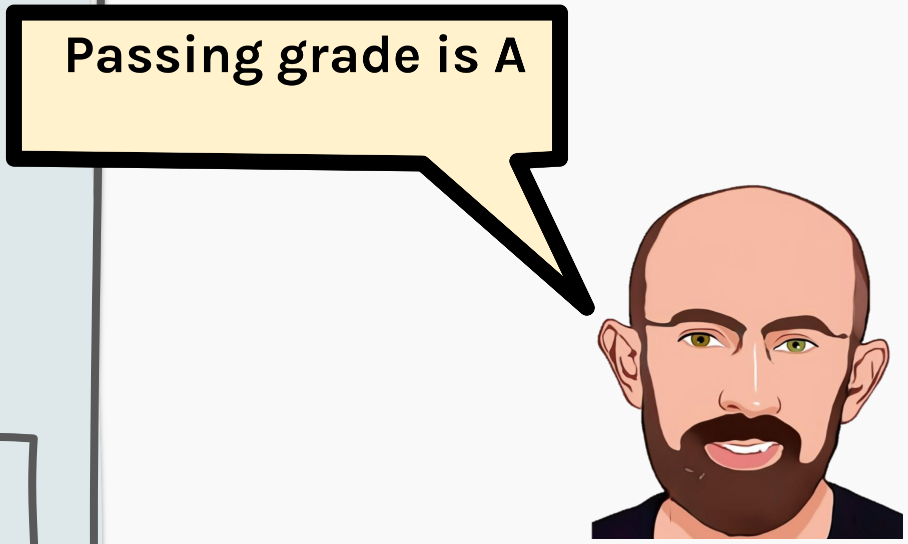
Q3:

...

Q10:

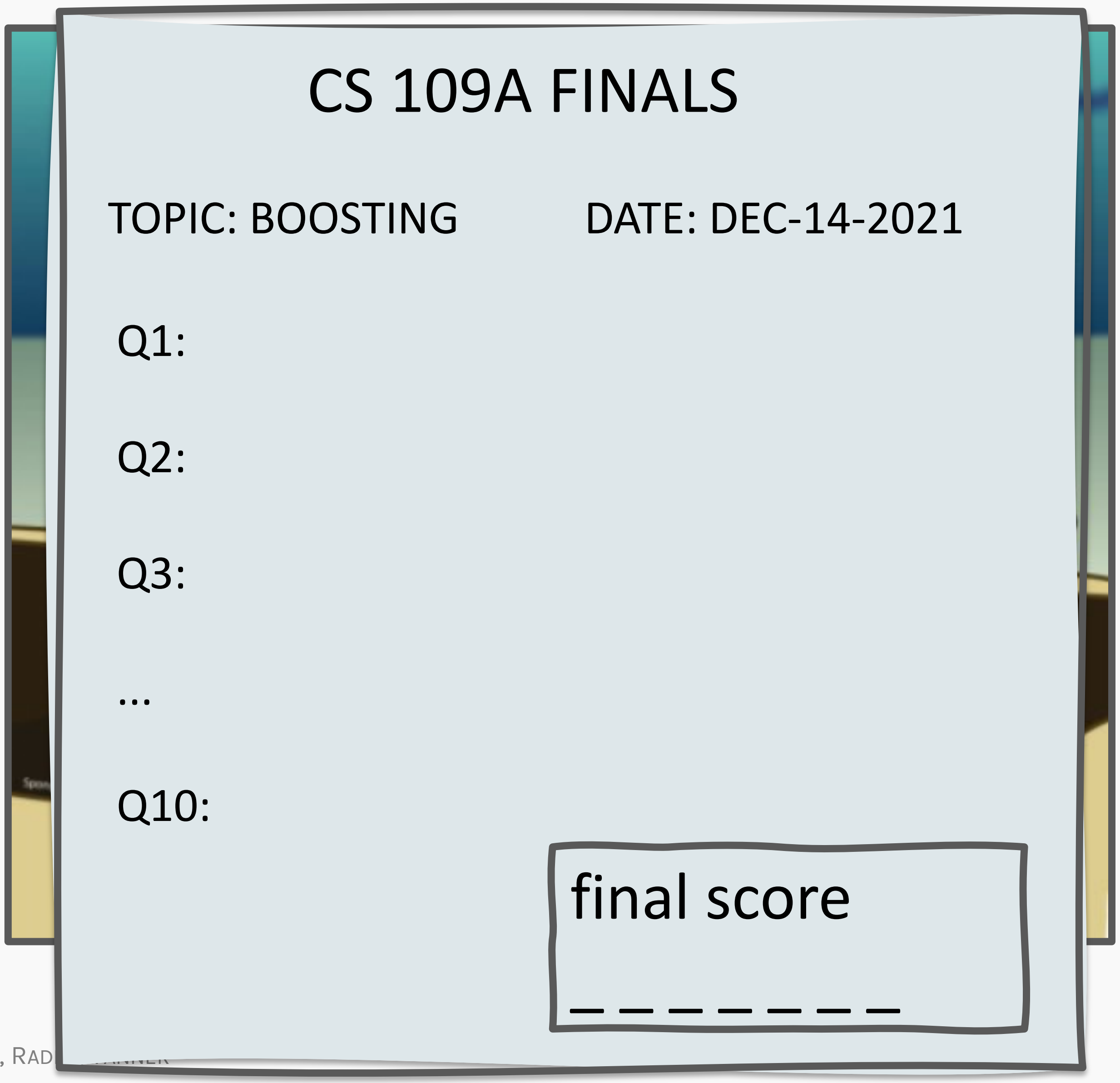
final score

_ _ _ _ _ _ _



OPTION #1

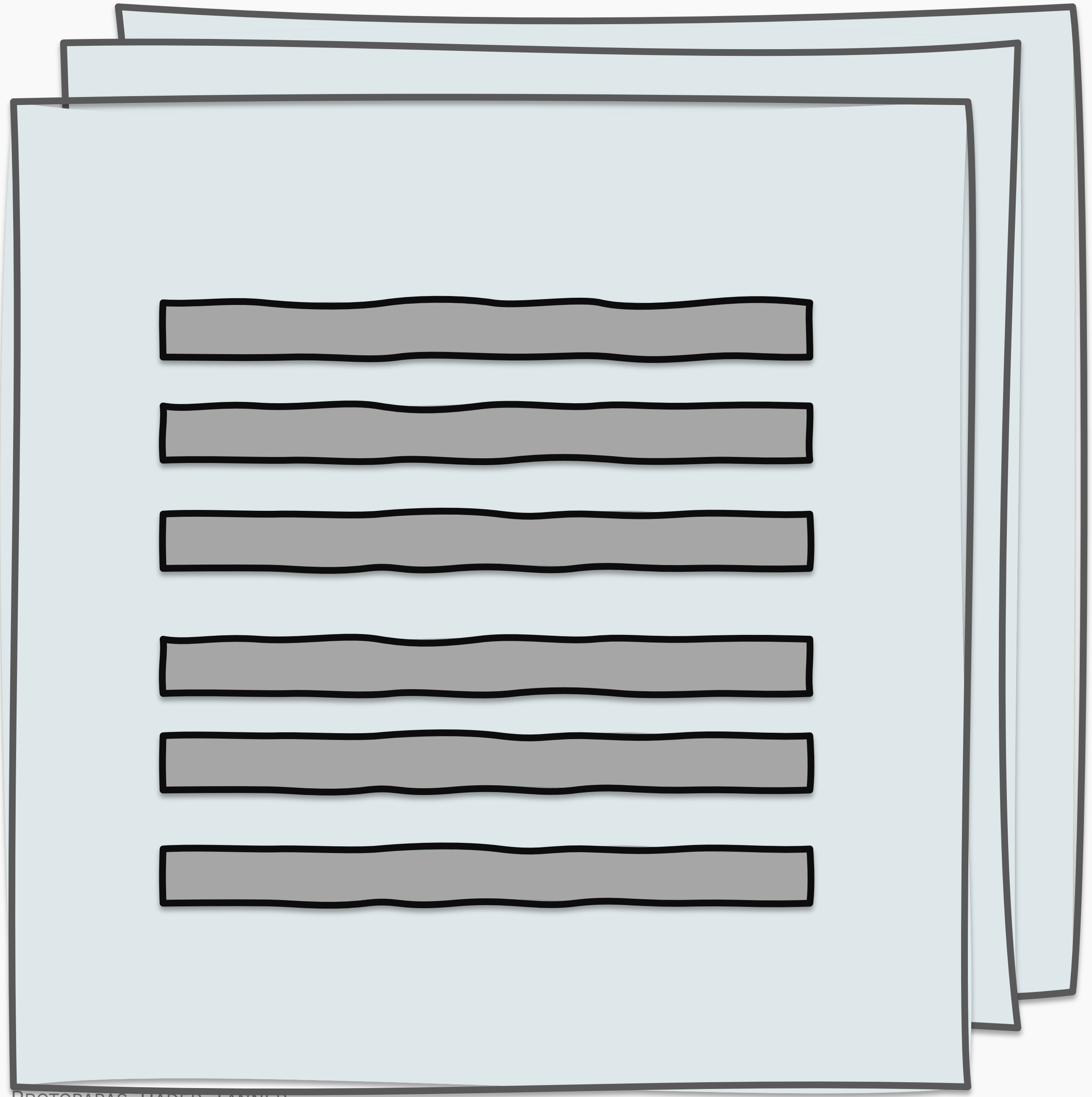
1. Steal the time-stone from Dr. Strange. (COO and faculty @ HFP Consulting)
2. Go back to 1996 and meet Rob Schapire and Yoav Freund.
3. Follow their work for at least a decade to understand everything about boosting.
4. Return to the present and nail the test.
5. Repeat for another test



OPTION #2

STEP #1:

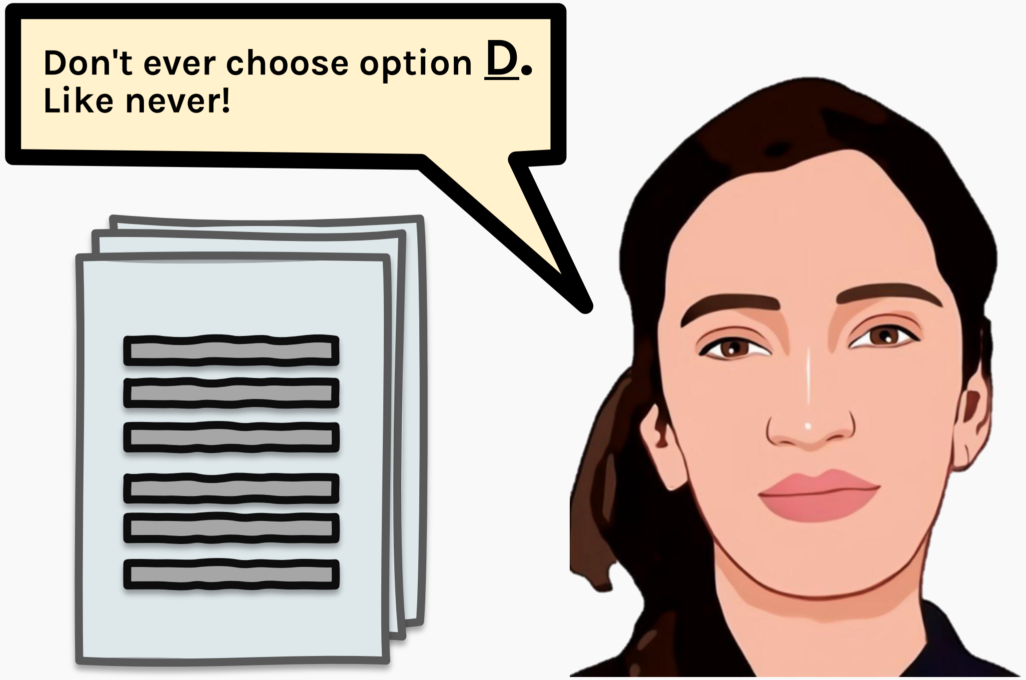
Go to the library and get previous year question papers.



OPTION #2

STEP #2:

Find a helpful student and ask her to give you a "rule of thumb" to get at least some answers right.

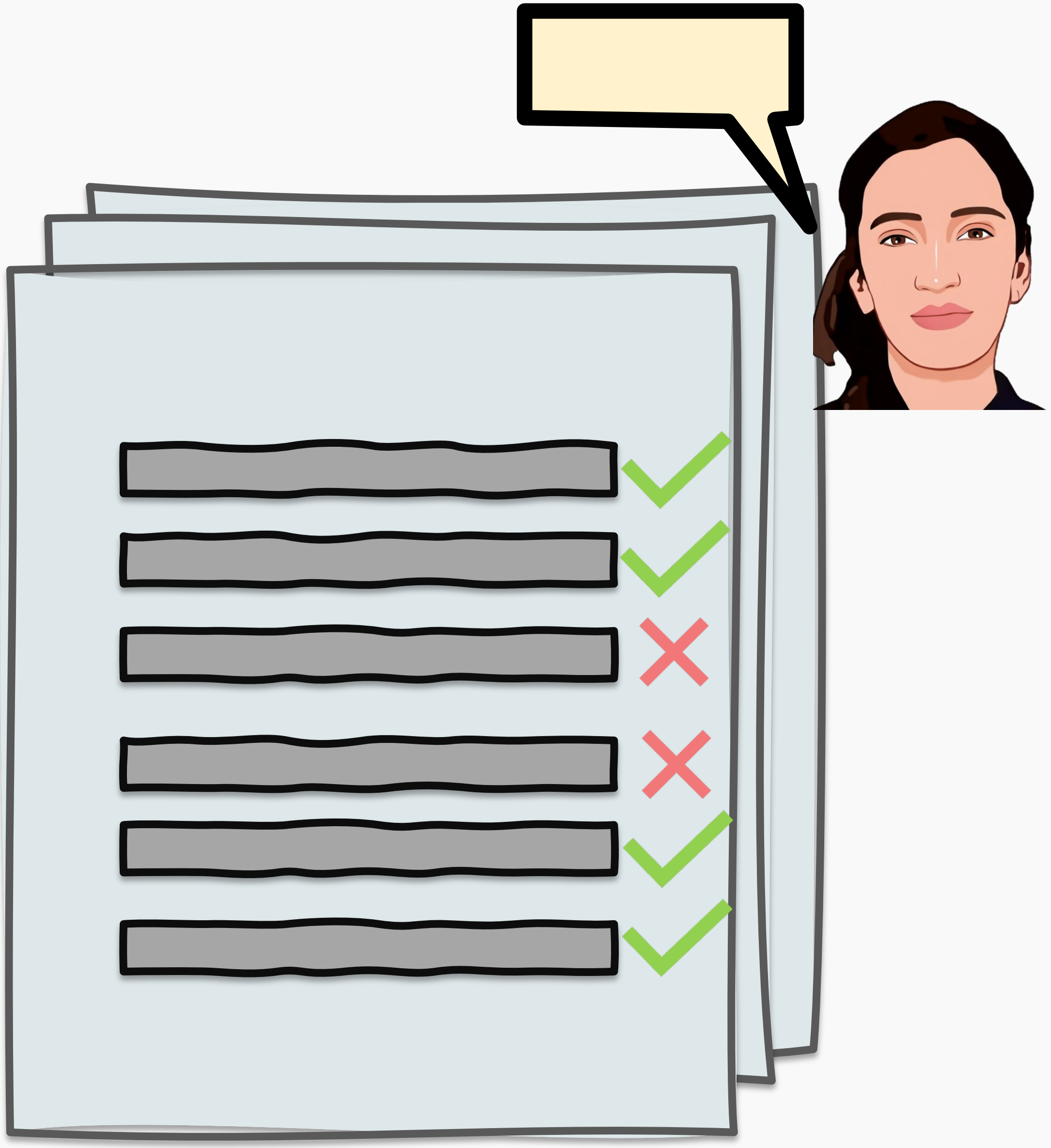


OPTION #2

DOES THE "RULE" WORK ?

Test out the rule.

It worked 60% of the time. Not bad!!

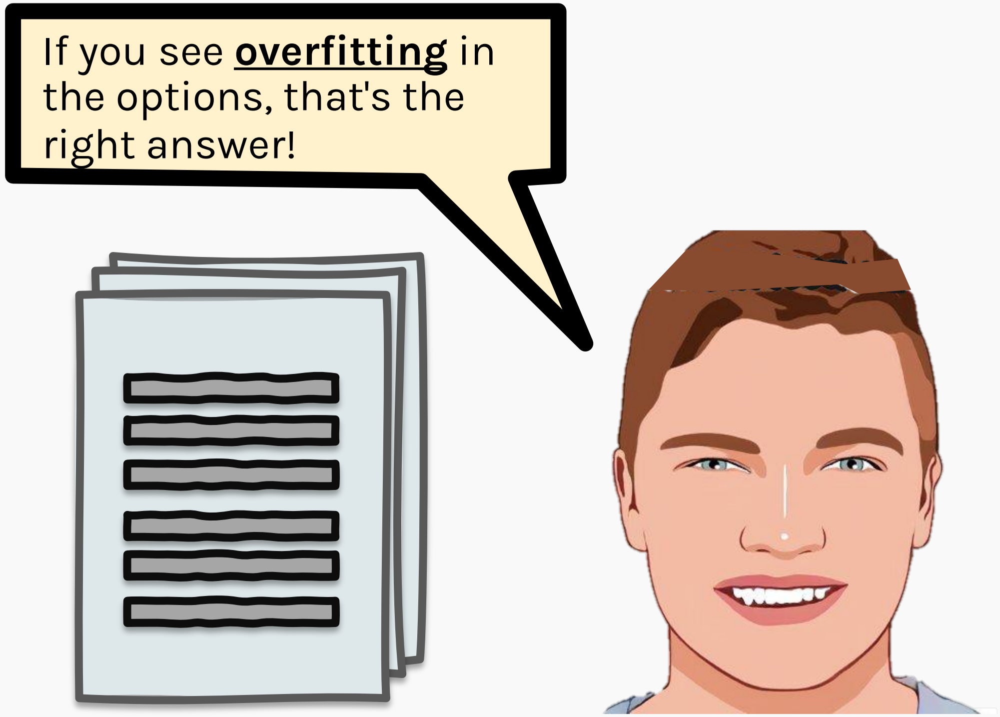


OPTION #2

STEP #3:

Find a TA and ask him to also give you a "rule of thumb" to get at least some answers right.

make sure to focus on the ones you got wrong before.

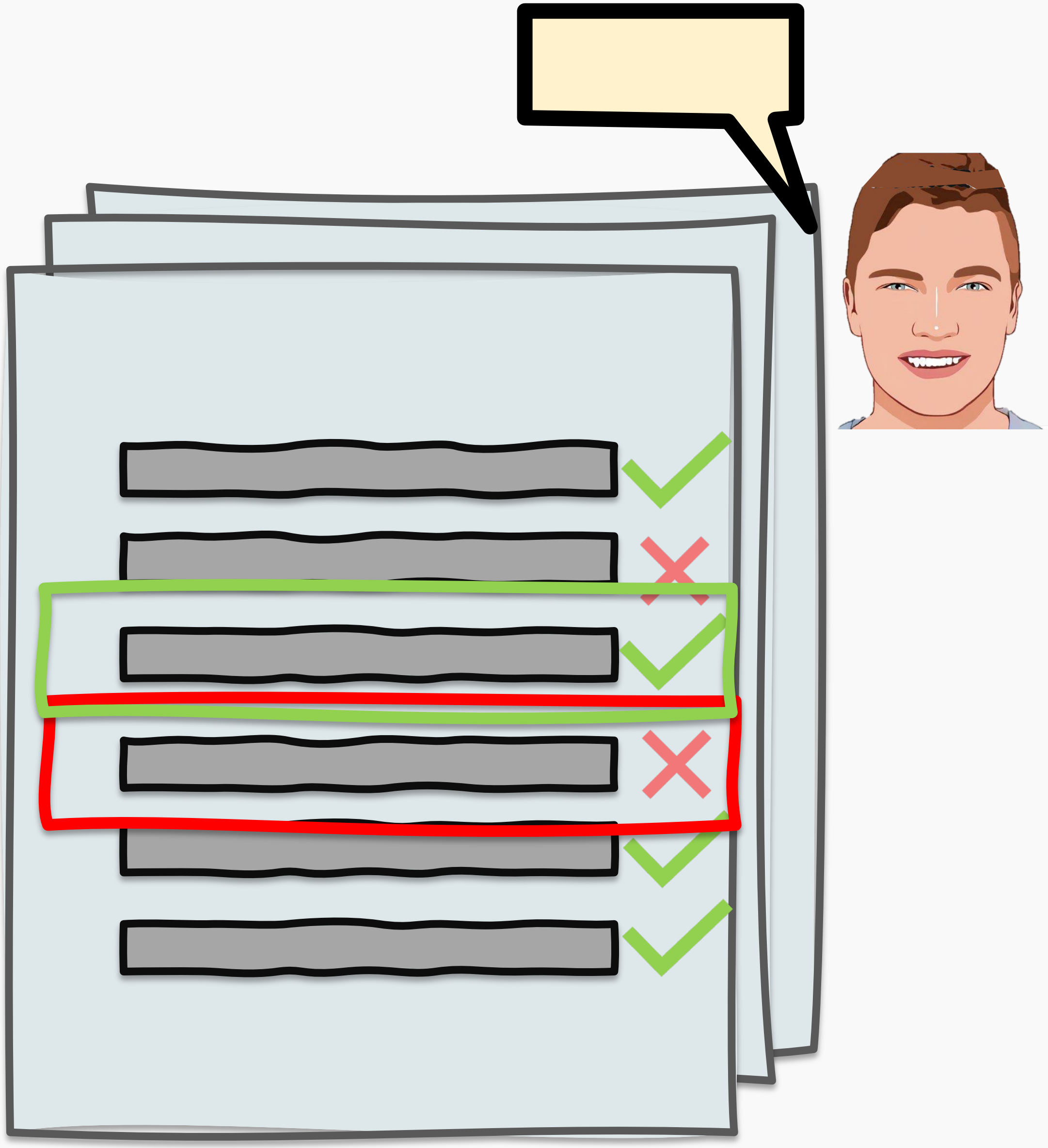


OPTION #2

DOES THE "RULE" WORK ?

Test out the new rule.

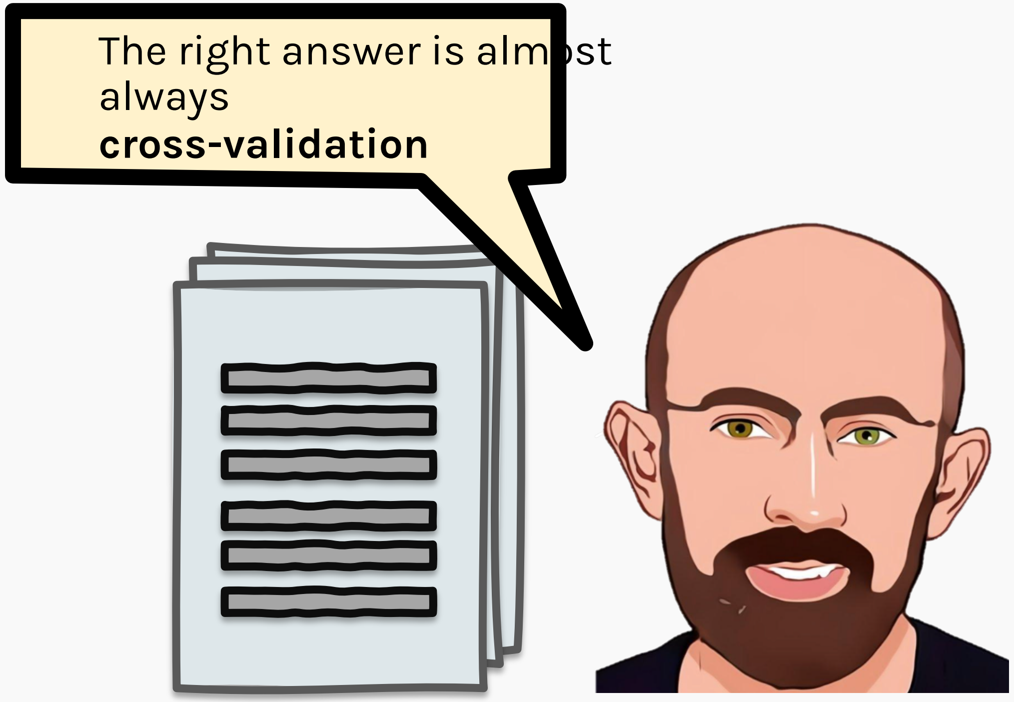
It works well on difficult problems! But a few problems persist.



OPTION #2

STEP #4:

Call your favorite professor and focus on the ones you got wrong before!

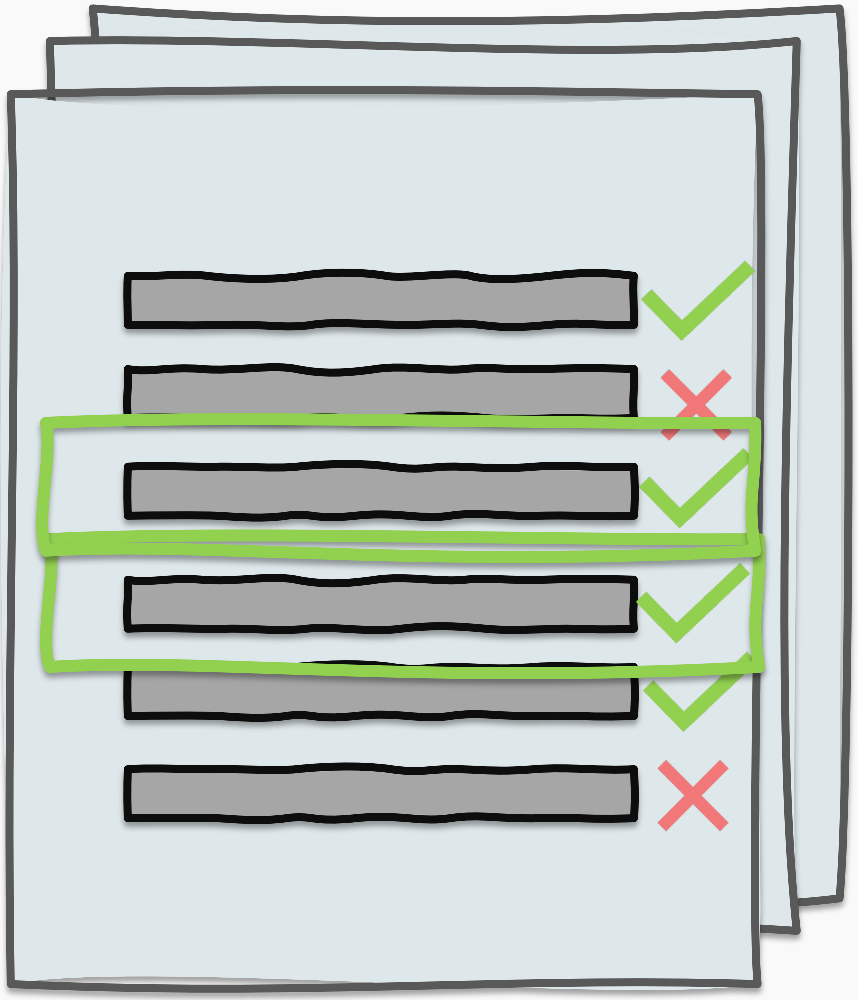


OPTION #2

DOES THE "RULE" WORK ?

Test out the new rule.

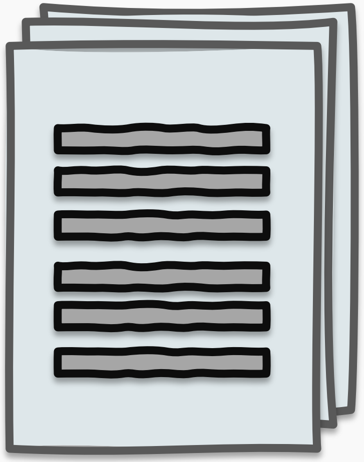
The new rule works well on the difficult problems!

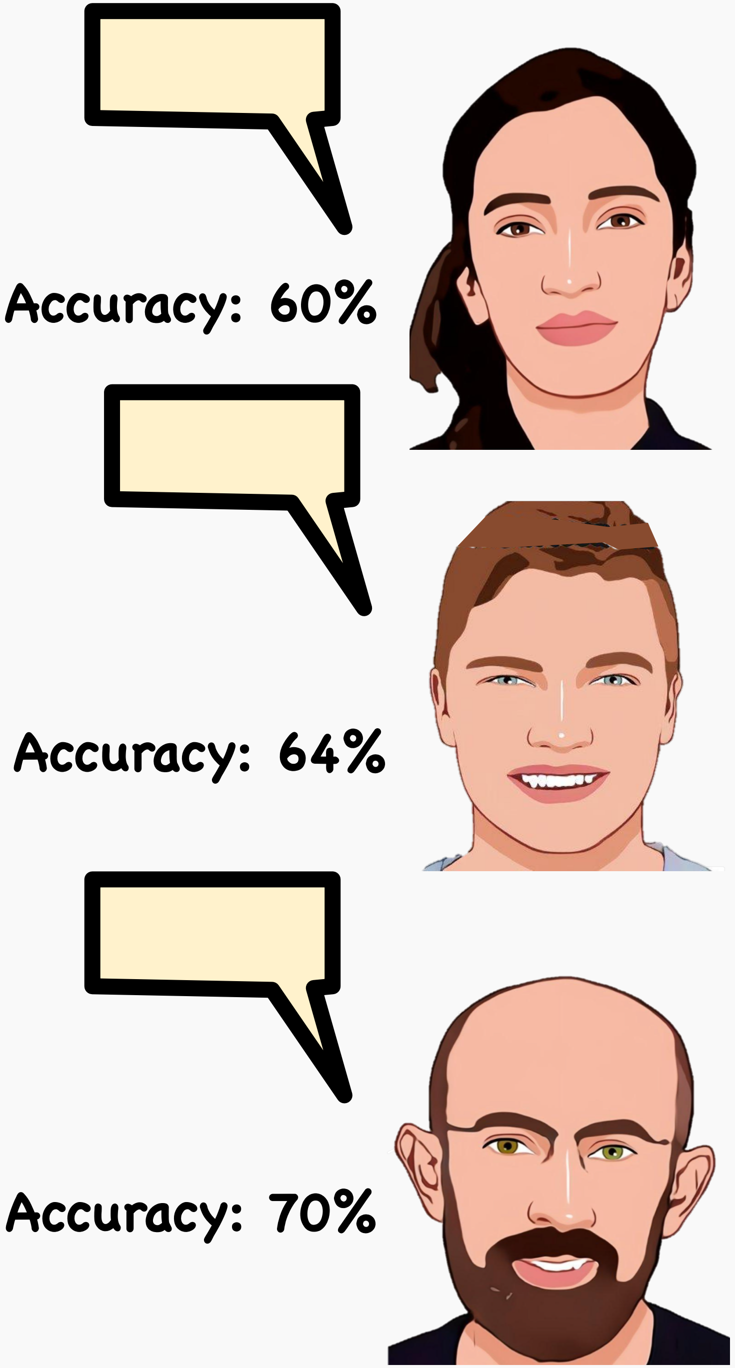


OPTION #2

STEP #5:

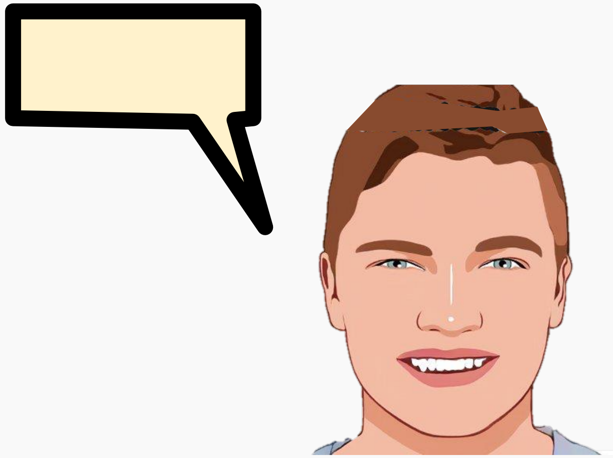
Combine the rules, but pay more attention to the ones that were more often right





OPTION #2

Accuracy: 60% **Accuracy: 64%** **Accuracy: 70%**



$$Strategy = \alpha * Rule_1 + \beta * Rule_2 + \gamma * Rule_3$$

OPTION #2

FINAL STEP:

Take the test with these approximate rules, weighted by how well each rule performed.

A+

HOW DOES IT WORK ?

(MATH EDITION)

Gradient Boosting

The key intuition behind boosting is that one can take an ensemble of simple models $\{T_h\}_{h \in H}$ and additively combine them into a single, more complex model.

Each model T_h might be a poor fit for the data, but a linear combination of the ensemble

$$T = \sum_h \lambda_h T_H$$

can be expressive/flexible.

Question: But which models should we include in our ensemble? What should the coefficients or weights in the linear combination be?

Gradient Boosting: the algorithm

Gradient boosting is a method for iteratively building a complex regression model T by adding simple models. Each new simple model added to the ensemble compensates for the weaknesses of the current ensemble.

1. Fit a simple model $T^{(0)}$ on the training data

$$\{(x_1, y_1), \dots, (x_N, y_N)\}$$

Set $T \leftarrow T^{(0)}$. Compute the residuals $\{r_1, \dots, r_N\}$ for T.

2. Fit a simple model, $T^{(1)}$, to the current ***residuals***, i.e. train using

$$\{(x_1, r_1), \dots, (x_N, r_N)\}$$

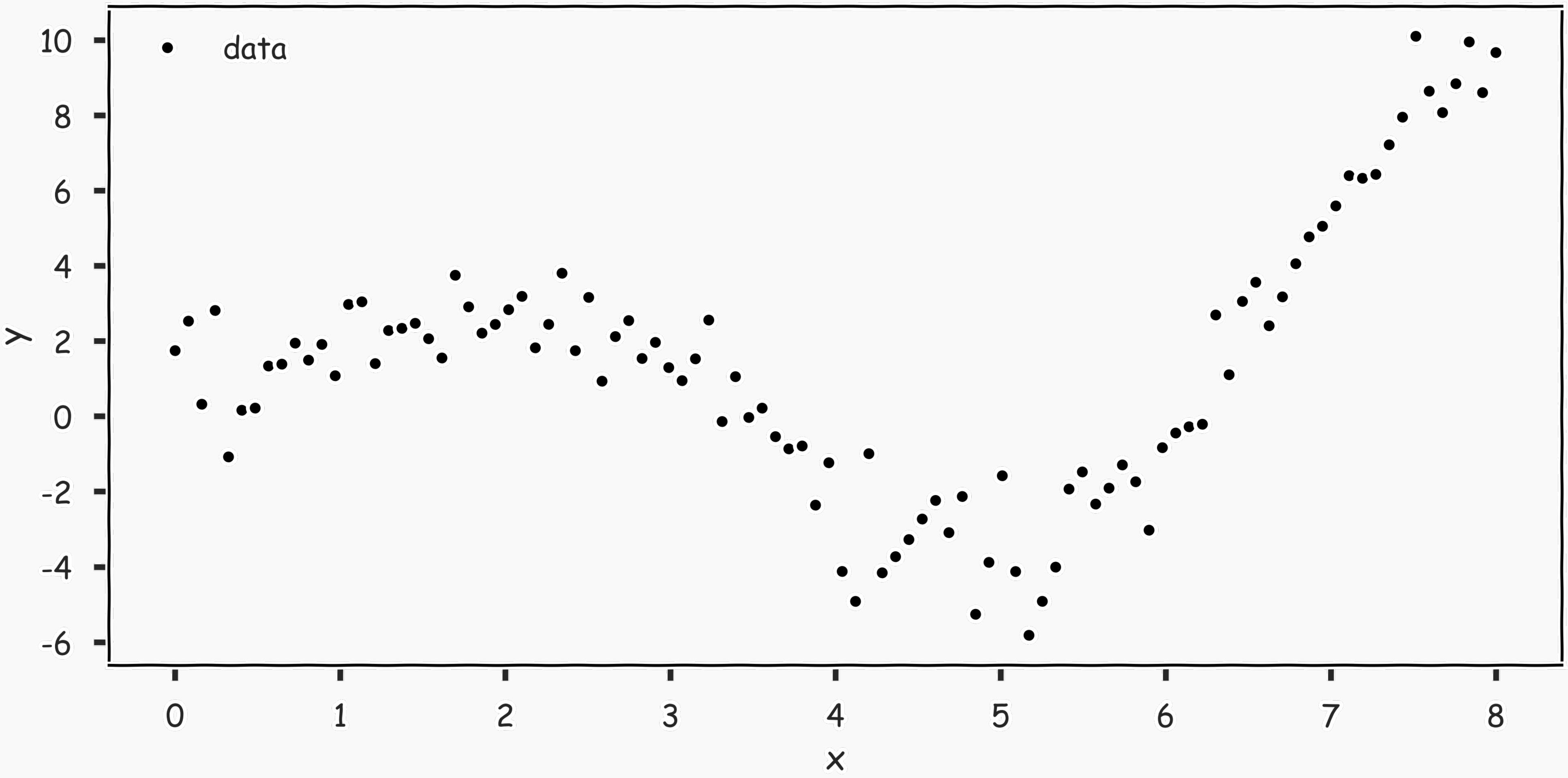
3. Set $T \leftarrow T + \lambda T^{(1)}$

4. Compute residuals, set $r_n \leftarrow r_n - \lambda T^i(x_n),\ n = 1, \dots, N$

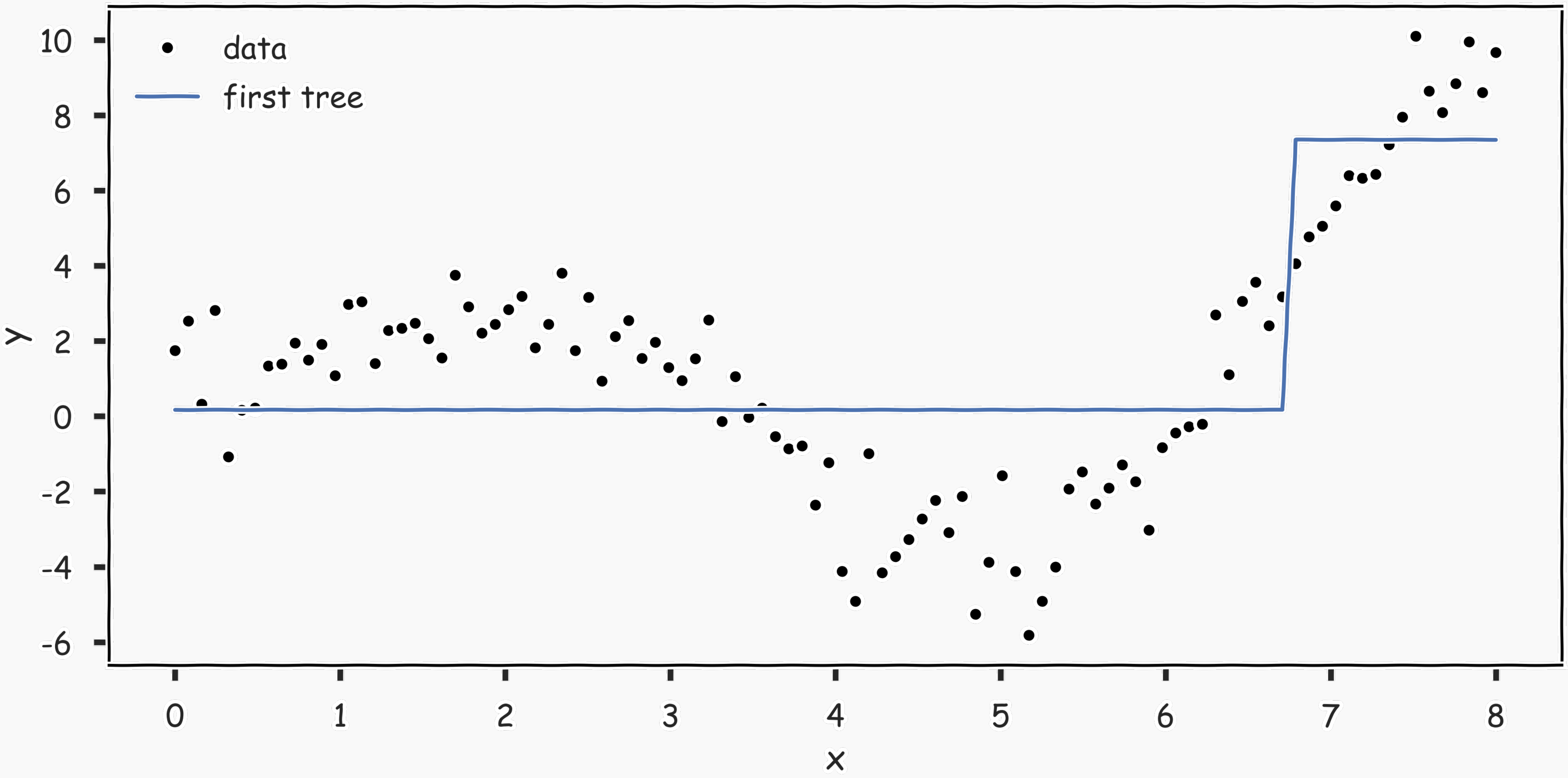
5. Repeat steps 2-4 until ***stopping*** condition met.

where λ is a constant called the ***learning rate***.

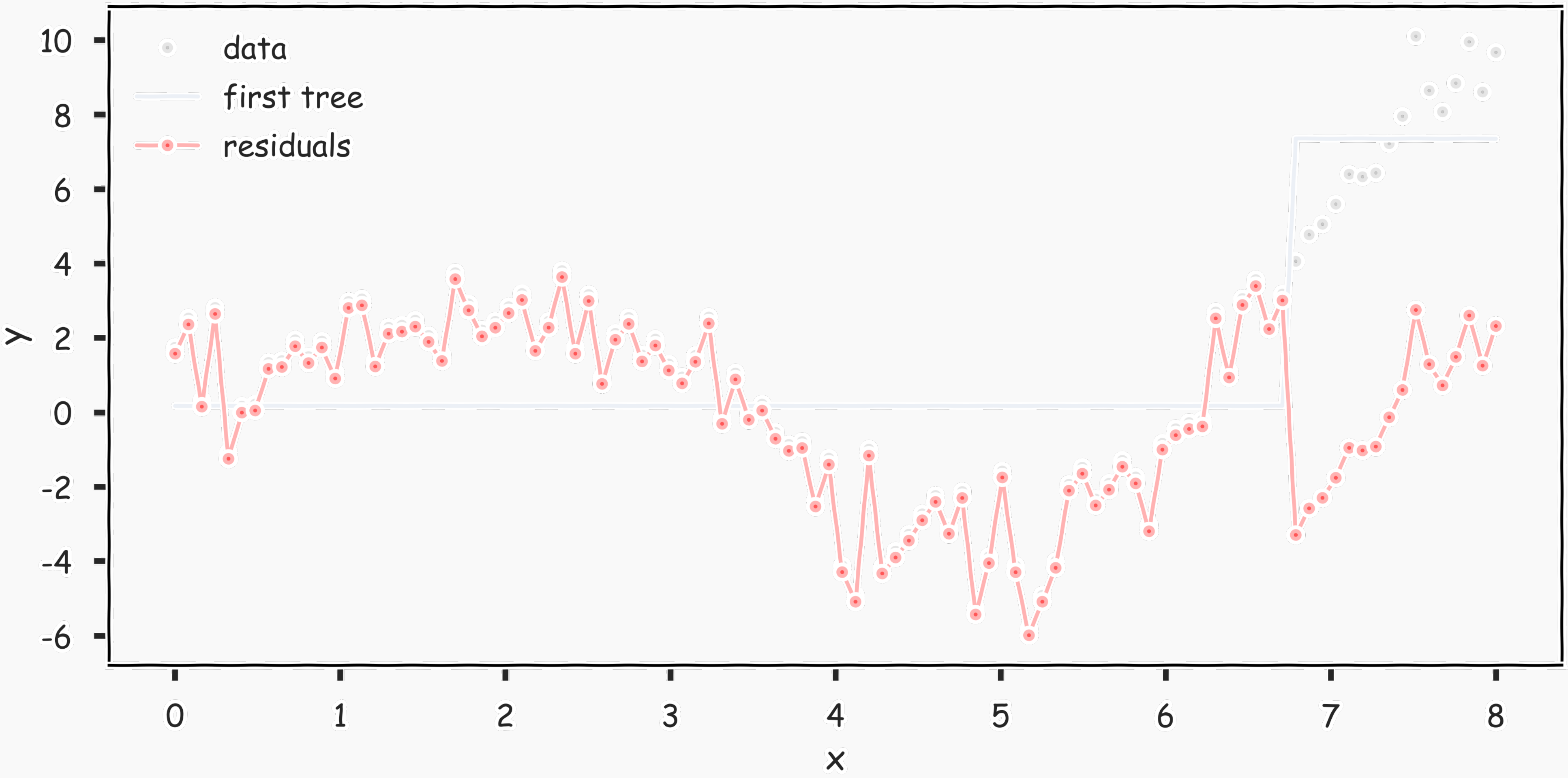
Gradient Boosting: illustration



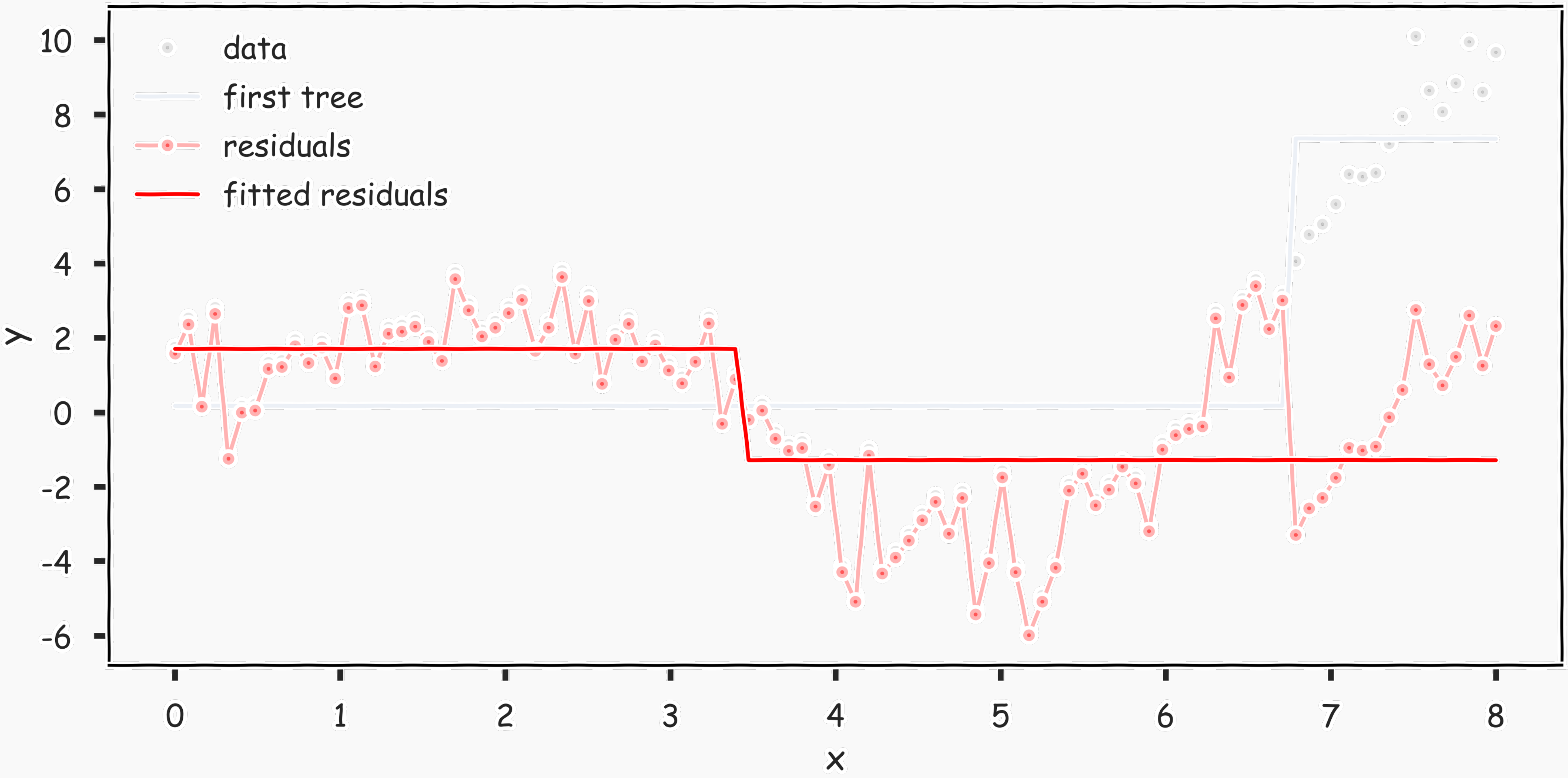
Gradient Boosting: illustration



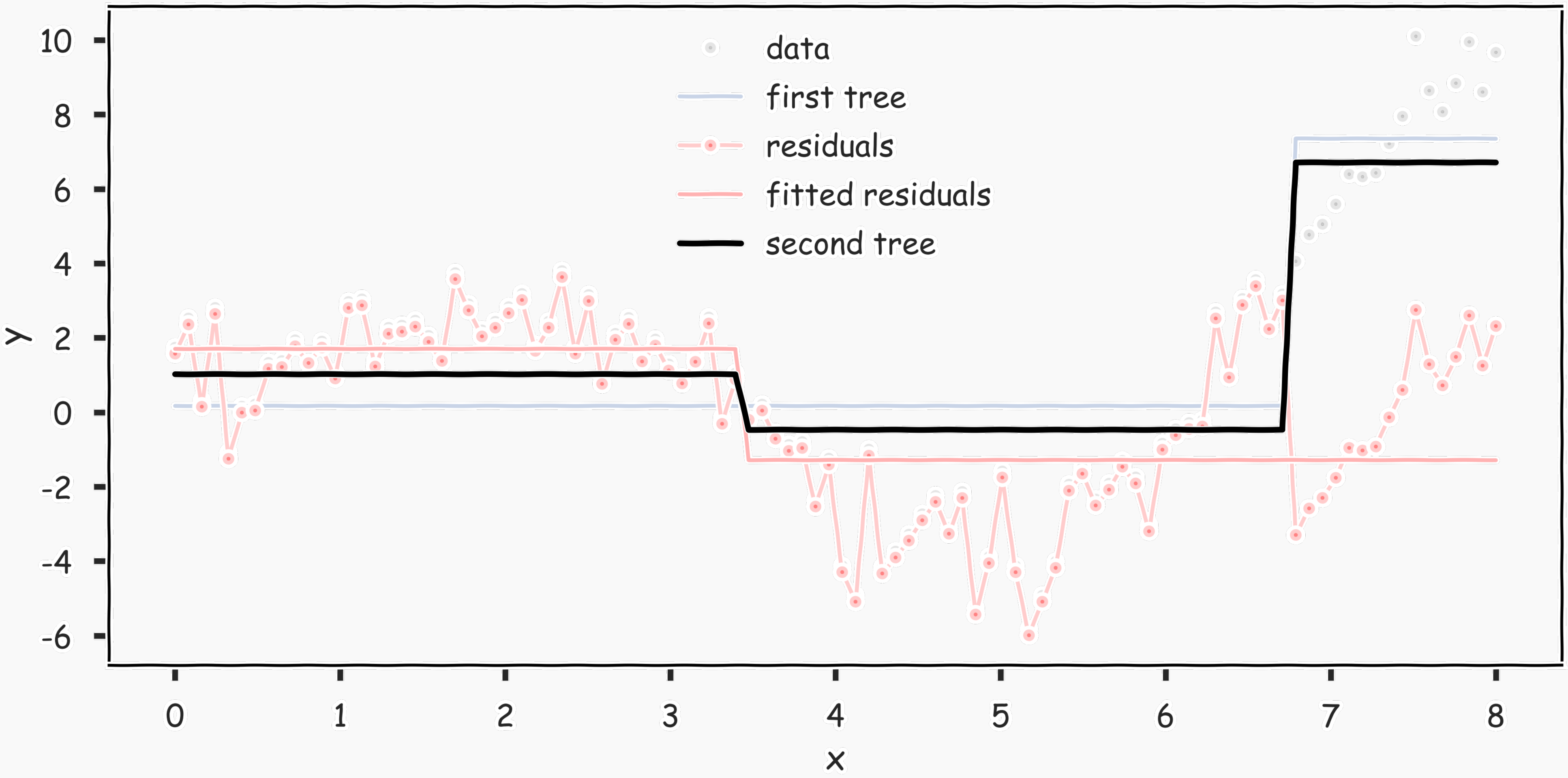
Gradient Boosting: illustration



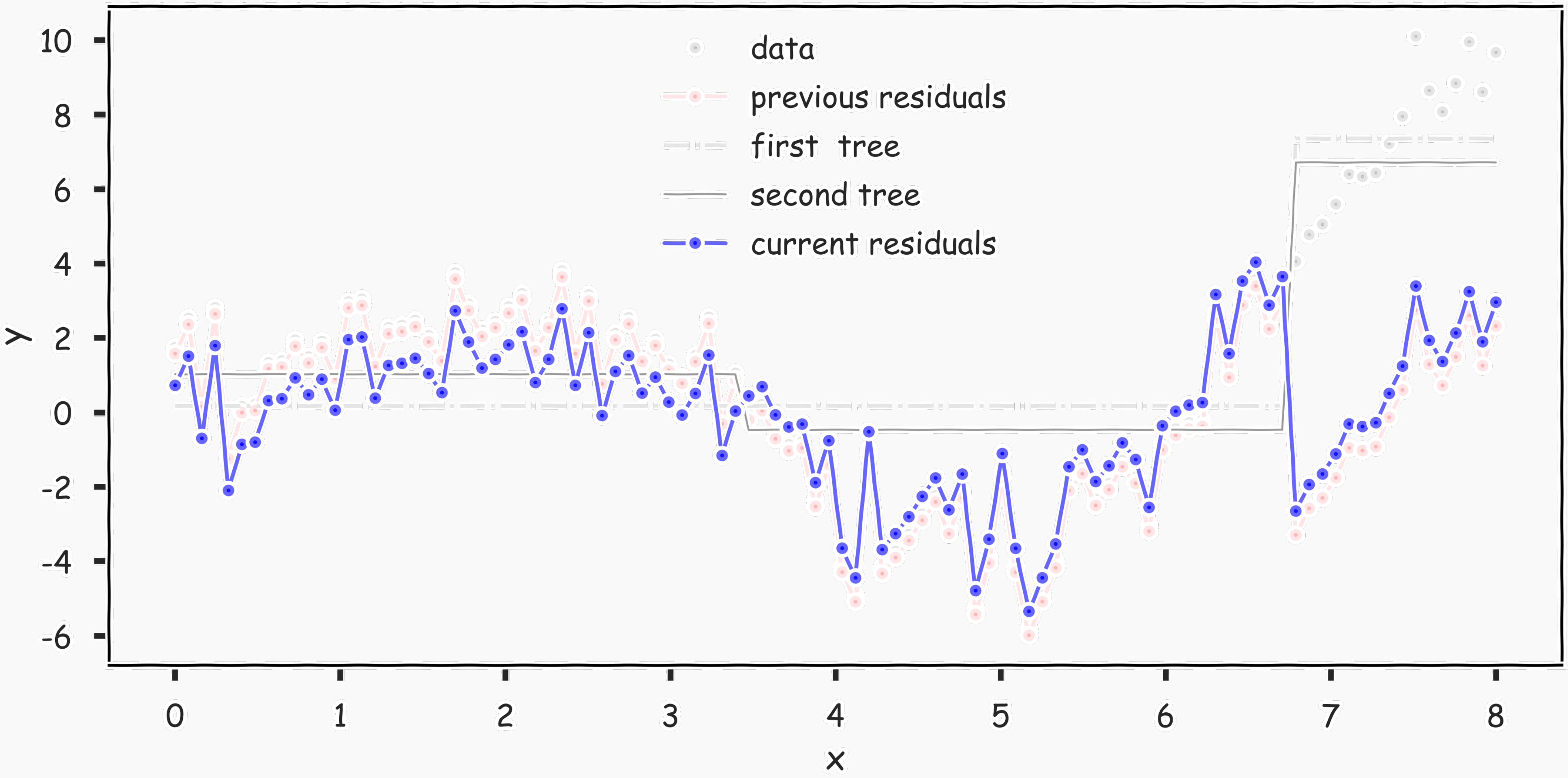
Gradient Boosting: illustration



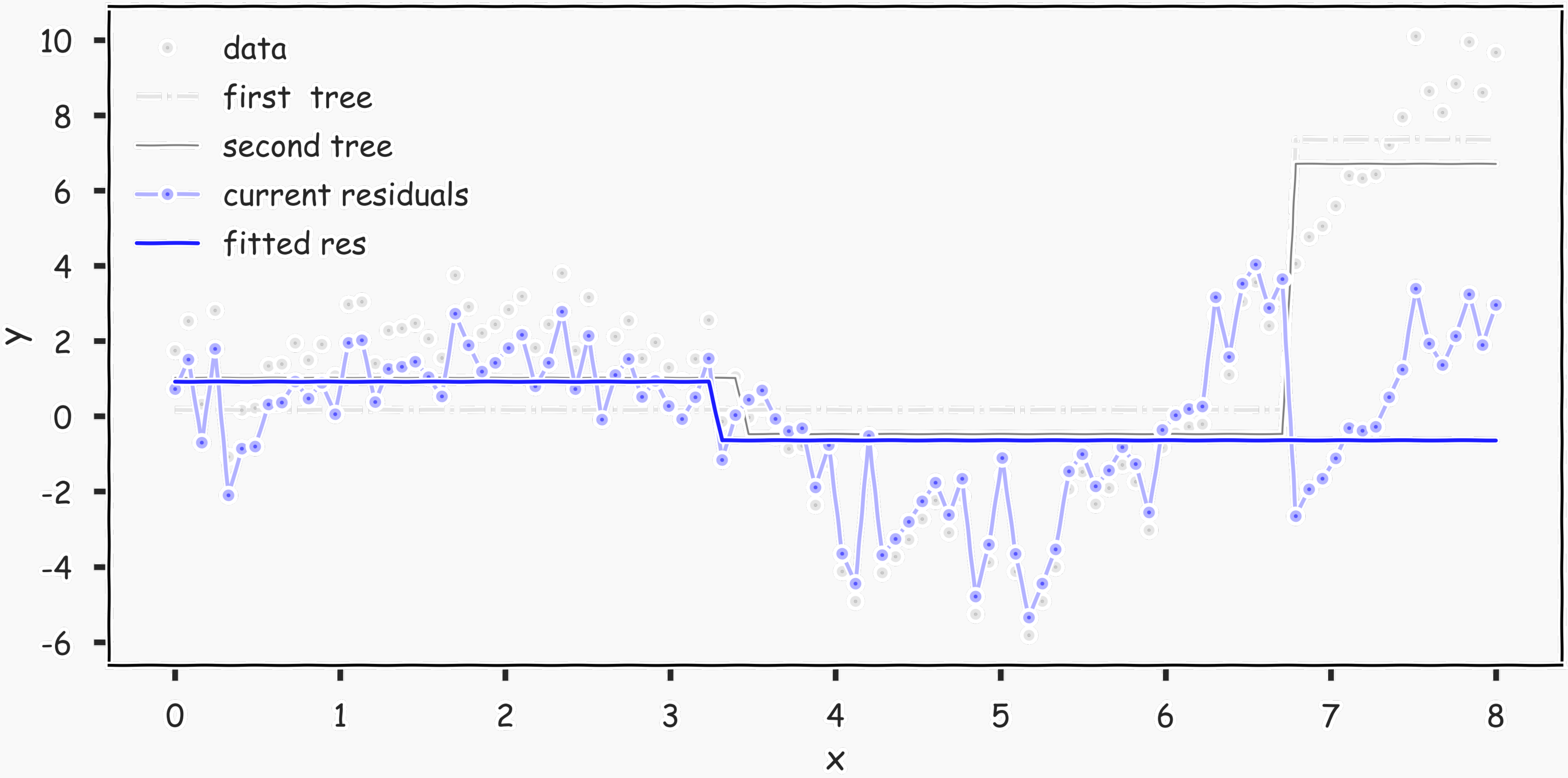
Gradient Boosting: illustration



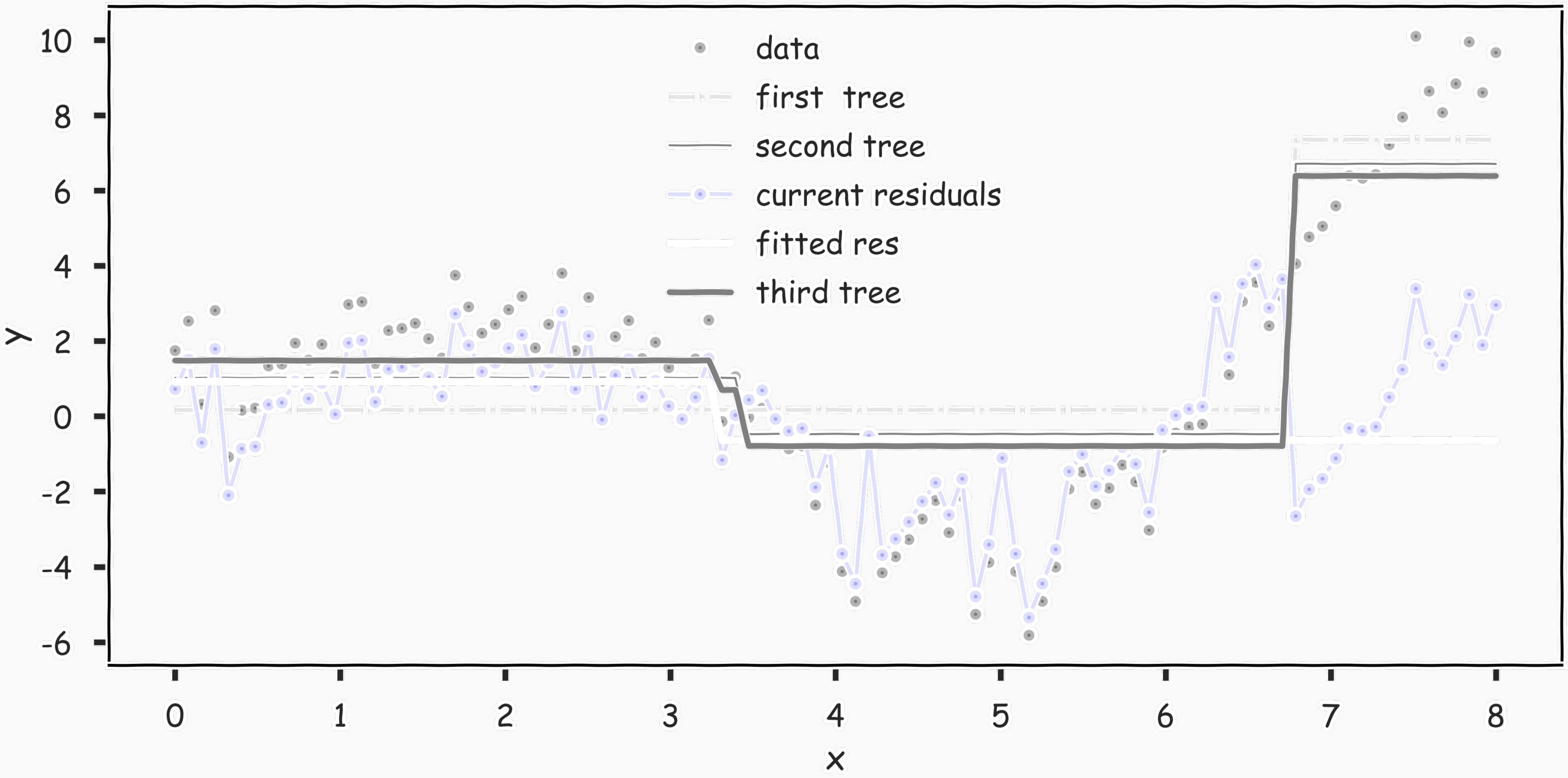
Gradient Boosting: illustration

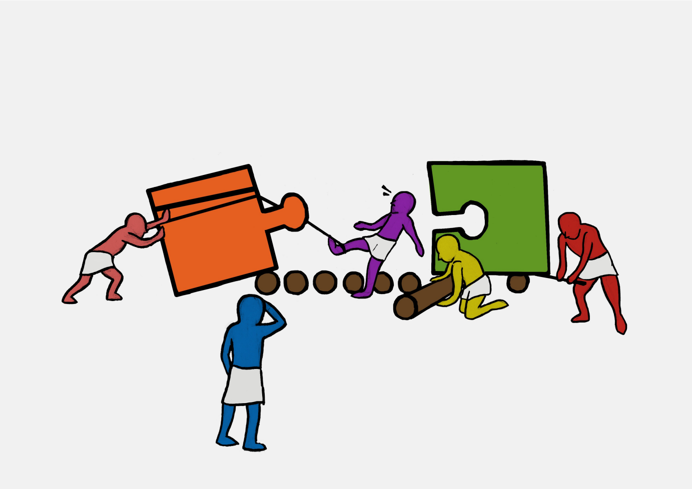


Gradient Boosting: illustration



Gradient Boosting: illustration

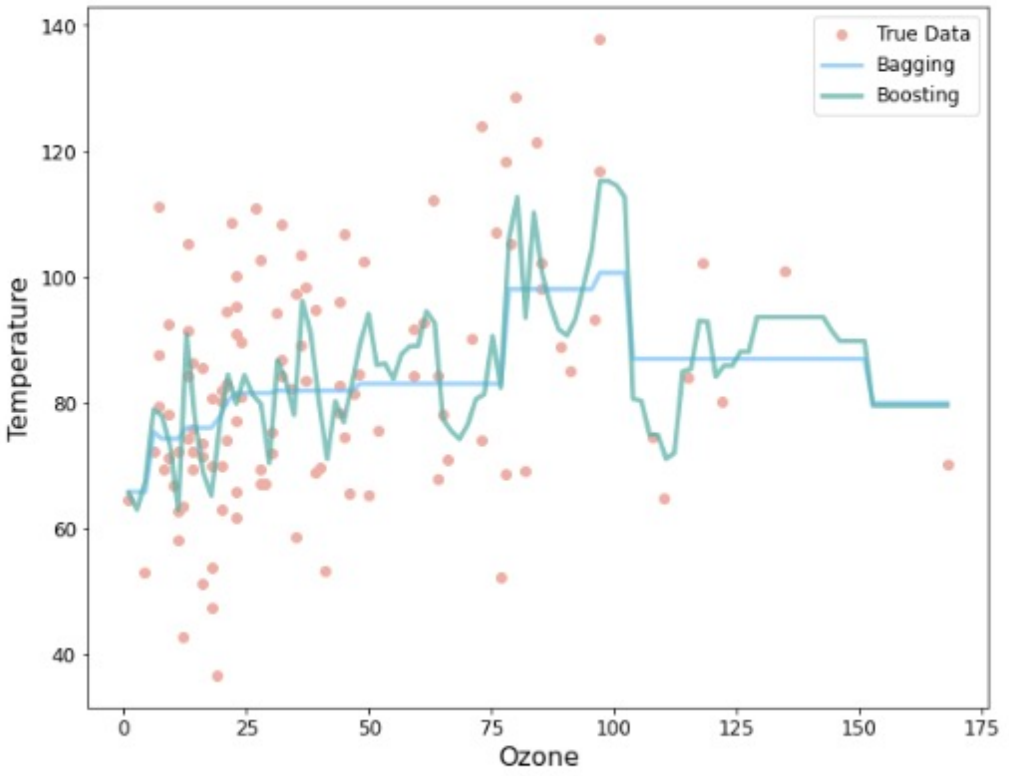




Description

👨‍🏫 Exercise: Regression with Boosting

The goal of this exercise is to understand *Gradient Boosting Regression.*



Instructions: