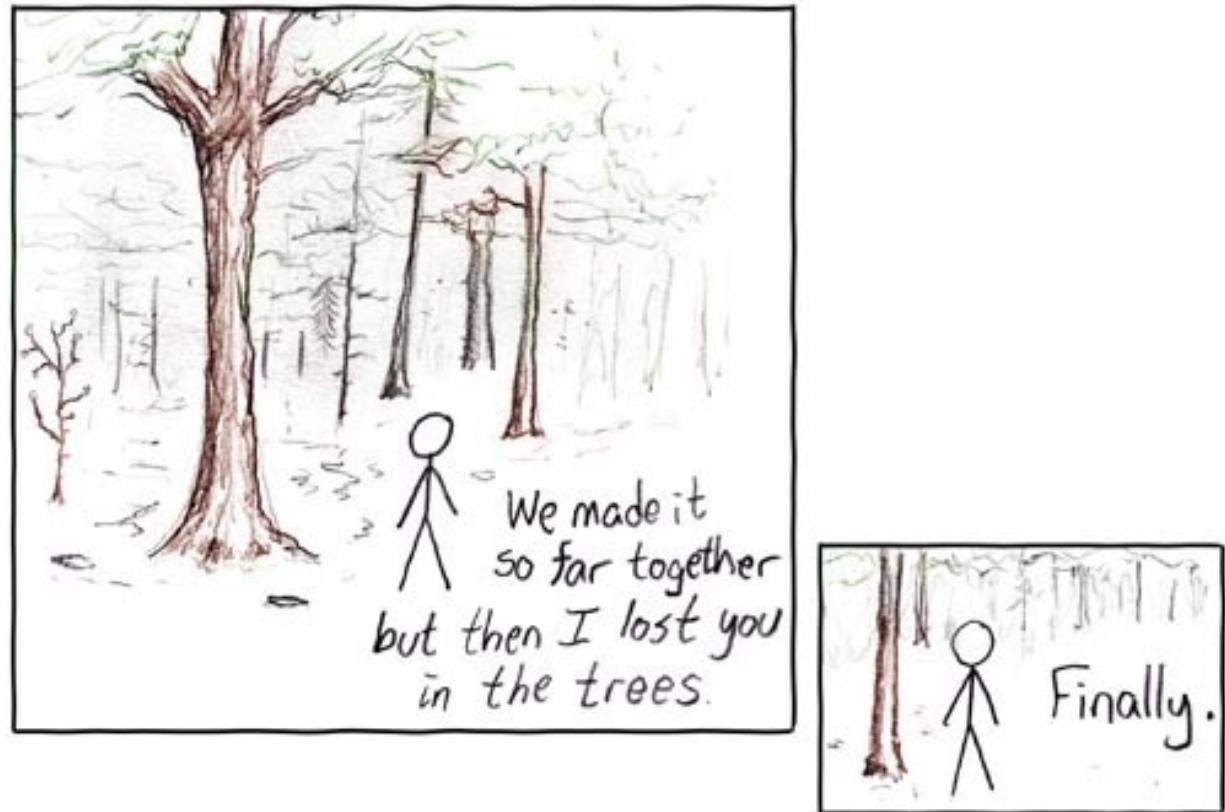


# Decision trees and random forests



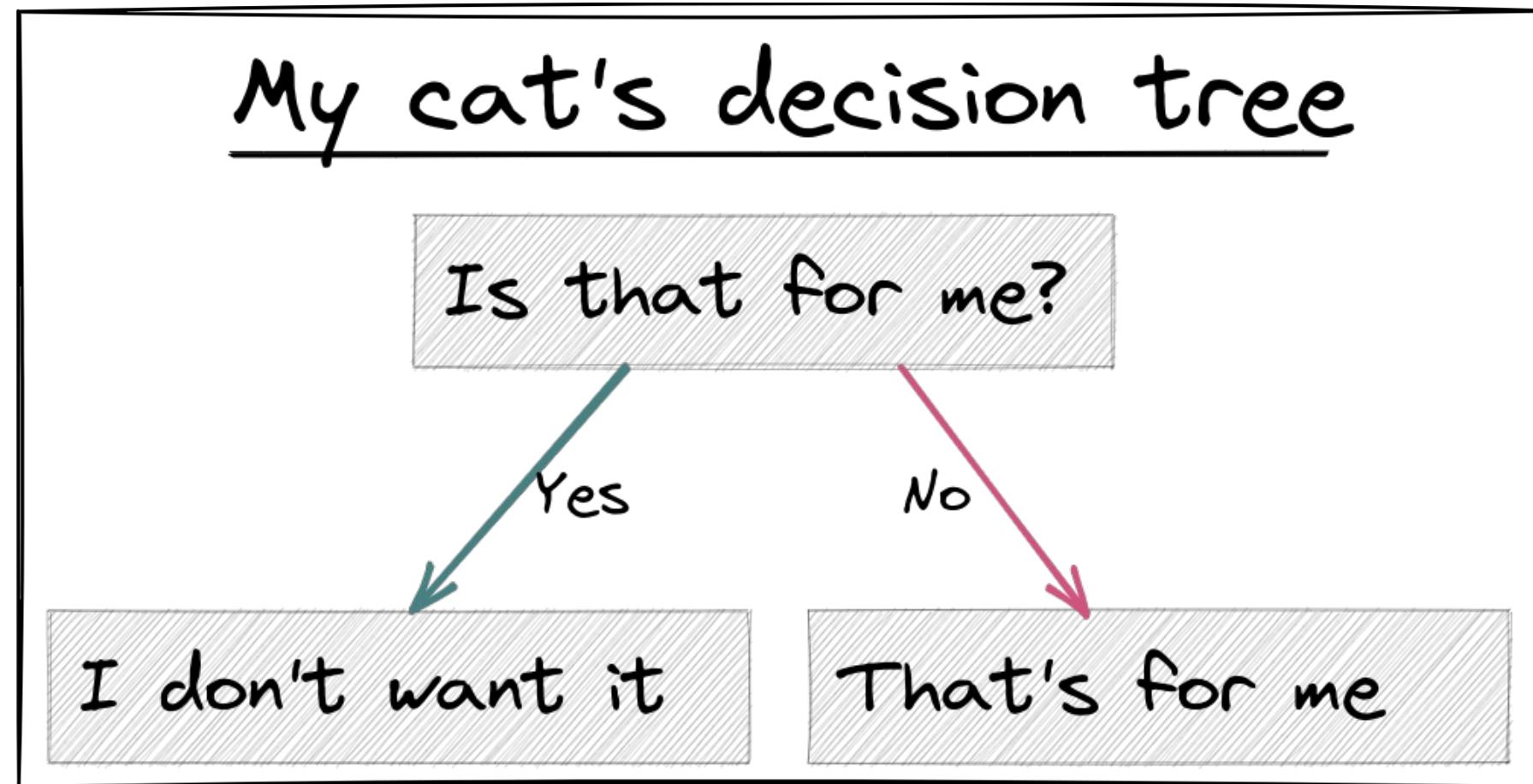
Week 11

Middlesex University Dubai;  
CST4050; Instructor: Ivan Reznikov

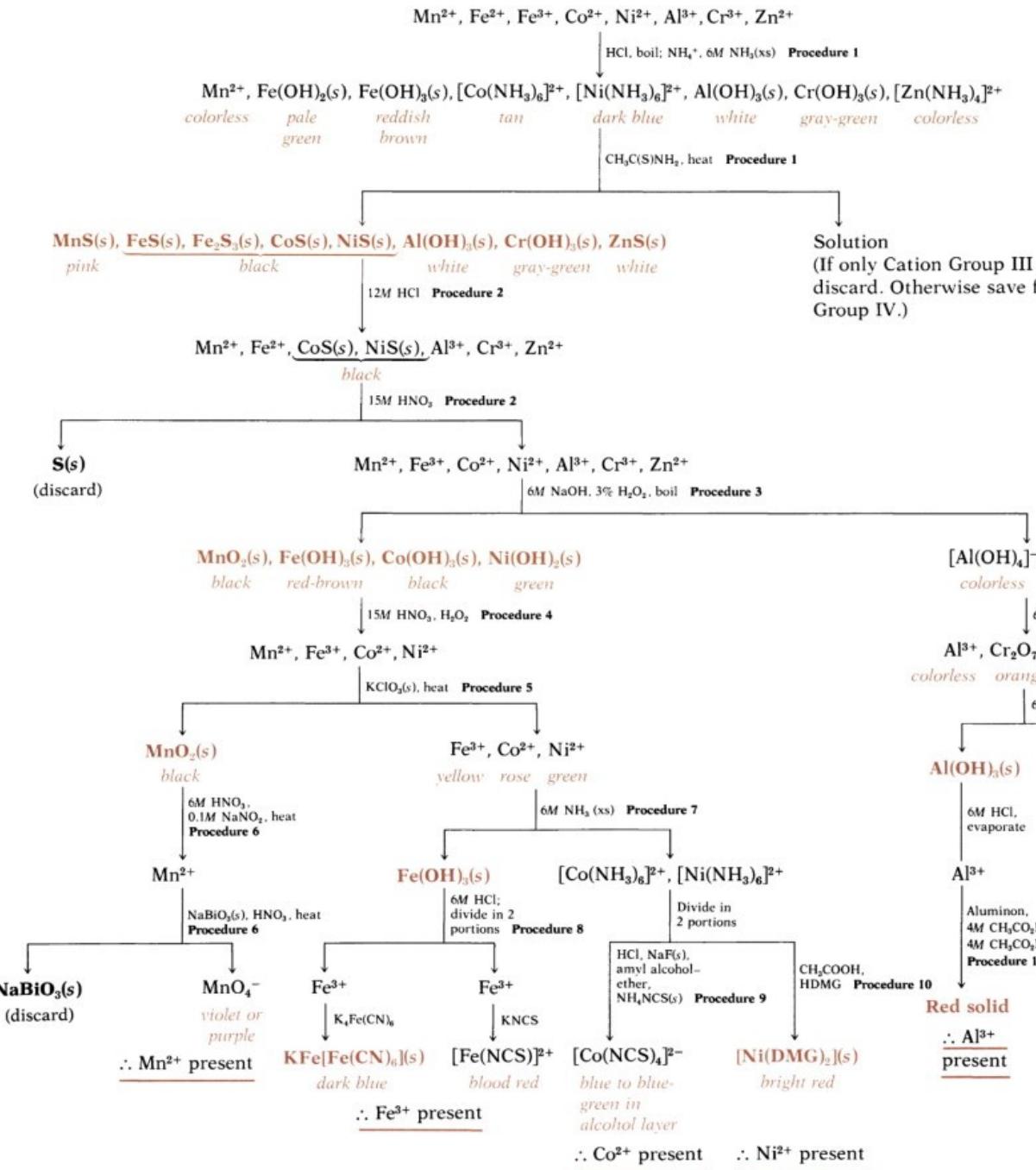
# Plan

- Real-life decision trees
- Pros and cons of decision trees
- Random Forest algorithm

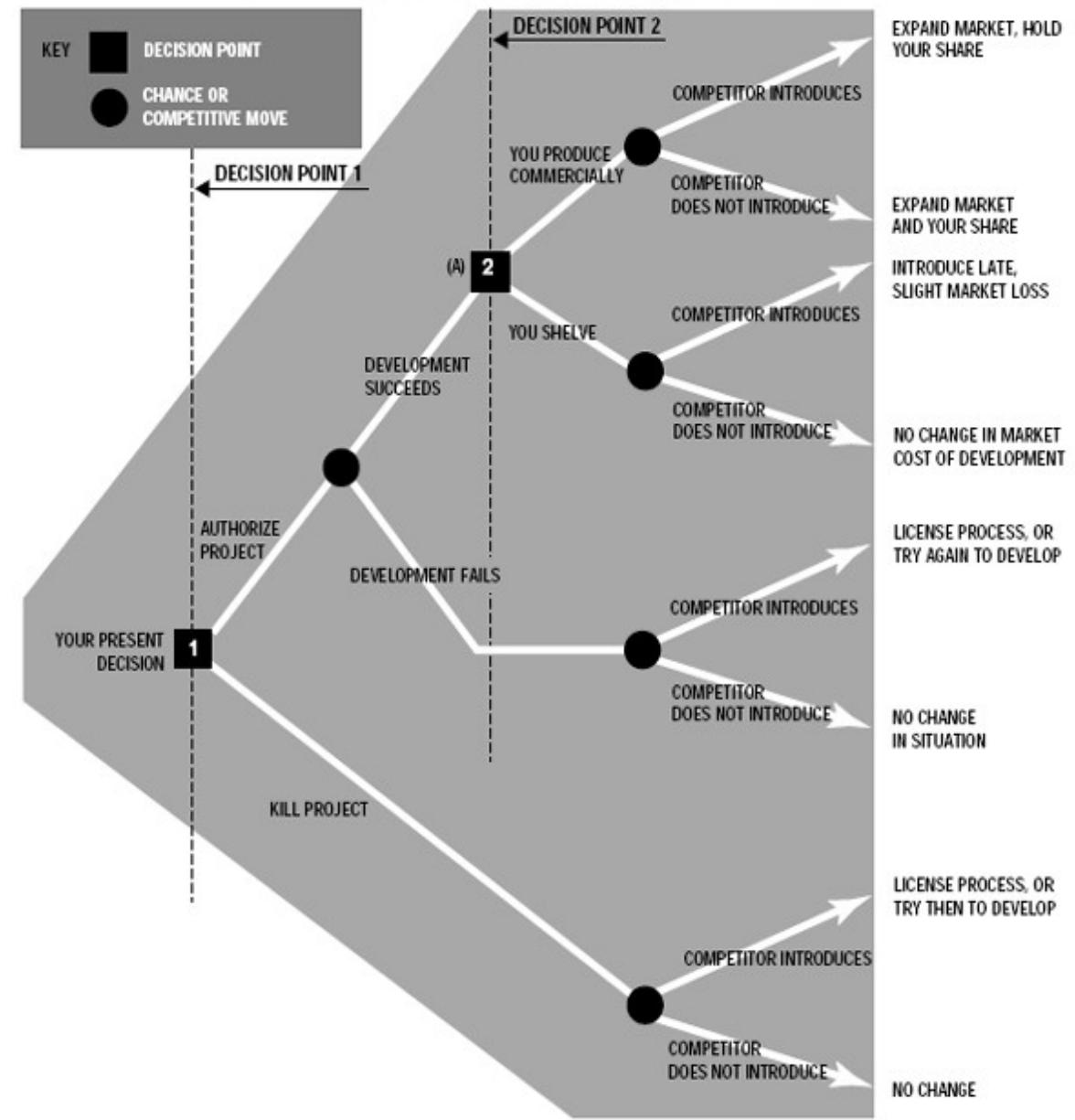
# Decision trees concept



# Decision trees examples



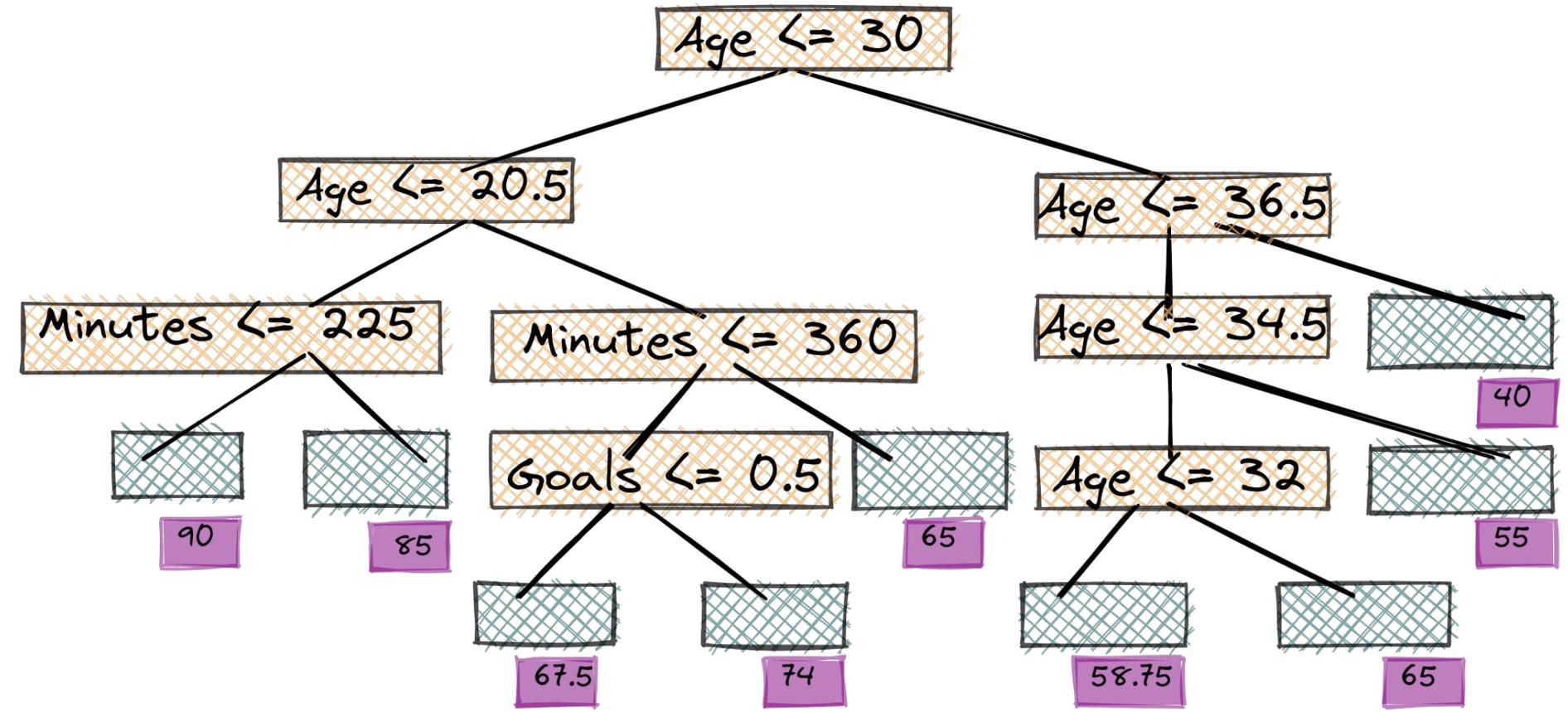
**EXHIBIT II. Decision Tree with Chains of Actions and Events**



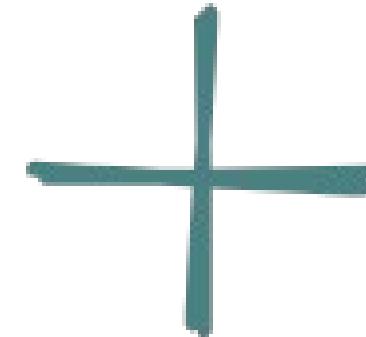
# Decision tree: implementation

Original data

target	Overall Rating	Age	Minutes Played	Goals	...
	40	37	360	0	0
	50	30	900	1	2
	55	36	360	0	0
	55	31	180	0	3
	60	35	180	2	0
	60	31	90	3	1
	60	28	180	6	1
	60	24	90	3	0
	65	33	180	0	0
	65	30	360	2	0
	65	27	90	0	0
	65	23	540	2	4
	70	34	90	1	2
	70	29	90	3	1
	70	22	180	0	0
	75	25	180	3	0
	75	21	90	1	3
	80	27	90	3	1
	85	20	360	4	1
	90	18	90	7	1



# Decision Trees



Easy to build

Easy to use

Easy to interpret results



Highly biased => inaccurate

# Random Forests



Step 1. Run a lot of trees:

Step 1.1. Create bootstrapped dataset

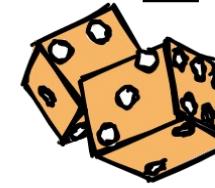
Step 1.2. Create a decision tree with random set of features at each step

Step 2. Aggregate results

Step 3. Evaluate RF model

Step 4. Optimize RF model

# Step 1.1 Bootstrapping



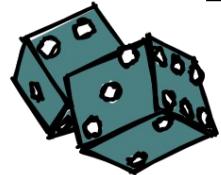
Original data

target	Overall Rating	Age	Minutes Played	Goals	...
	40	37	360	0	0
	50	30	900	1	2
1	55	36	360	0	0
	55	31	180	0	3
	60	35	180	2	0
	60	31	90	3	1
	60	28	180	6	1
	60	24	90	3	0
	65	33	180	0	0
	65	30	360	2	0
	65	27	90	0	0
	65	23	540	2	4
	70	34	90	1	2
1	70	29	90	3	1
	70	22	180	0	0
	75	25	180	3	0
1	75	21	90	1	3
	80	27	90	3	1
	85	20	360	4	1
	90	18	90	7	1

Bootstrapped data

target	Overall Rating	Age	Minutes Played	Goals	...
	40	37	360	0	0
	55	36	360	0	0
1	55	36	360	0	0
	55	31	180	0	3
	60	31	90	3	1
	60	31	90	3	1
	60	31	90	3	1
	65	33	180	0	0
	65	27	90	0	0
	65	23	540	2	4
1	70	29	90	3	1
1	70	29	90	3	1
	70	22	180	0	0
1	75	21	90	1	3
1	75	21	90	1	3
	80	27	90	3	1
	85	20	360	4	1
	85	20	360	4	1
	85	20	360	4	1
	90	18	90	7	1

# Step 1.2a Build decision tree



Split candidates:  
→  
- Age  
- Goals

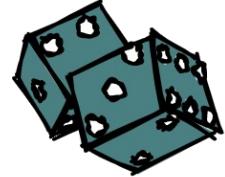
Bootstrapped data

target	Overall Rating	Age	Minutes Played	Goals	...	
	40	37	360	0	0	x1
	55	36	360	0	0	x2
	55	36	360	0	0	x2
	55	31	180	0	3	x1
	60	31	90	3	1	x3
	60	31	90	3	1	x3
	60	31	90	3	1	x3
	65	33	180	0	0	x1
	65	27	90	0	0	x1
	65	23	540	2	4	x1
	70	29	90	3	1	x2
	70	29	90	3	1	x2
	70	22	180	0	0	x1
	75	21	90	1	3	x2
	75	21	90	1	3	x2
	80	27	90	3	1	x1
	85	20	360	4	1	x3
	85	20	360	4	1	x3
	85	20	360	4	1	x3
	90	18	90	7	1	x1

Best to split on: Age

Age  $\leq 30$

# Step 1.2b Build decision tree



Split candidates:

- Goals
- Minutes Played

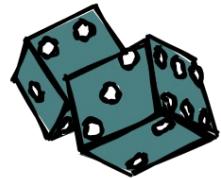
target	Overall Rating	Age	Minutes Played	Goals	...	
	40	37	360	0	0	x1
	55	36	360	0	0	x2
	55	36	360	0	0	x2
	55	31	180	0	3	x1
	60	31	90	3	1	x3
	60	31	90	3	1	x3
	60	31	90	3	1	x3
	65	33	180	0	0	x1
	65	27	90	0	0	x1
	65	23	540	2	4	x1
	70	29	90	3	1	x2
	70	29	90	3	1	x2
	70	22	180	0	0	x1
	75	21	90	1	3	x2
	75	21	90	1	3	x2
	80	27	90	3	1	x1
	85	20	360	4	1	x3
	85	20	360	4	1	x3
	85	20	360	4	1	x3
	90	18	90	7	1	x1

Best to split on: Goals

Age  $\leq 30$

Goals  $\leq 3.5$

# Step 1.2c Build decision tree



Split candidates:  
→

- Age
- ...

target	Overall Rating	Age	Minutes Played	Goals	...
	40	37	360	0	0
	55	36	360	0	0
	55	36	360	0	0
	55	31	180	0	3
	60	31	90	3	1
	60	31	90	3	1
	60	31	90	3	1
	65	33	180	0	0
	65	27	90	0	0
	65	23	540	2	4
	70	29	90	3	1
	70	29	90	3	1
	70	22	180	0	0
	75	21	90	1	3
	75	21	90	1	3
	80	27	90	3	1
	85	20	360	4	1
	85	20	360	4	1
	85	20	360	4	1
	90	18	90	7	1

Best to split on: Age

Age  $\leq 30$

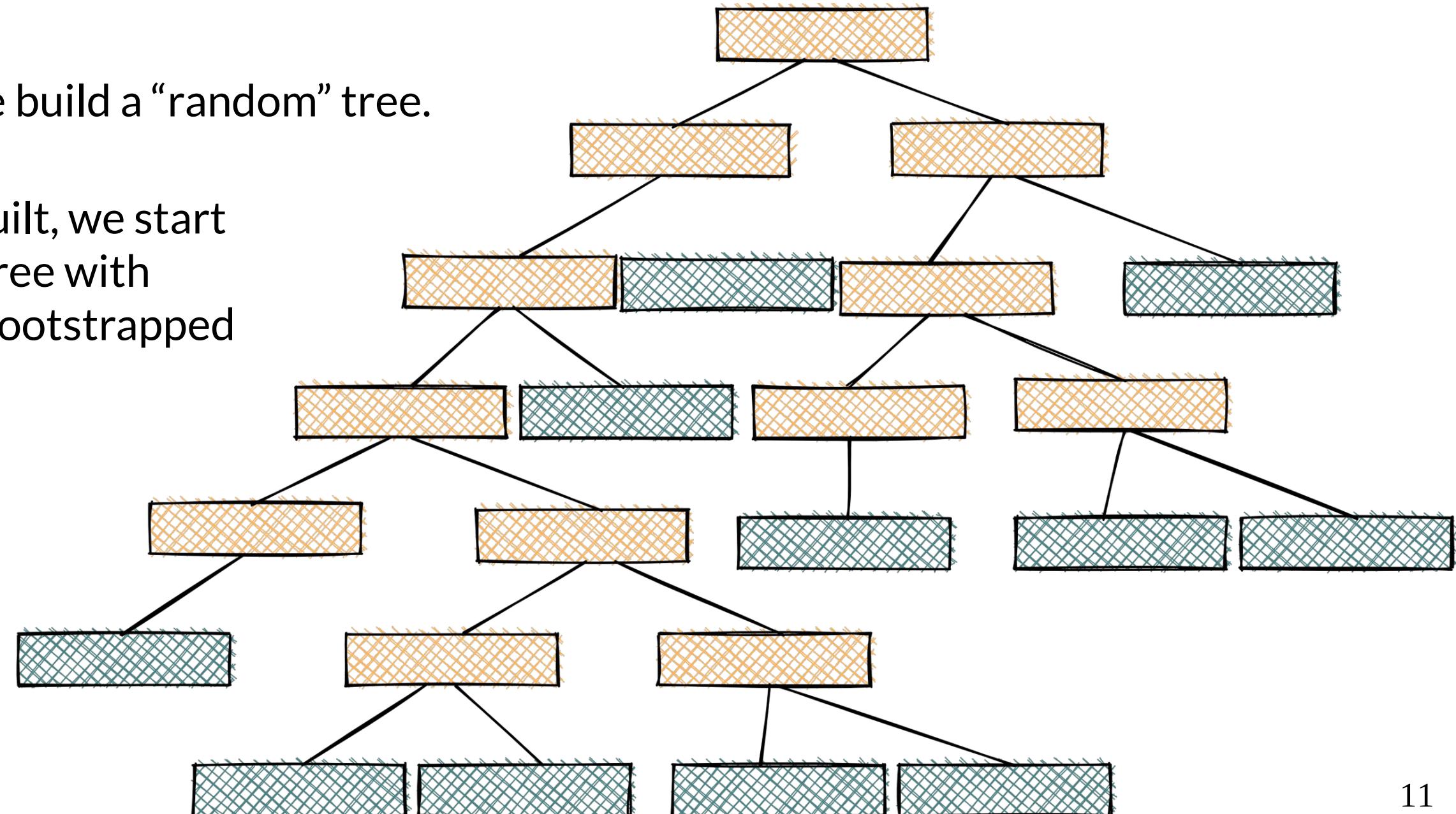
Goals  $\leq 3.5$

Age  $\leq 36.5$

# **Step 1.2d Build decision tree**

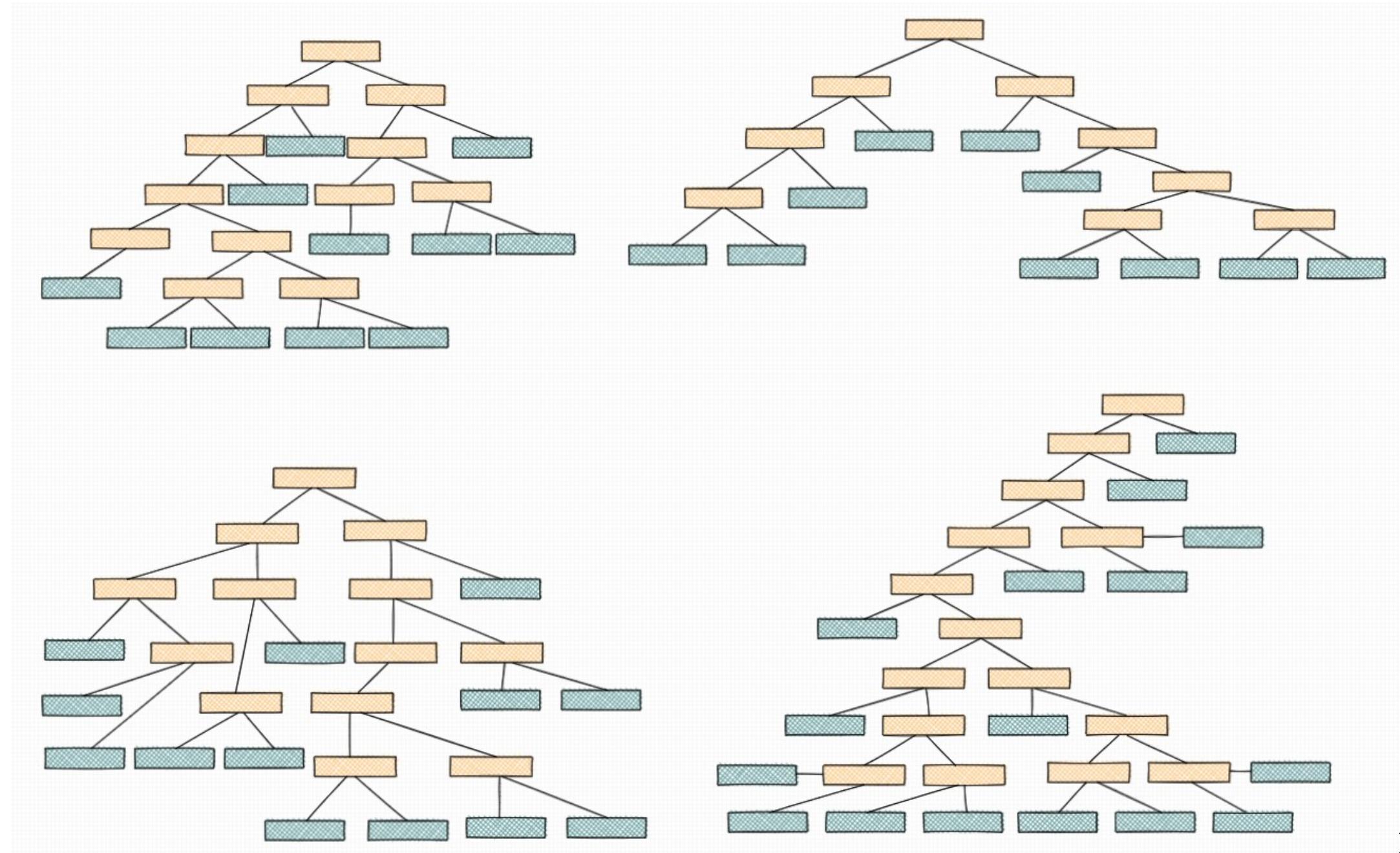
Step-by-step we build a “random” tree.

Once a tree is built, we start building a new tree with new randomly bootstrapped data.



# Step 1. Build a lot of trees

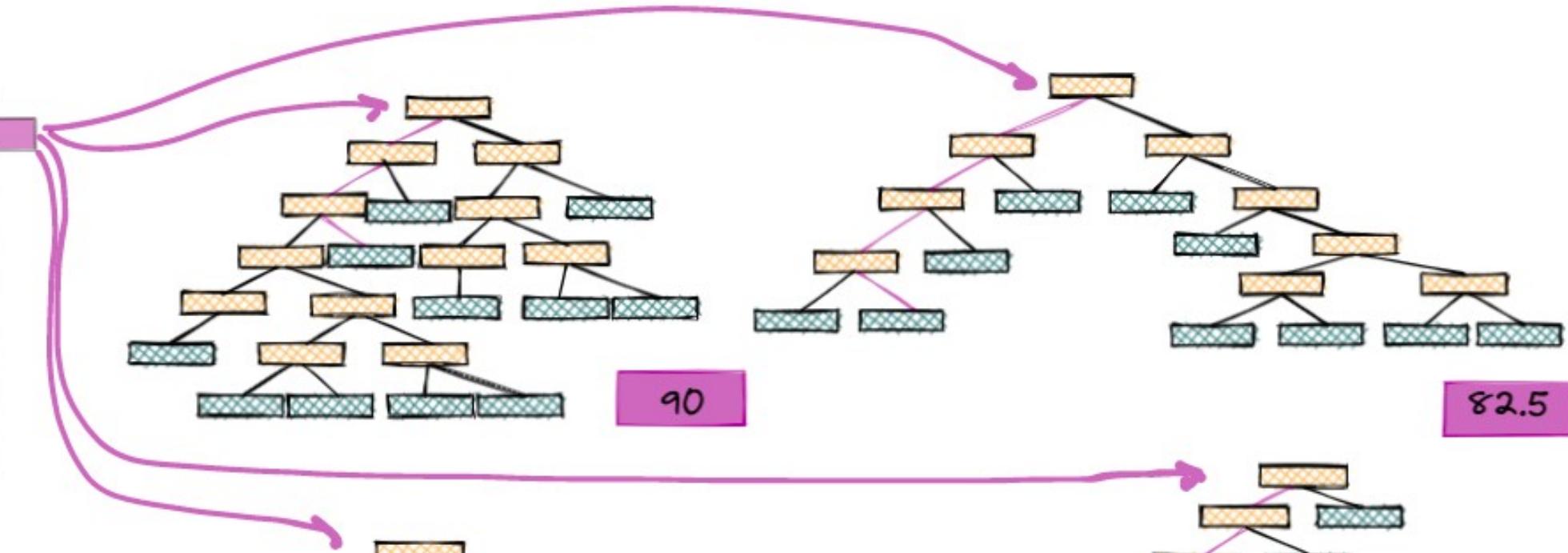
We will result in a  
lot of trees – a  
random forest



# Step 2. Aggregate results

Test data

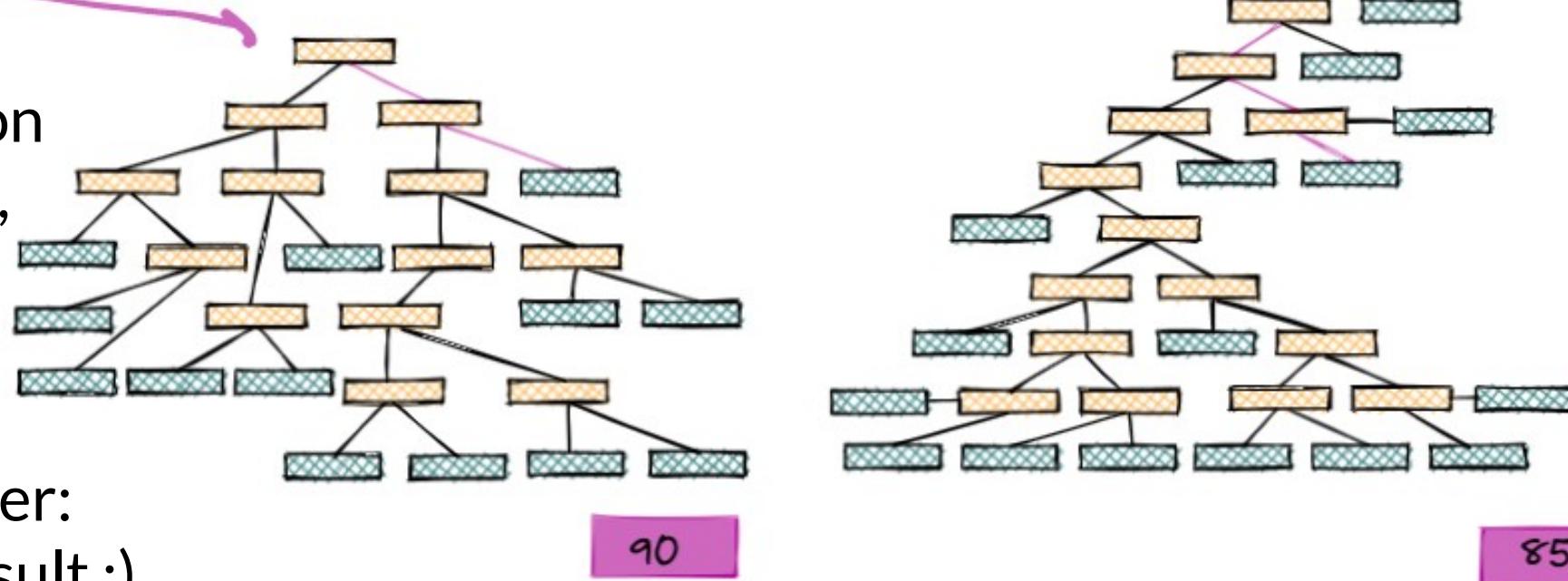
target	Overall Rating	Age	Minutes Played	Goals	...
	29	180	6	0	
	28	90	3	1	
	24	180	1	0	
	35	360	2	1	
	31	90	0	2	
	31	405	2	2	
	23	270	0	3	
	24	90	0	0	
	32	135	0	0	
	33	180	3	1	
	19	90	1	2	
	20	90	3	1	
	26	315	4	5	



Bagging = Bootstrap aggregation

For a regression task like above, we can apply any aggregation function we want – mean, median, mode, etc.

For classification it's even simpler: just count the most common result :)

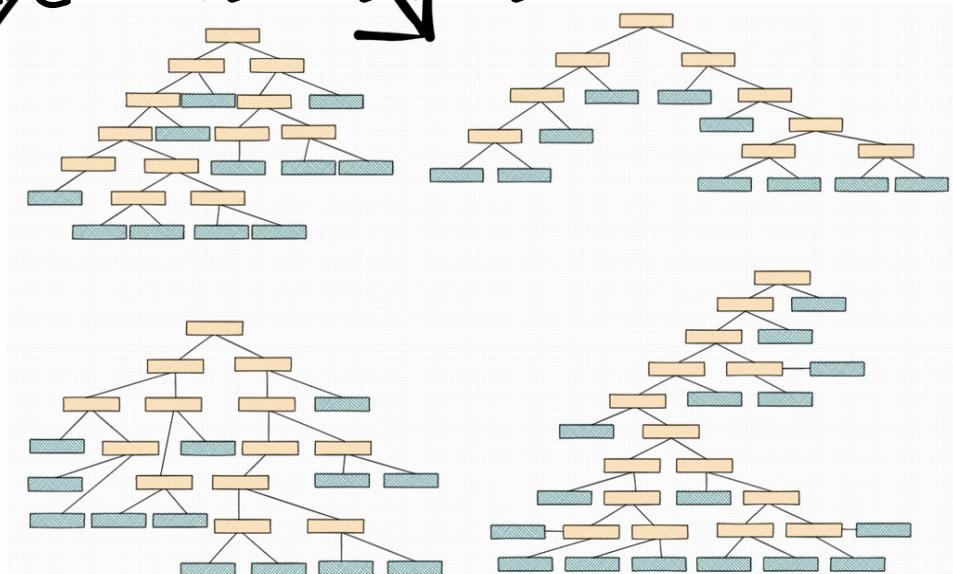


# Step 3. Evaluate RF model

Original data

target	Overall Rating	Age	Minutes Played	Goals	...	x1
	40	37	360	0	0	x1
	50	30	900	1	2	
	55	36	360	0	0	x2
	55	31	180	0	3	x1
	60	35	180	2	0	
	60	31	90	3	1	x3
	60	28	180	6	1	
	60	24	90	3	0	
	65	33	180	0	0	x1
	65	30	360	2	0	
	65	27	90	0	0	x1
	65	23	540	2	4	x1
	70	34	90	1	2	
	70	29	90	3	1	x2
	70	22	180	0	0	x1
	75	25	180	3	0	
	75	21	90	1	3	x2
	80	27	90	3	1	x1
	85	20	360	4	1	x3
	90	18	90	7	1	x1

Each OOB sample we run on trees that were built without using it



Out-of-Bag (OOB) dataset can be used to calculate model accuracy

Aggregation and return result

Evaluate with RMSE, etc.

# Step 4. Optimizing RF model

- Decision Trees tuning
  - Trees parameters: max depth, split model, etc
  - Node/Leaf parameters: min split samples, min leaf samples
- Random Forest tuning:
  - Number of trees built
  - Number of features used during bagging
  - Metric for Out-of-Box evaluation