ICS 504: Machine Learning
Lecture 6
Ensemble Learning: Random Forests

Dr. Caroline Sabty

caroline.sabty@giu-uni.de

Faculty of Informatics and Computer Science

German International University in Cairo

Acknowledgment

The course and the slides are based on the slides of Dr. Seif Eldawlatly and based on the course created by Prof. Jose Portilla

ICS 603: Advanced Machine Learning

- Textbooks
  - Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006 (You can find here: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf)
  - Machine Learning, Tom Mitchell, McGraw Hill, 1997
  - Neural Networks and Learning Machines, Simon Haykin, Pearson, 2008
  - Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2017
  - Research papers

- Office Hours by email

Ensemble Learning

Ensemble Learning

- References:

- “Introduction to Data Mining" by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, First edition (2006) or Second edition (2018), Pearson Education

- “Ensemble Learning: Pattern Classification Using Ensemble Methods,” Lior Rokach (2019), World Scientific Publishing

Ensemble Learning

- The main idea behind the ensemble methodology is to weigh several individual pattern classifiers and combine them, in order to obtain a classifier that outperforms all of them

- The ensemble methodology imitates our second nature to seek several opinions before making any crucial decision

- This is wisdom of the crowds

Random Forest Classifier

Random Forest Classifier

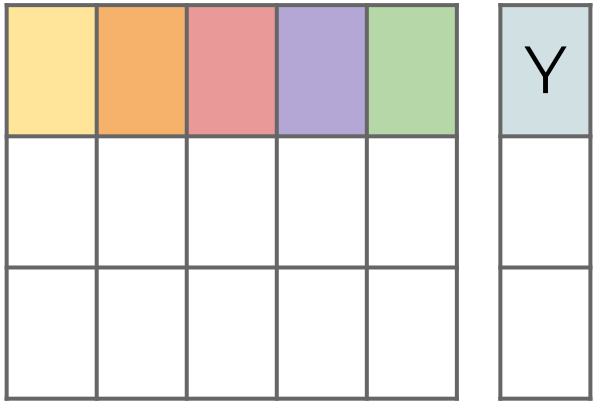
- Random forest is a class of ensemble methods specifically designed for decision tree classifiers
- It uses the Bagging methodology
- It combines the predictions made by multiple decision trees, where each tree is generated based on the values of an independent set of samples
- Different from the original Bagging method, while training each decision tree, rather than choosing the best split among all attributes, N of the attributes are randomly chosen and the best split from among those variables is identified

Random Forests

- Random Forests have the ability to greatly increase the performance based on expanding ideas from the Decision Tree.
- Random Forests are known as **ensemble** learners, since they rely on an ensemble of models (multiple decision trees).
- Why not just continue to use Decision Trees?
- What is the motivation behind Random Forests and how do they improve on Decision Trees?
- Let's think back to the construction of a single decision tree...

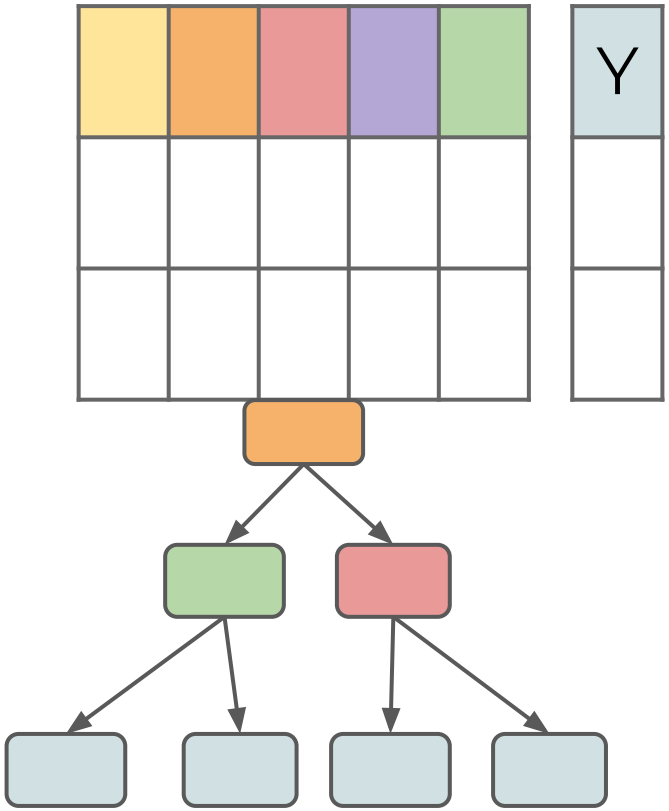
Random Forests

- Imagine a data set with features and label:



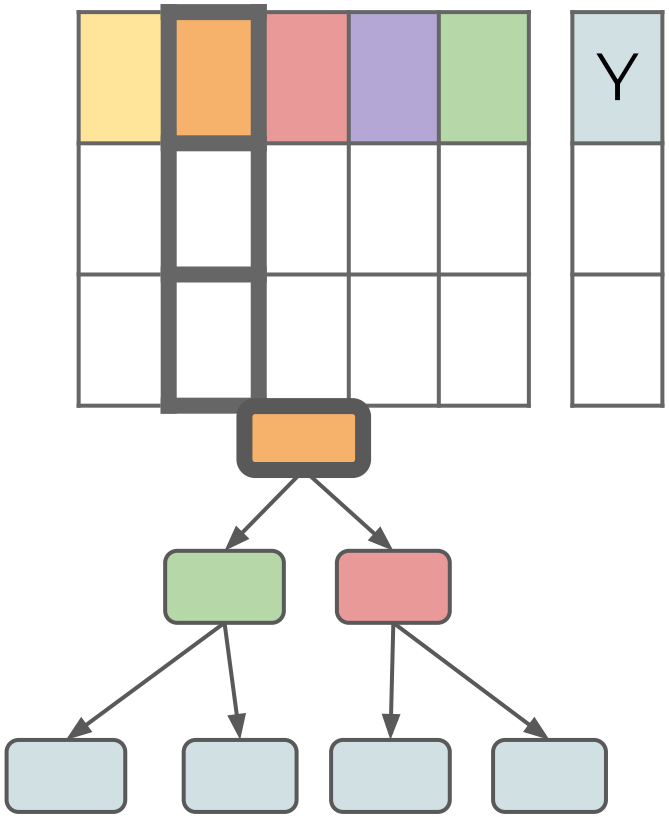
Random Forests

- Decision Tree restricted by gini impurity:



Random Forests

- No guarantee of using all features
- Root node will always be the same
- Root feature has huge influence over tree



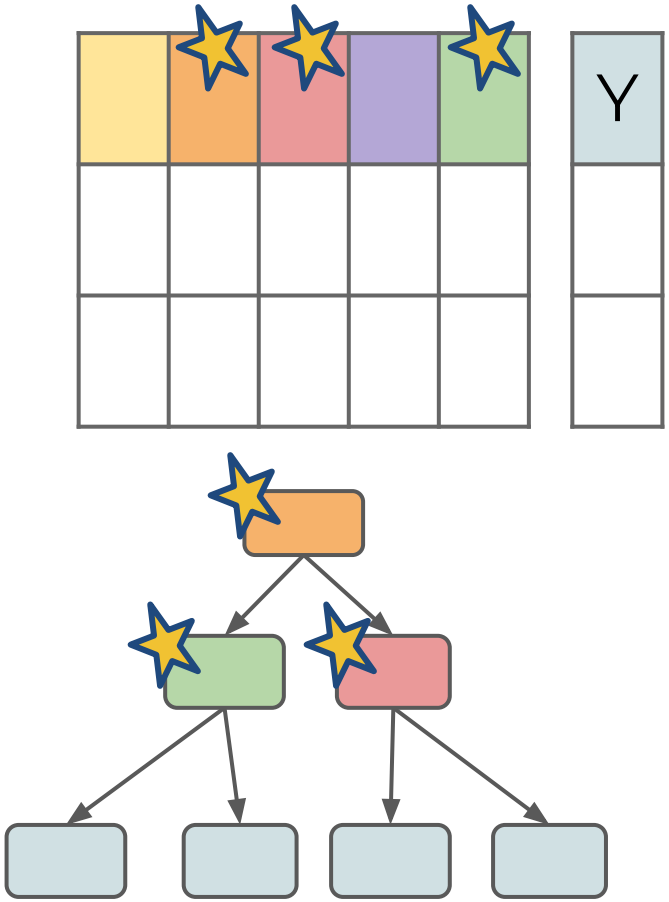
Random Forests

- We could try adjusting rules, such as:
  - Splitting Criterion (Information Gain)
  - Minimum Gini Impurity Decrease
  - Setting Depth Limits
  - Limits on number of terminal leaf nodes
- However for every set of rules there is only one decision tree possible
- Even with all these added hyperparameter adjustments, the single decision tree is still limited:
  - Single feature for root node
  - Splitting criteria can lead to some features not being used
  - Potential for overfitting to data

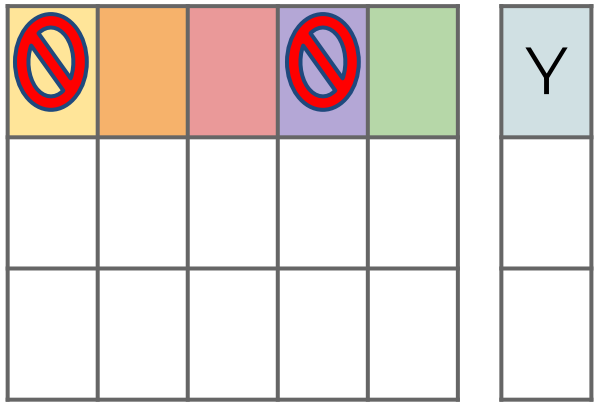
Random Forests History

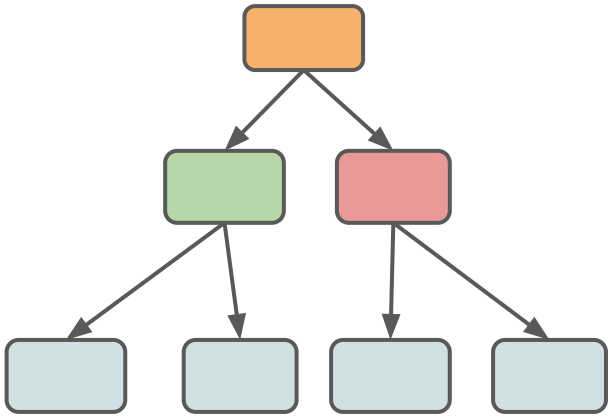
- 1995: Tin Kam Ho presents Random Decision Forests at the 3rd International Conference on Document Analysis and Recognition in Montreal.

Random Forests

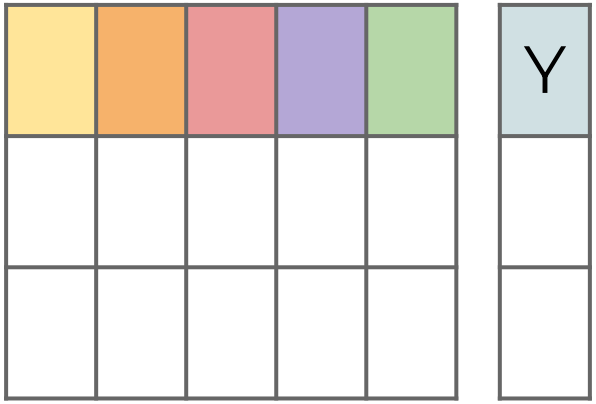


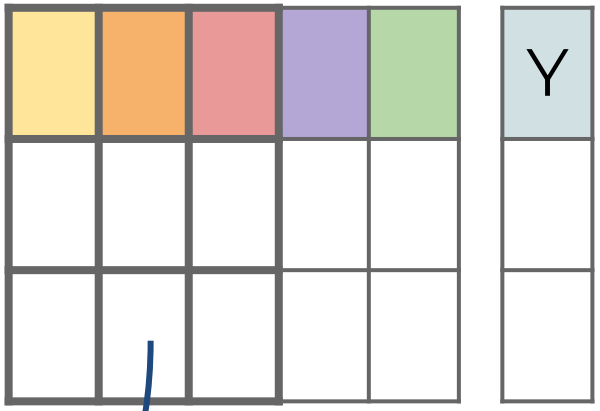
Random Forests

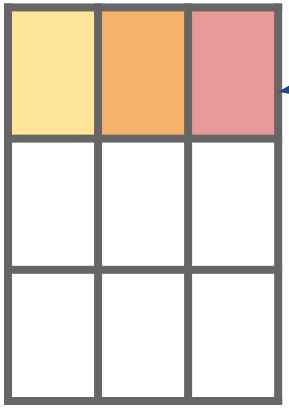




Random Forests





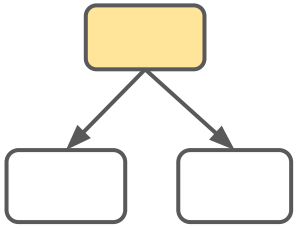


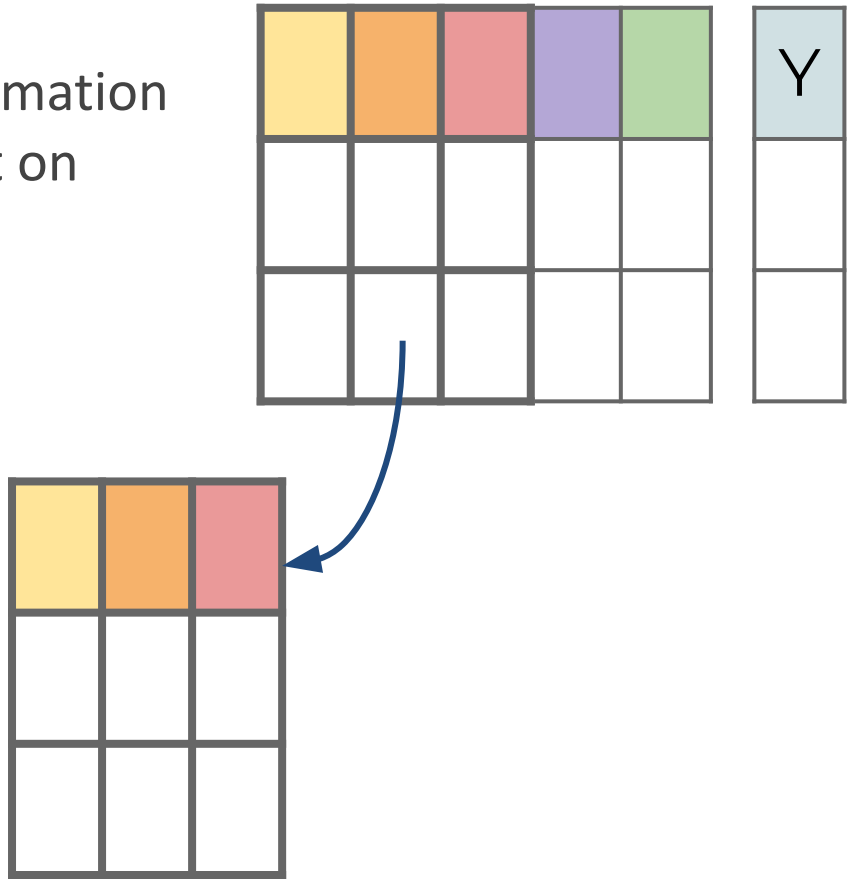
Create subsets of randomly picked features at each potential split

KEY IDEA

Random Forests

- For this subset based on my information will decide about the root to split on

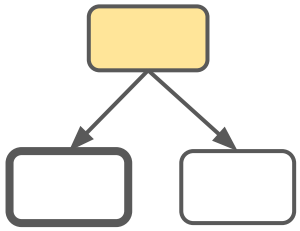


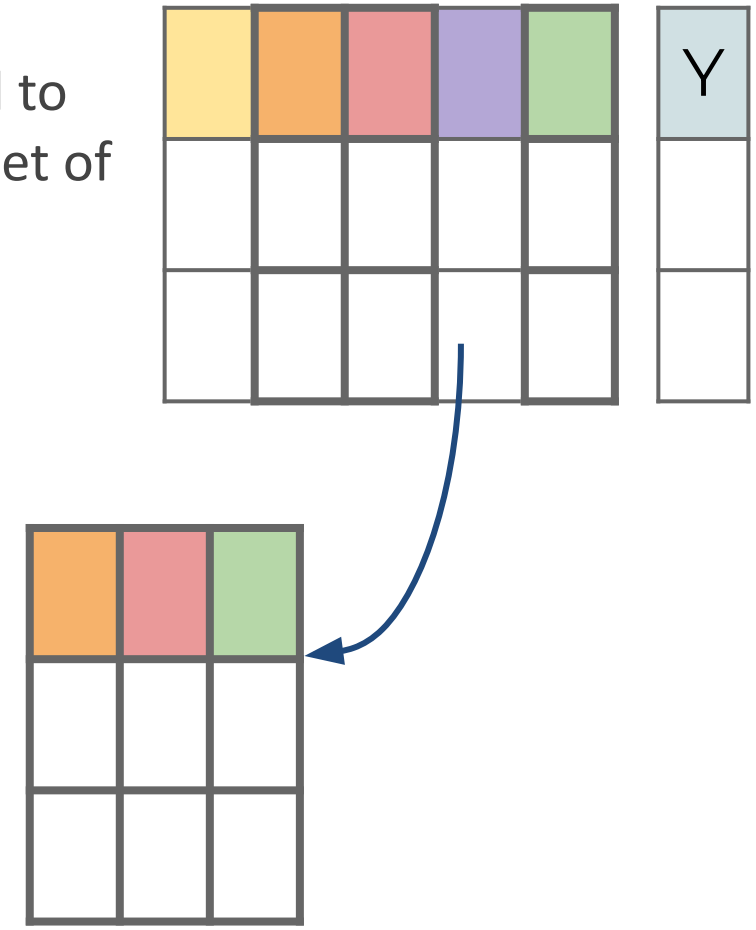


Randomly select set of features

Random Forests

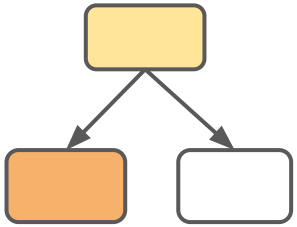
- For another node will decide if I need to split again, I will choose another subset of features

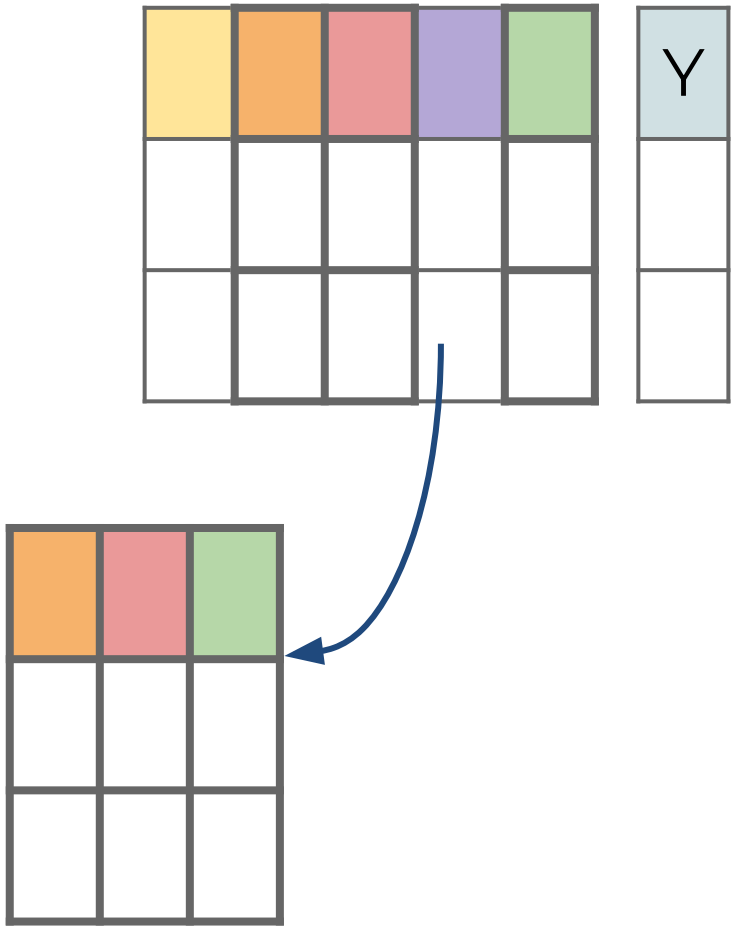




Another random subset of features

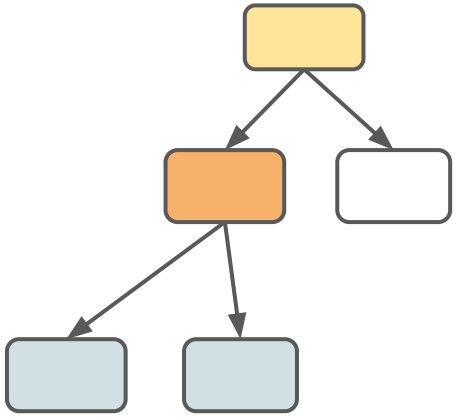
Random Forests

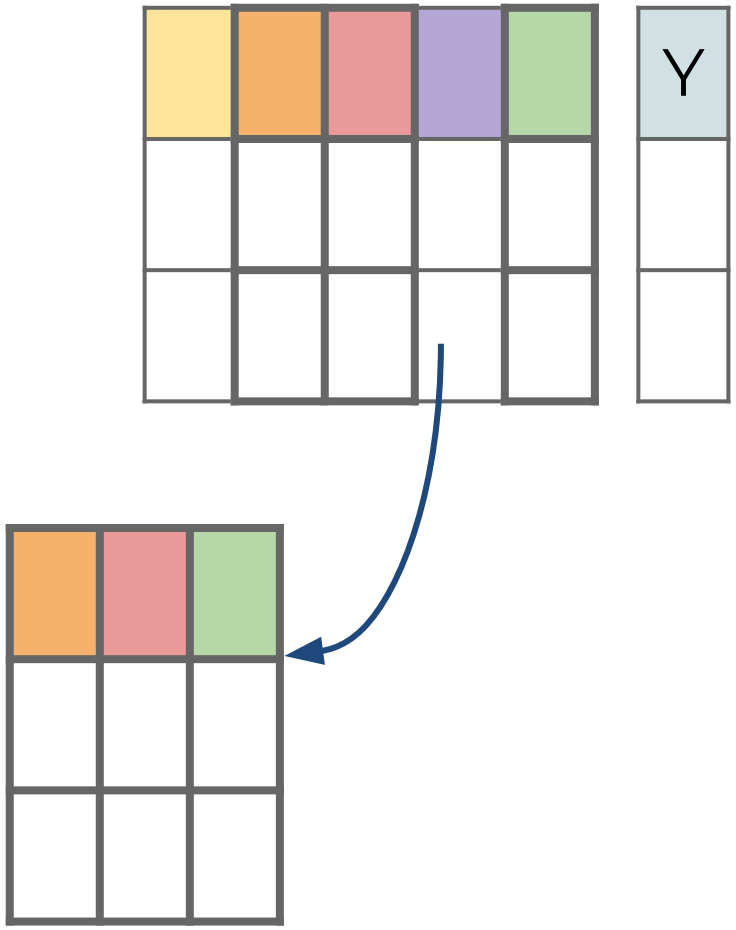




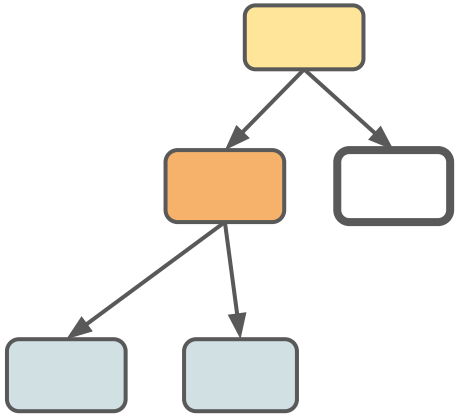
Random Forests

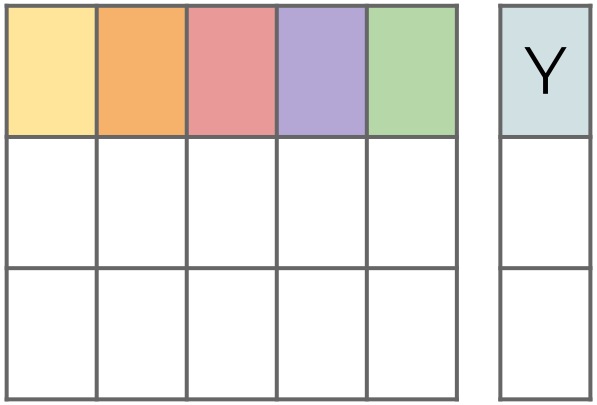
- In case I have enough decrease in Gini impurity I will split to terminal nodes



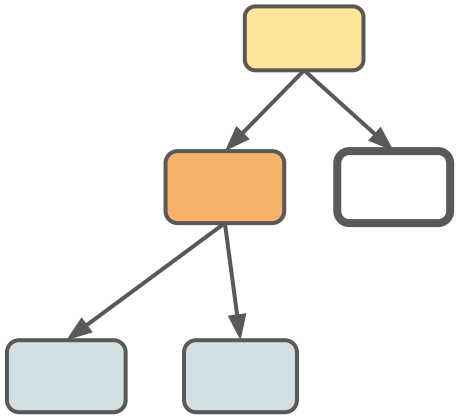


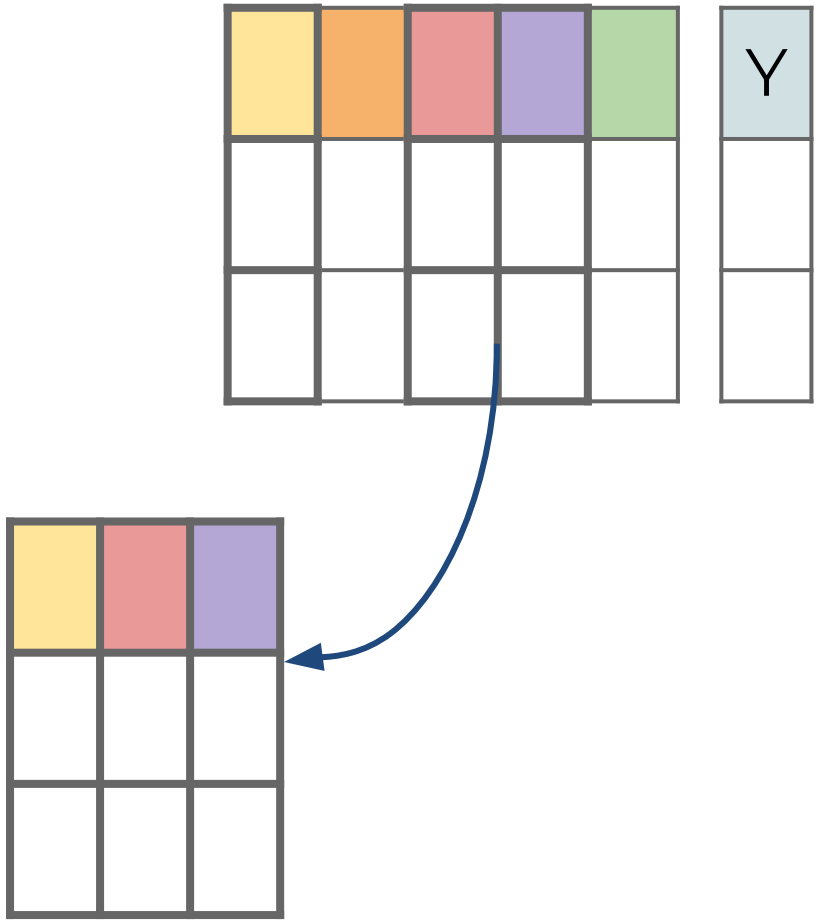
Random Forests





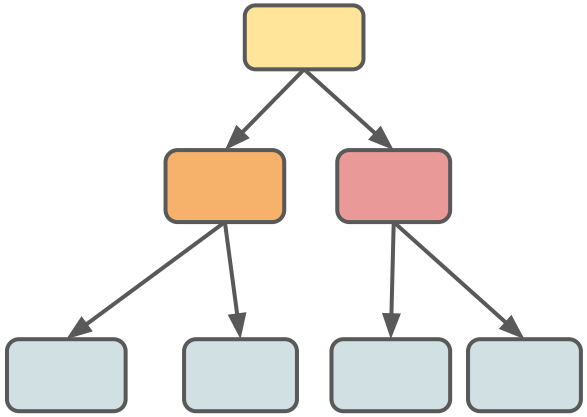
Random Forests

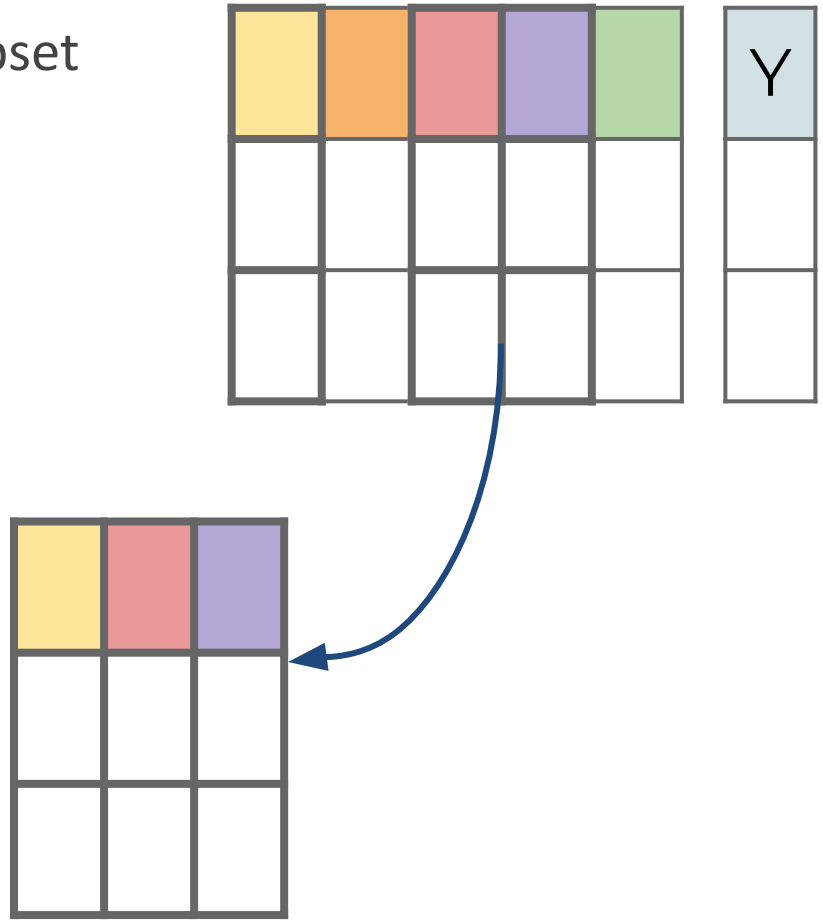




Random Forests

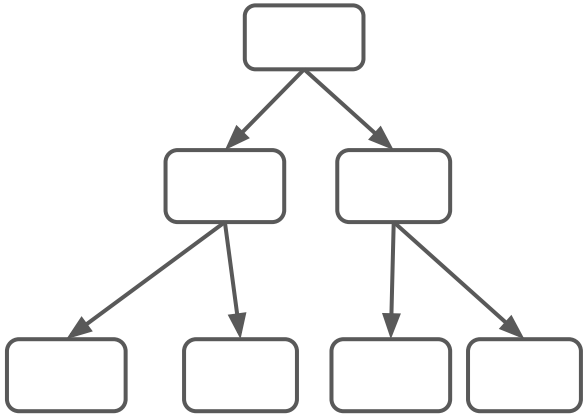
- By choosing randomly selective subset of features we are sure we are not overfitting these trees

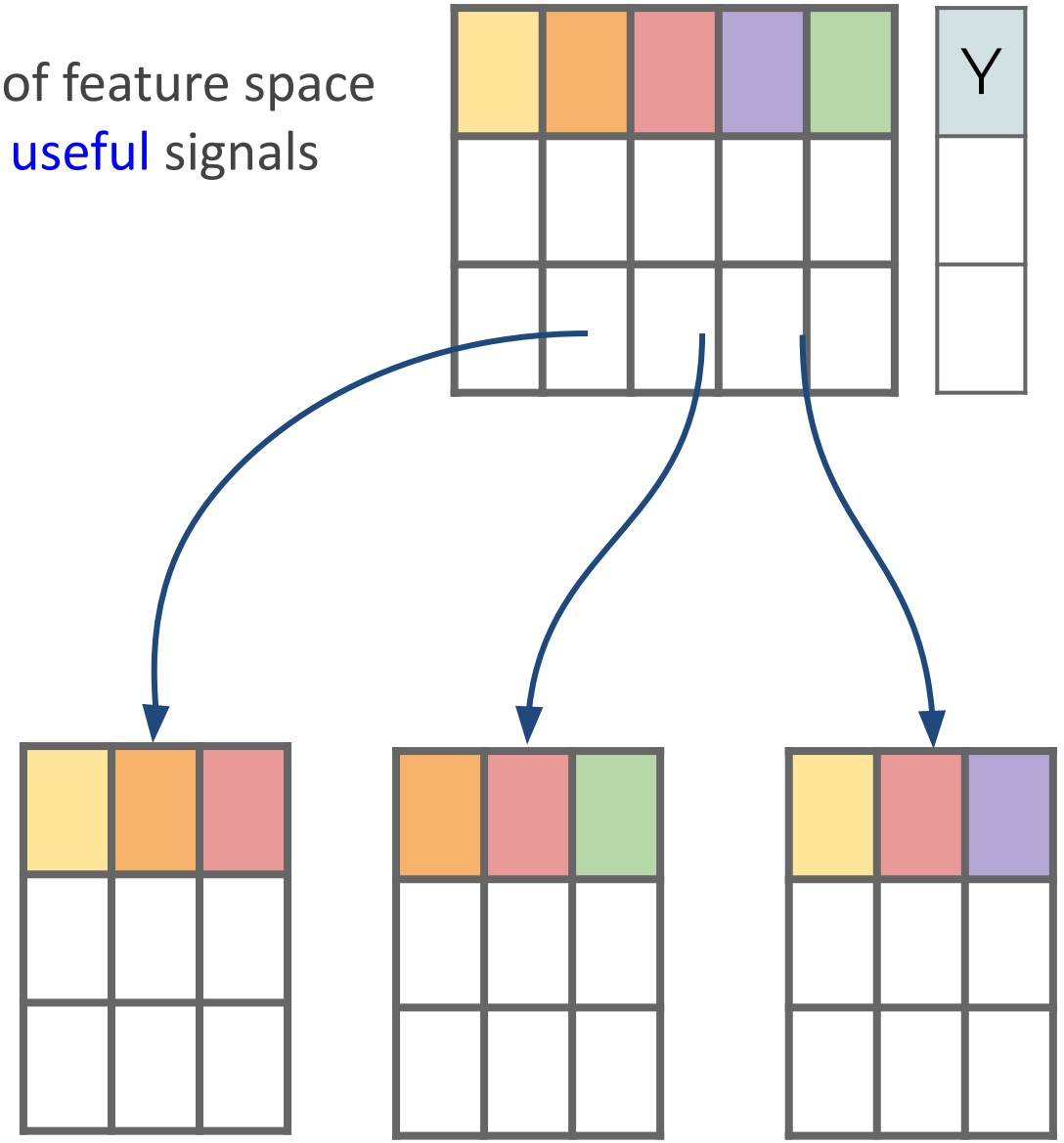




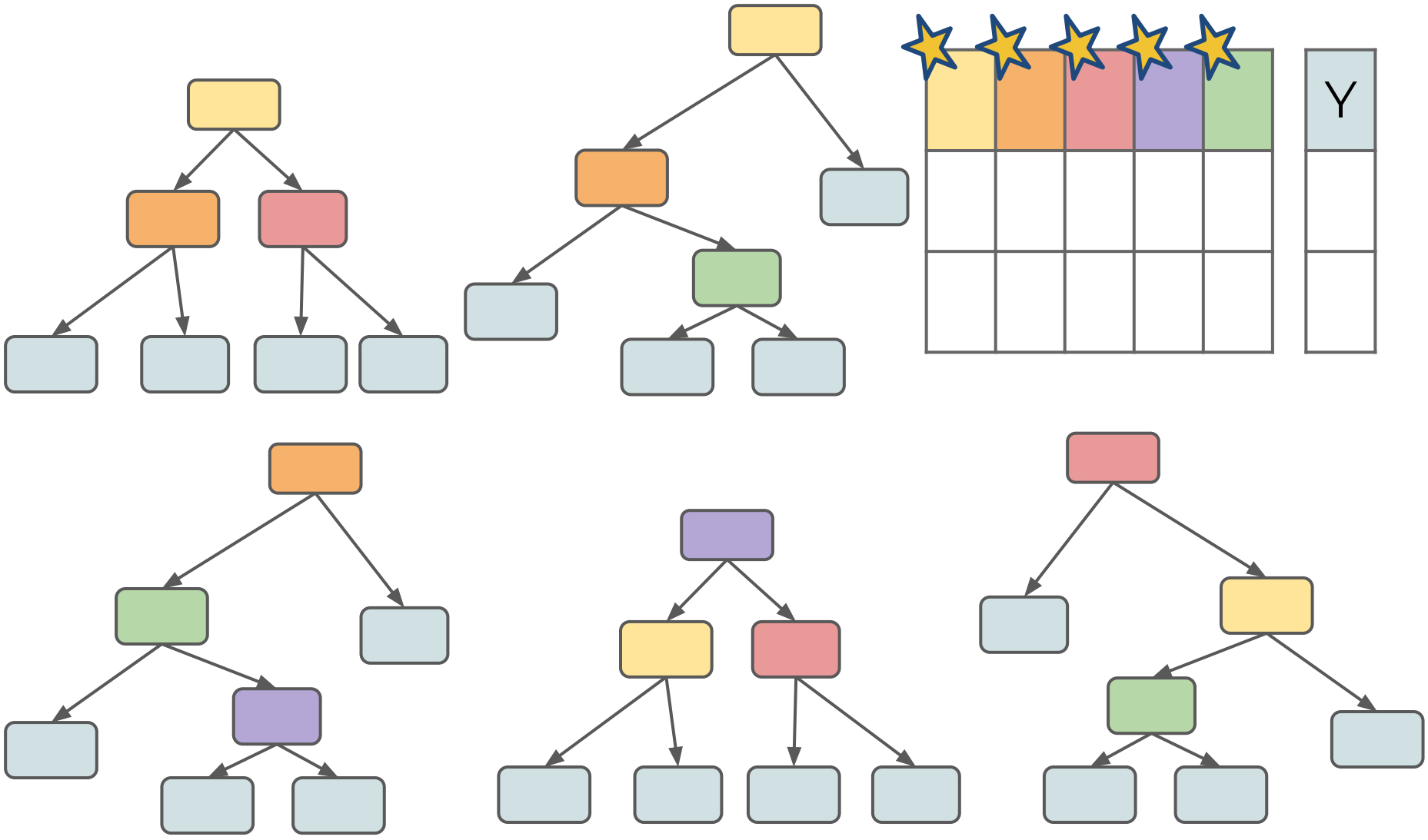
Random Forests

- Explore different aspects of feature space
  use all features that have useful signals

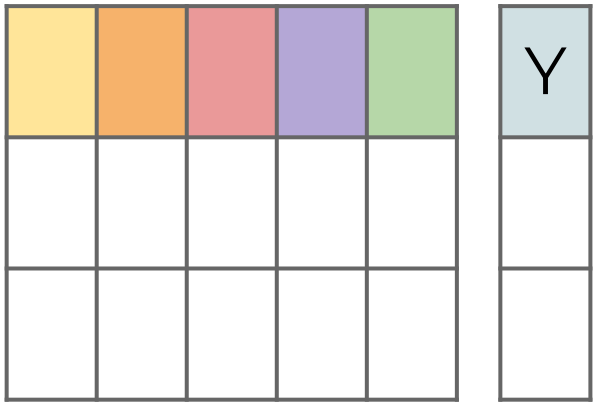


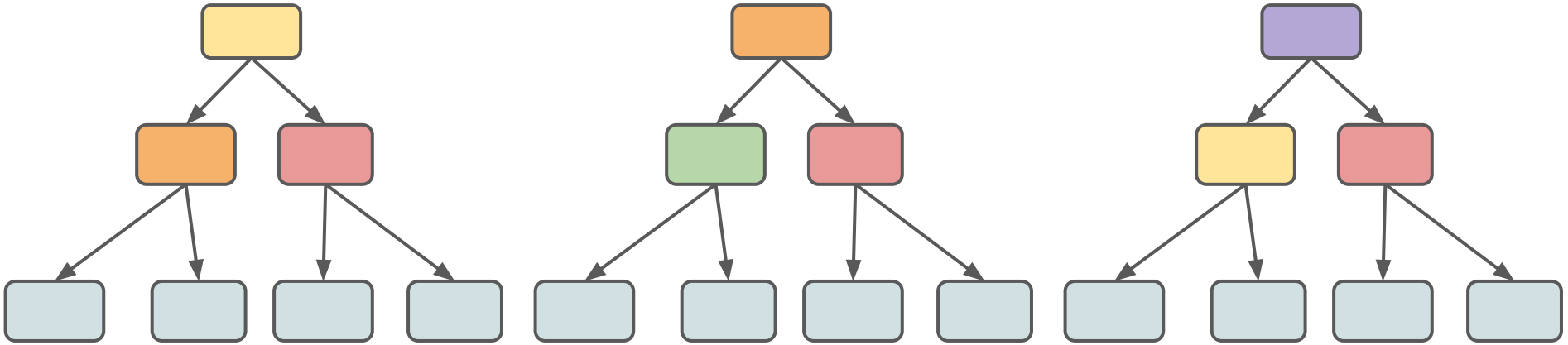


Ensemble of Trees



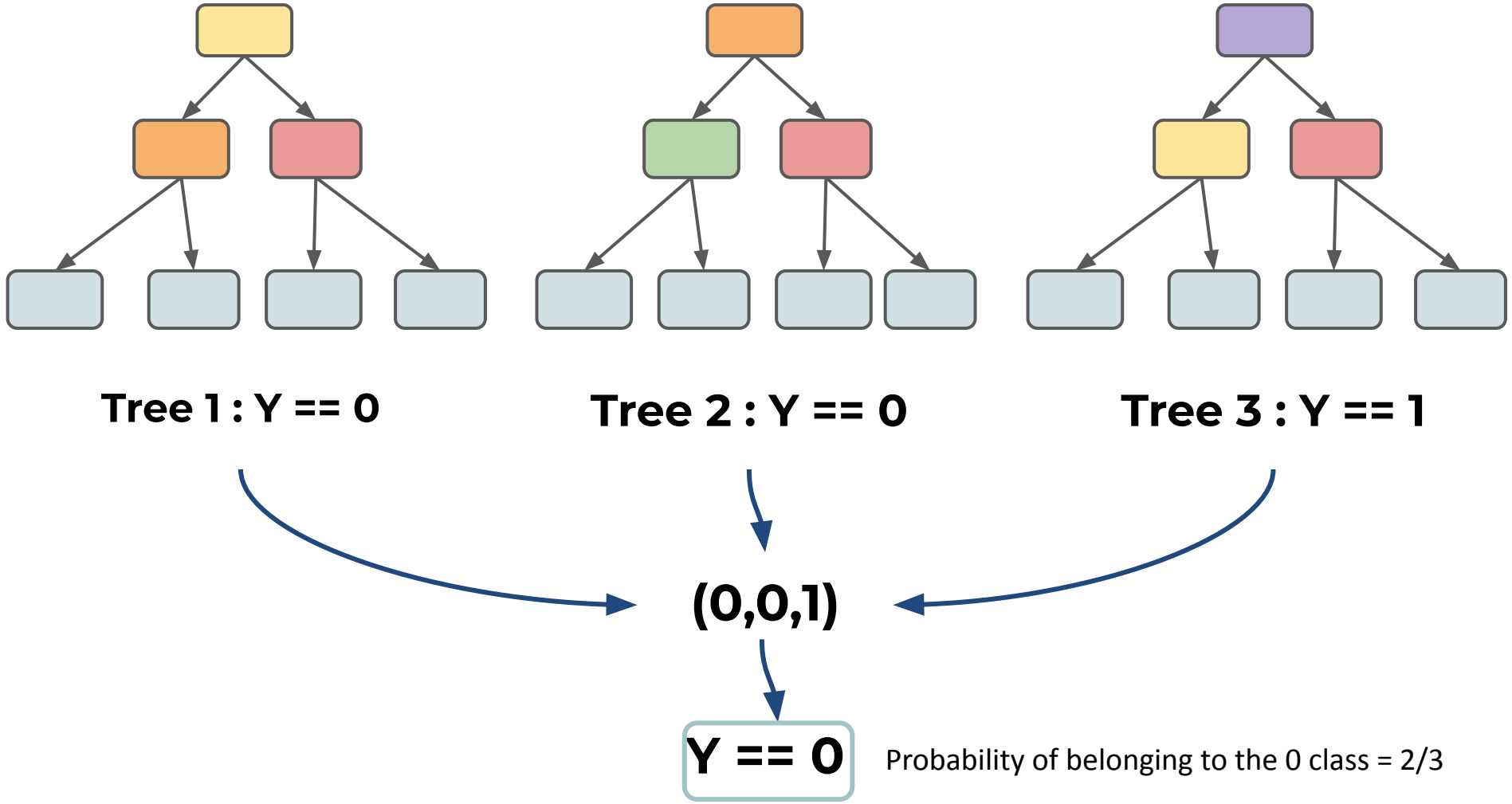
Random Forests



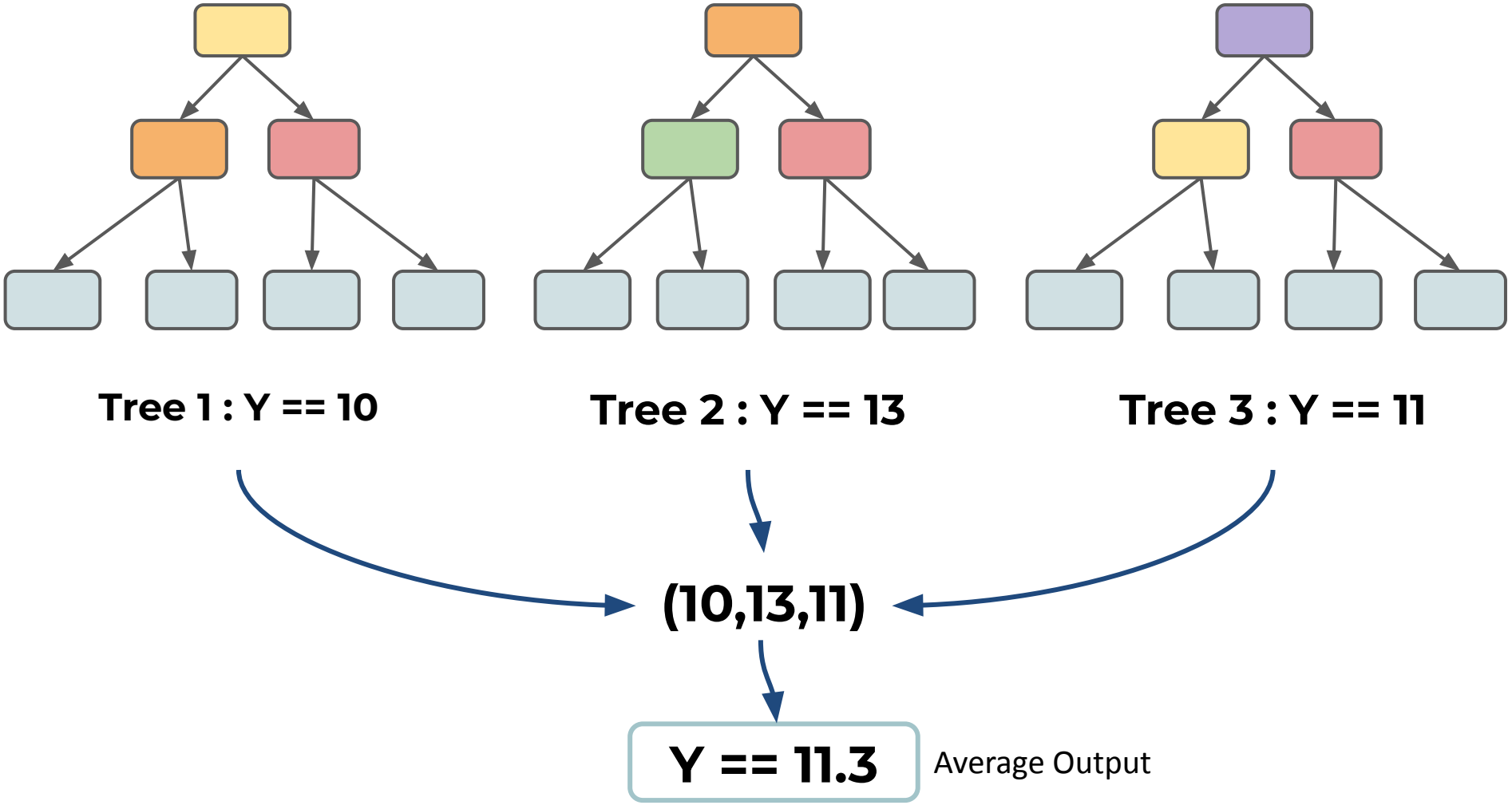


Majority Vote

- How to get prediction out of many models?



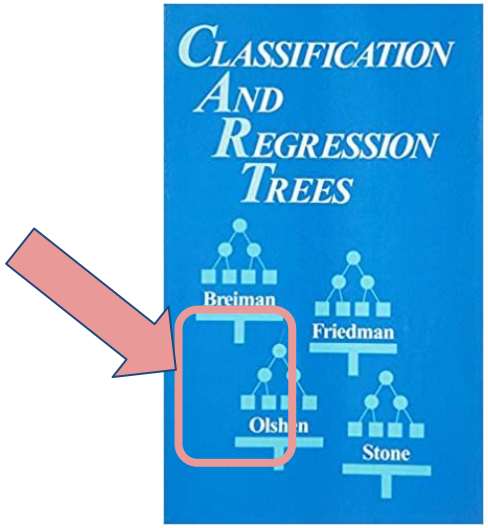
Regression Task (Continuous Labels)



Random Forests

- Ho and other researchers showed that adding the random feature subspace constraint could allow trees to grow much deeper without causing overfitting.
- Since a Random Forest is an ensemble of many decision trees, many of the hyperparameters between both models are shared.

- 2001: Leo Breiman publishes Random Forests in the journal Machine Learning.
- Introduces bootstrapping samples and out of bag error.



Random Forests

- Random Forests expand on decision trees by creating an ensemble of estimators (multiple decision trees).
- We will look directly at Scikit-Learn's Random Forest class call and see what additional hyperparameters we have compared to a single decision tree.
- Since a Random Forest is an ensemble of many decision trees, many of the hyperparameters between both models are shared.
- What are the important hyperparameters unique to a Random Forest?

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0) ¶                    [source]
```

```
class sklearn.ensemble.RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None) ¶                                                    [source]
```

Random Forests

- Random Forest Hyperparameters:
  - **Number of Estimators**
    - *How many decision trees to use total in forest?*
  - **Number of Features**
    - *How many features to include in each subset?*
  - **Bootstrap Samples**
    - *Allow for bootstrap sampling of each training subset of features?*
  - **Out-of-Bag Error**
    - *Calculate OOB error during training?*

Random Forest Classifier

Hyperparameters, Number of Estimators and Features

Random Forests

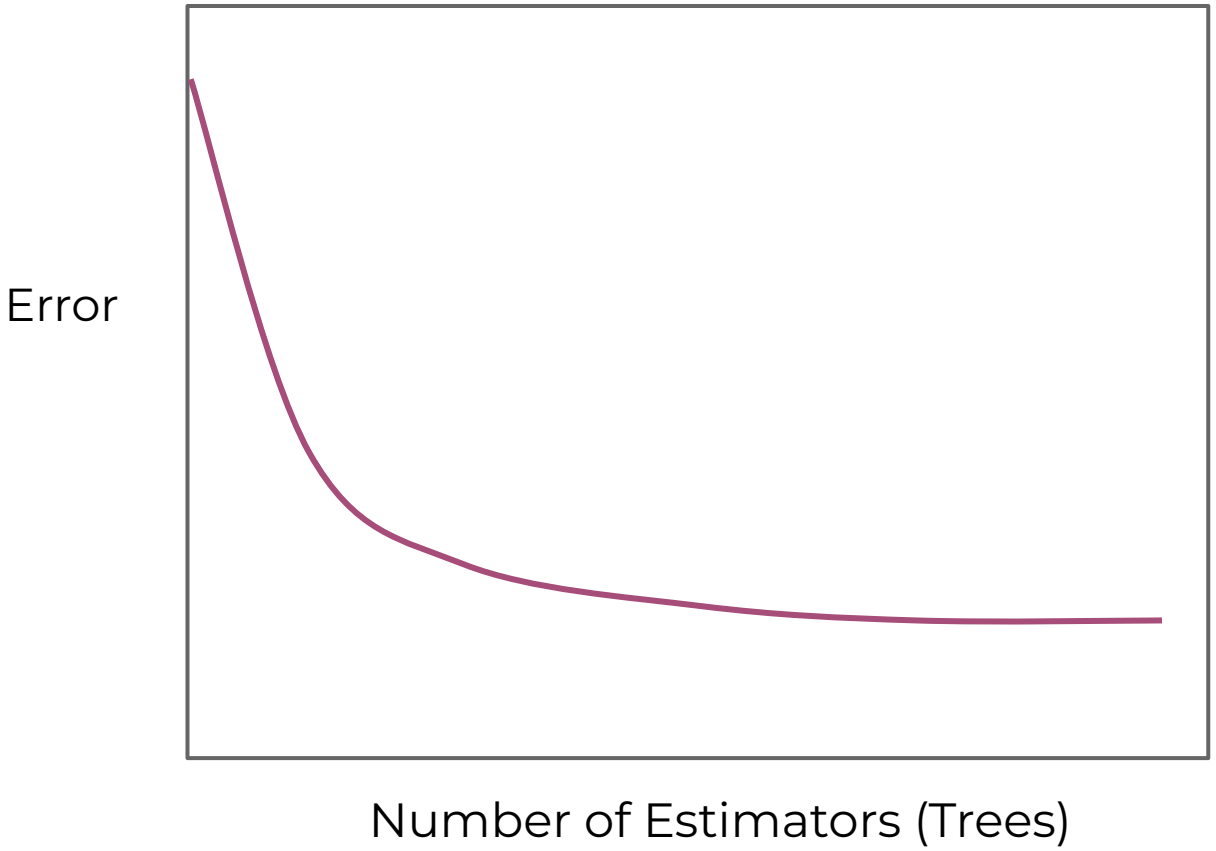
- Number of Estimators
  - Intuitively, we know the more decision trees, the more opportunities to learn from a variety of feature subset combinations.
  - Is there a limit to adding more trees?
  - Is there a danger of overfitting?
- From Leo Breiman's official page on Random Forests:
  - "*Random forests does not overfit. You can run as many trees as you want. It is fast.*"

www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks

Random Forests

- How to choose number of trees?
  - Reasonable Default Value: 100
  - Publications suggest 64-128 trees.
  - Cross Validate a grid search of trees.
  - Plot Error versus number of trees (similar to elbow method of KNN).
    - Should notice diminishing error reduction after some N trees.

- Error vs. Trees



- After a certain number of trees, two things that can occur:
  - Different random selections don't reveal any more information.
    - Trees become highly correlated.
  - Different random selections are simply duplicating trees that have already been created.
- This allows us to be quite lenient in setting number of estimators hyperparameters, as overfitting is of minimal concern.
- Now let's discuss how to choose the number of features to randomly select at each split.

- Random Forest Hyperparameters:
  - Number of Features
    - *How many features to include in each subset when splitting at a node?*
- Number of Features in Subset?
  - Original Publication suggested subset of $\log_2(N+1)$ random features in subset given a set of N total features.
  - From Leo Breiman's official page on Random Forests:
  - "An interesting difference between regression and classification is that the correlation increases quite slowly as the number of features used increases."

www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks

- **Number of Features in Subset?**
  - Current suggested convention is **sqrt(N)** in the subset given **N** features.
  - Later suggestions by Breiman indicated **N/3** may be more suitable for regression tasks, typically larger than **sqrt(N)**.
  - ISLR indicates this should be treated as a tuning parameter, with **sqrt(N)** as a good starting point.
  - All the previous options shown were explored using empirical methods, so it is likely you will need to adjust based on your specific dataset.

- Hyperparameter Review:
  - Number of Estimators:
    - Start with 100 as default, feel free to grid search for higher values.
  - Number of Features for Selection:
    - Start with sqrt(N), grid search for other possible values (N/3).

Random Forest Classifier

Hyperparameters, Bootstrap Samples and OOB Error

- **Bootstrap Samples**
  - *Allow for bootstrap sampling of each training subset of features?*
- First, let's understand "bootstrapping" in general terms…
- **What is Bootstrapping?**
  - A term used to describe "random sampling with replacement".

- What is Bootstrapping?
- Let’s see a quick example given a set of letters…

A

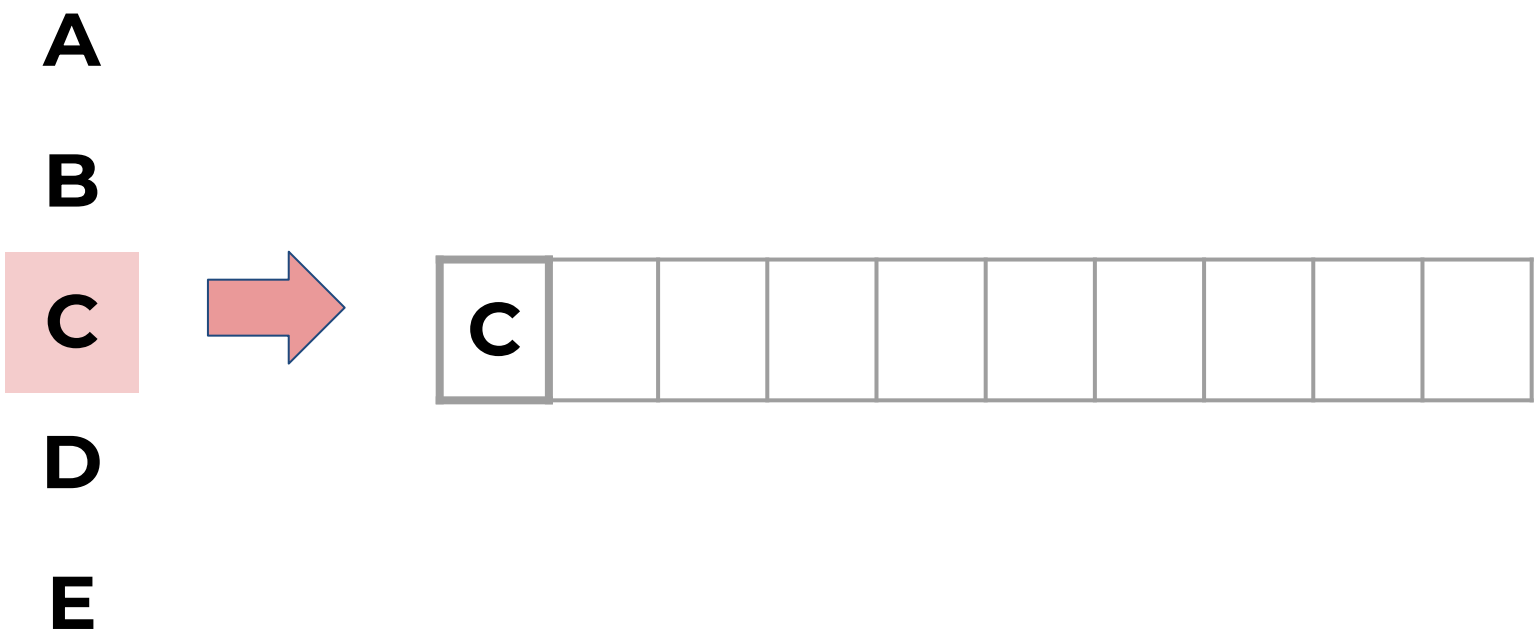
B

C

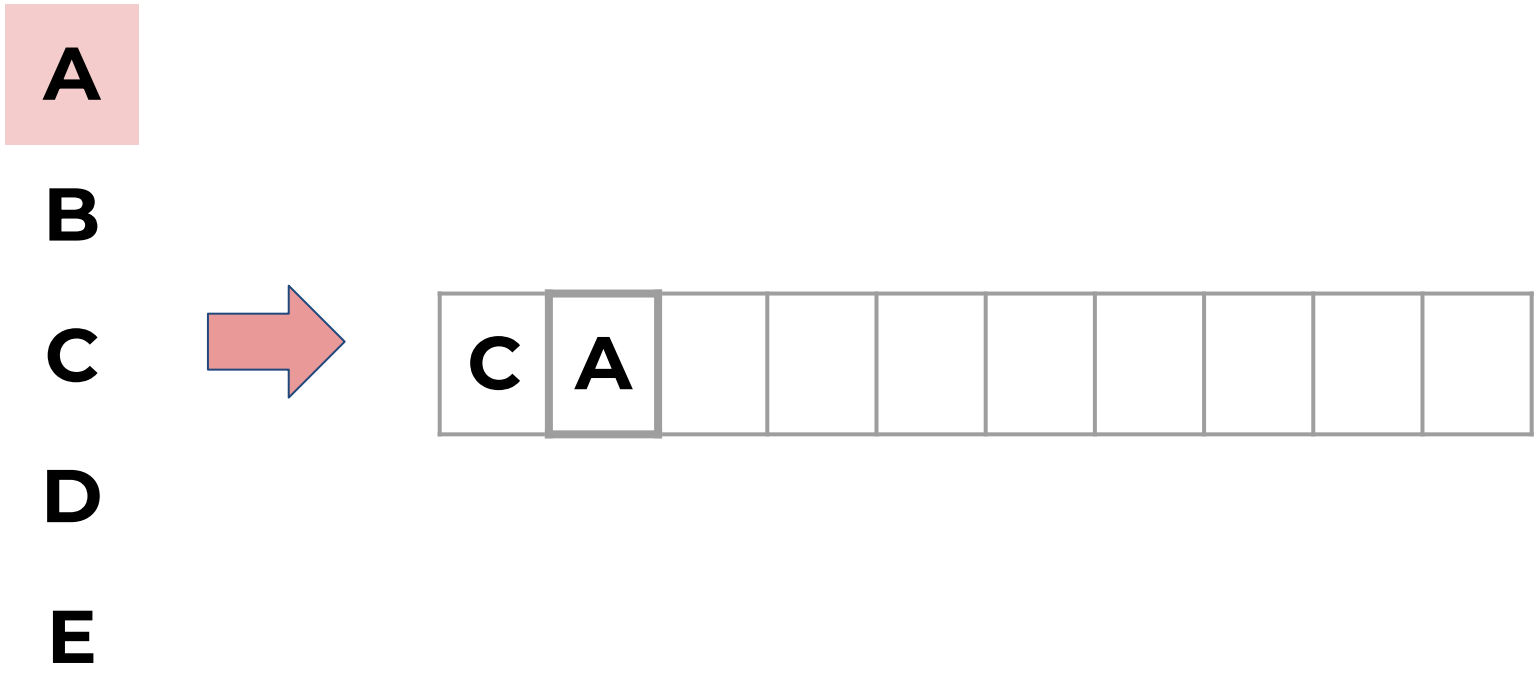
D

E

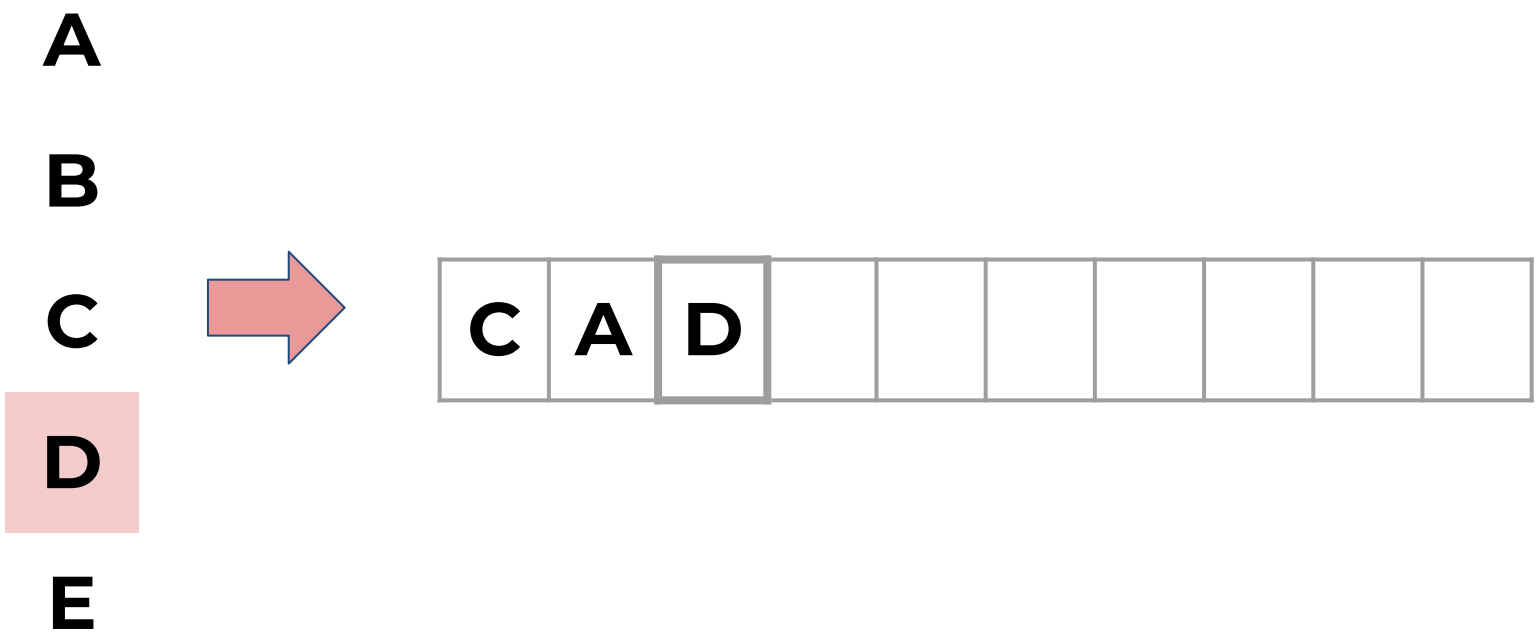
- What is Bootstrapping?



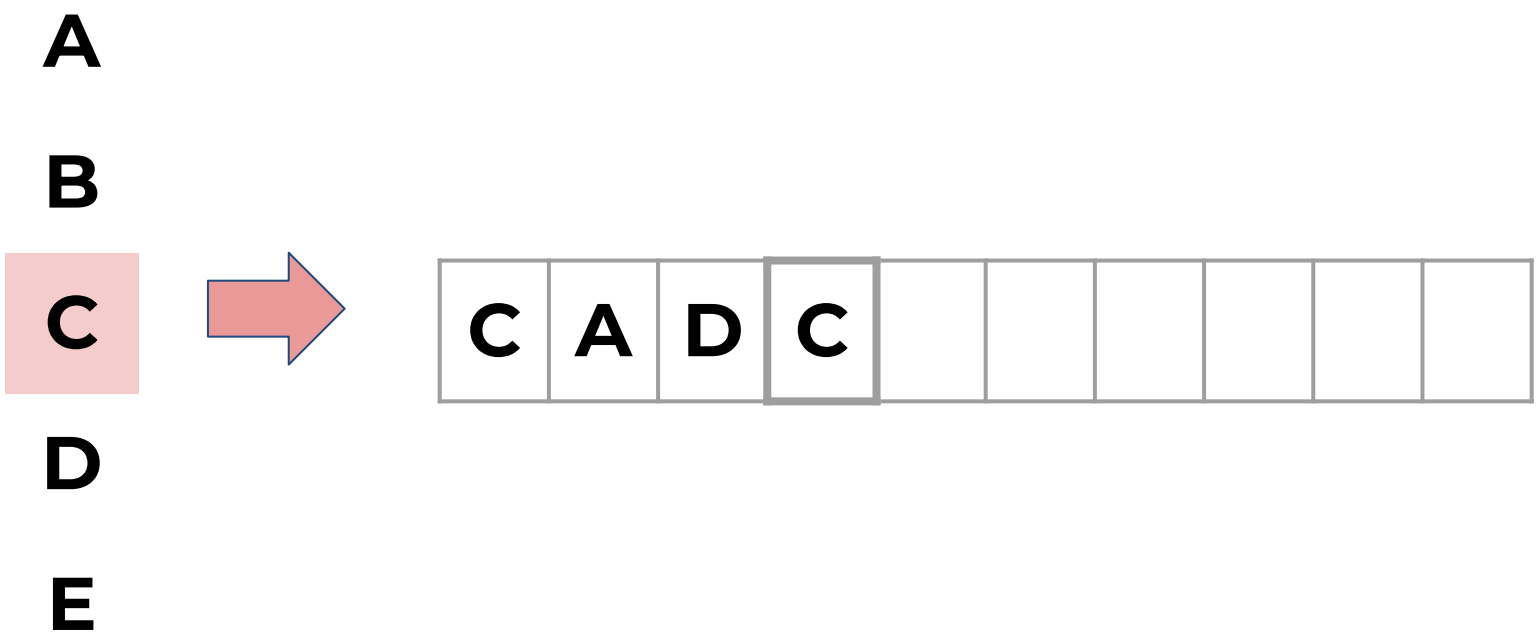
- What is Bootstrapping?



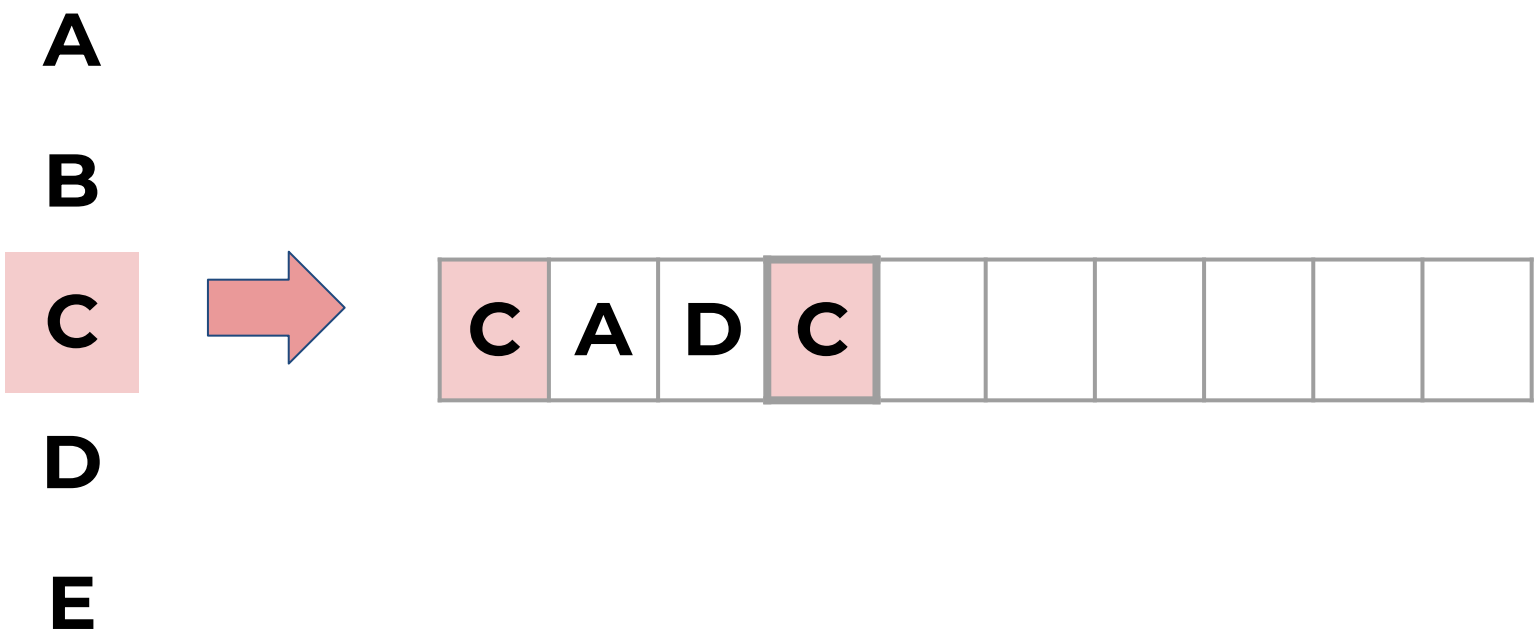
- What is Bootstrapping?



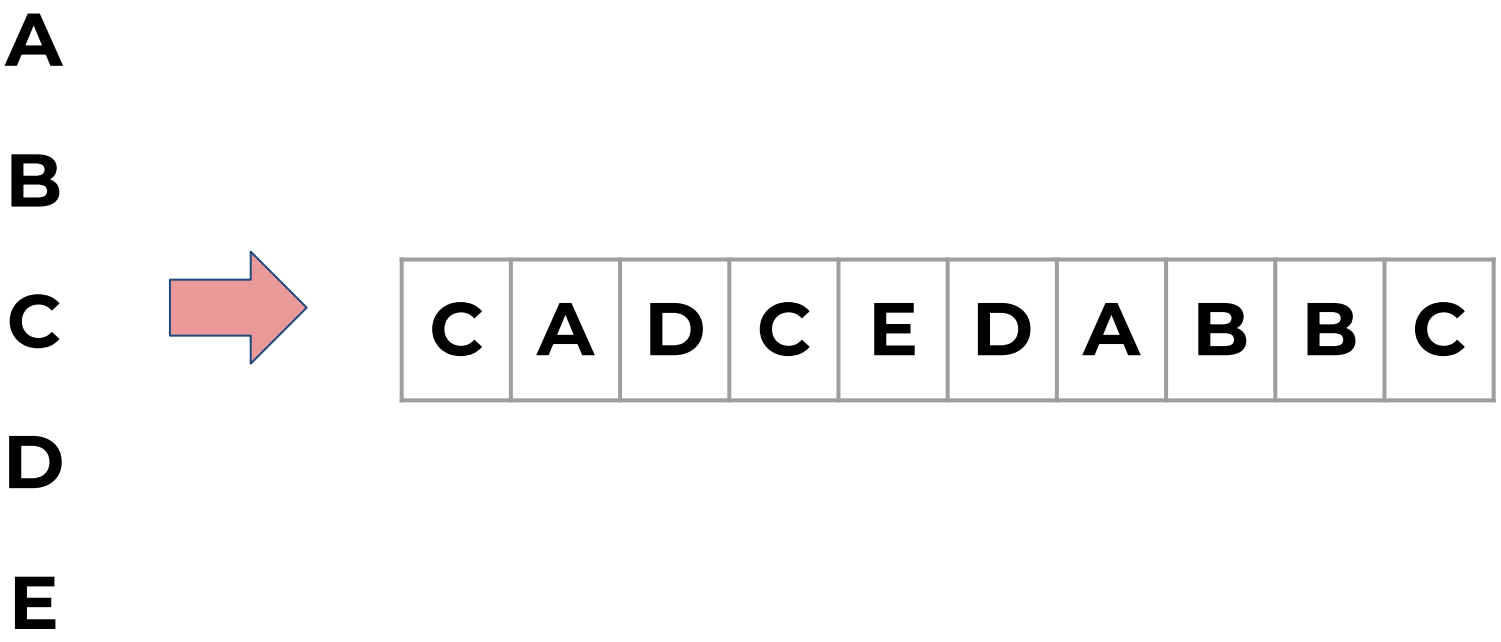
- What is Bootstrapping?



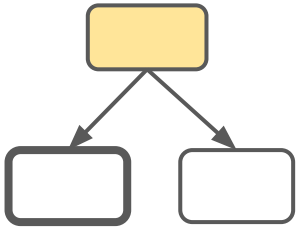
- What is Bootstrapping?

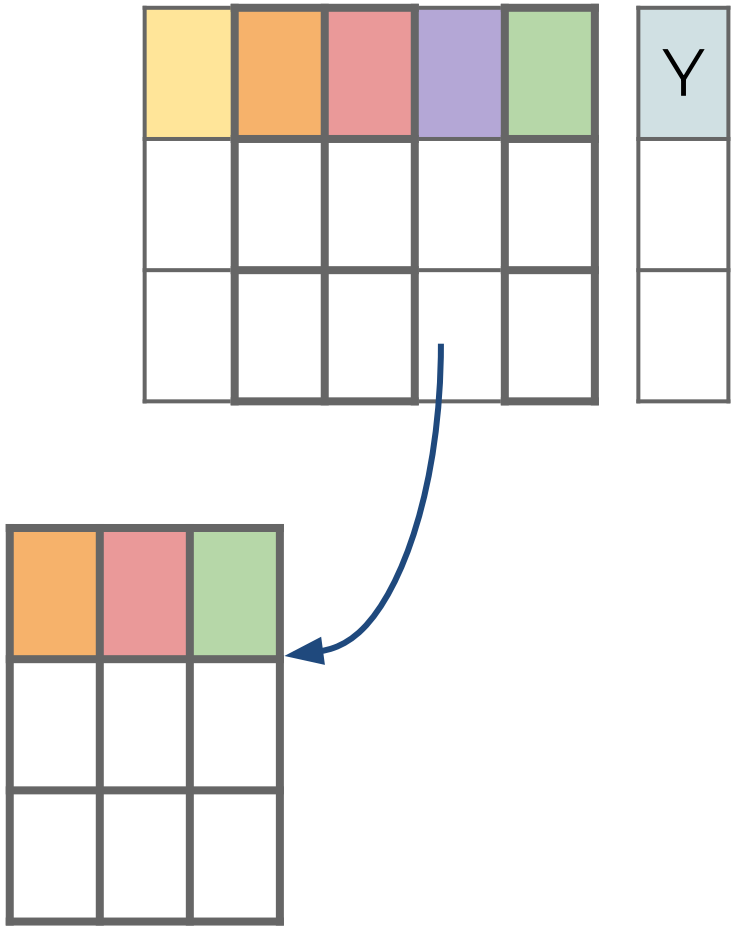


- What is Bootstrapping?

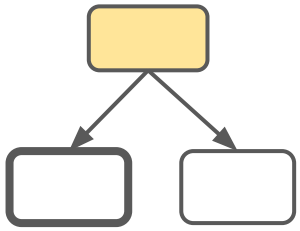


- Bootstrapping in Random Forest
  - Recall for each split we are randomly selecting a subset of features.
  - This random subset of features helps create more diverse trees that are not correlated to each other.

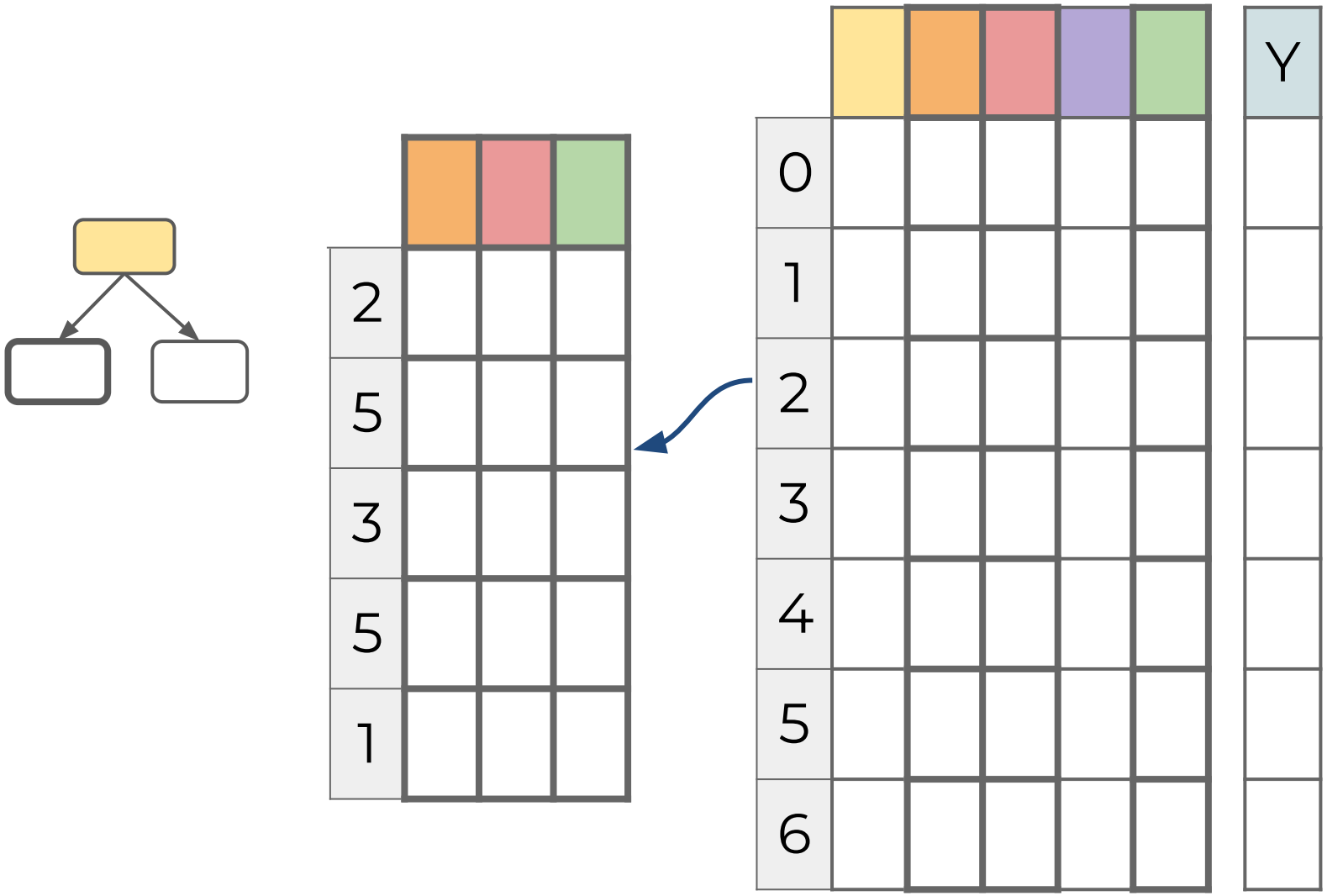



Y

- Bootstrapping in Random Forest
  - To further differentiate trees, we could bootstrap a selection of rows for each split.
  - This results in two randomized training components:
    - Subset of Features Used
    - Bootstrapped rows of data



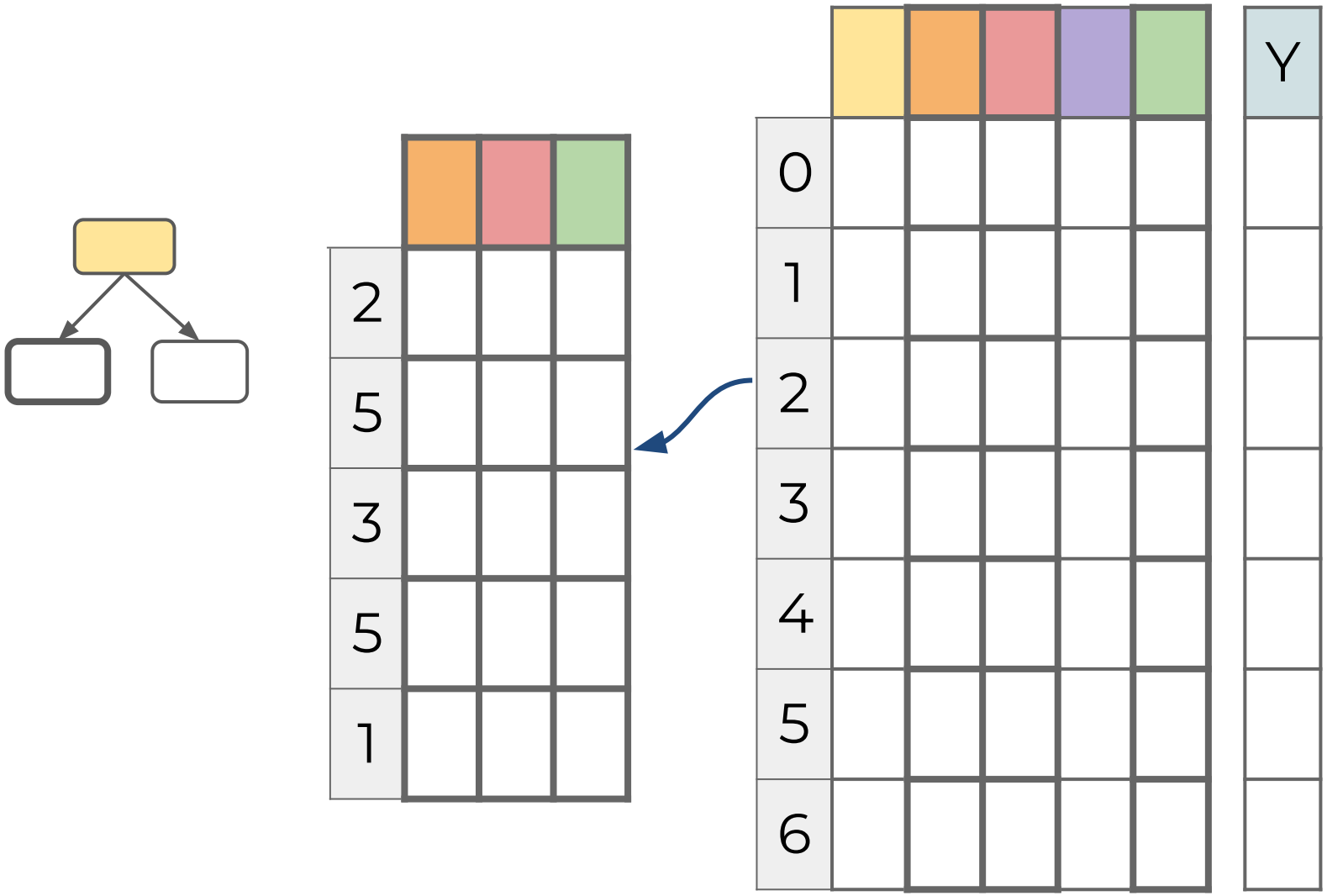
						Y
0						
1						
2						
3						
4						
5						
6						

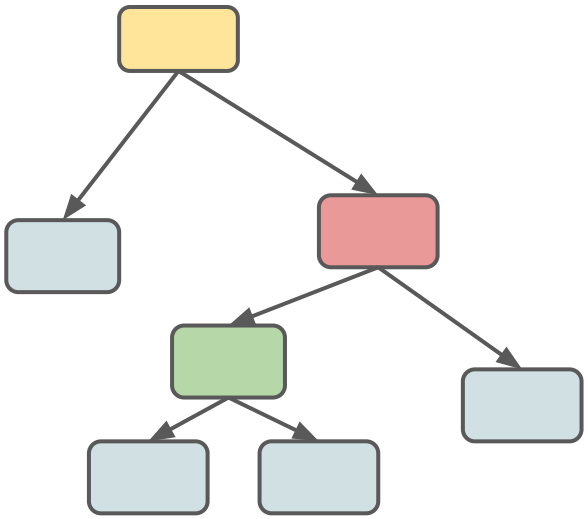

Y
0
1
2
3
4
5
6
2
5
3
5
1

- Bootstrapping can be set to False during training (it is True by default).
- Bootstrapping is yet another hyperparameter meant to reduce correlation between trees, since trees are then trained on different subsets of feature columns and data rows!

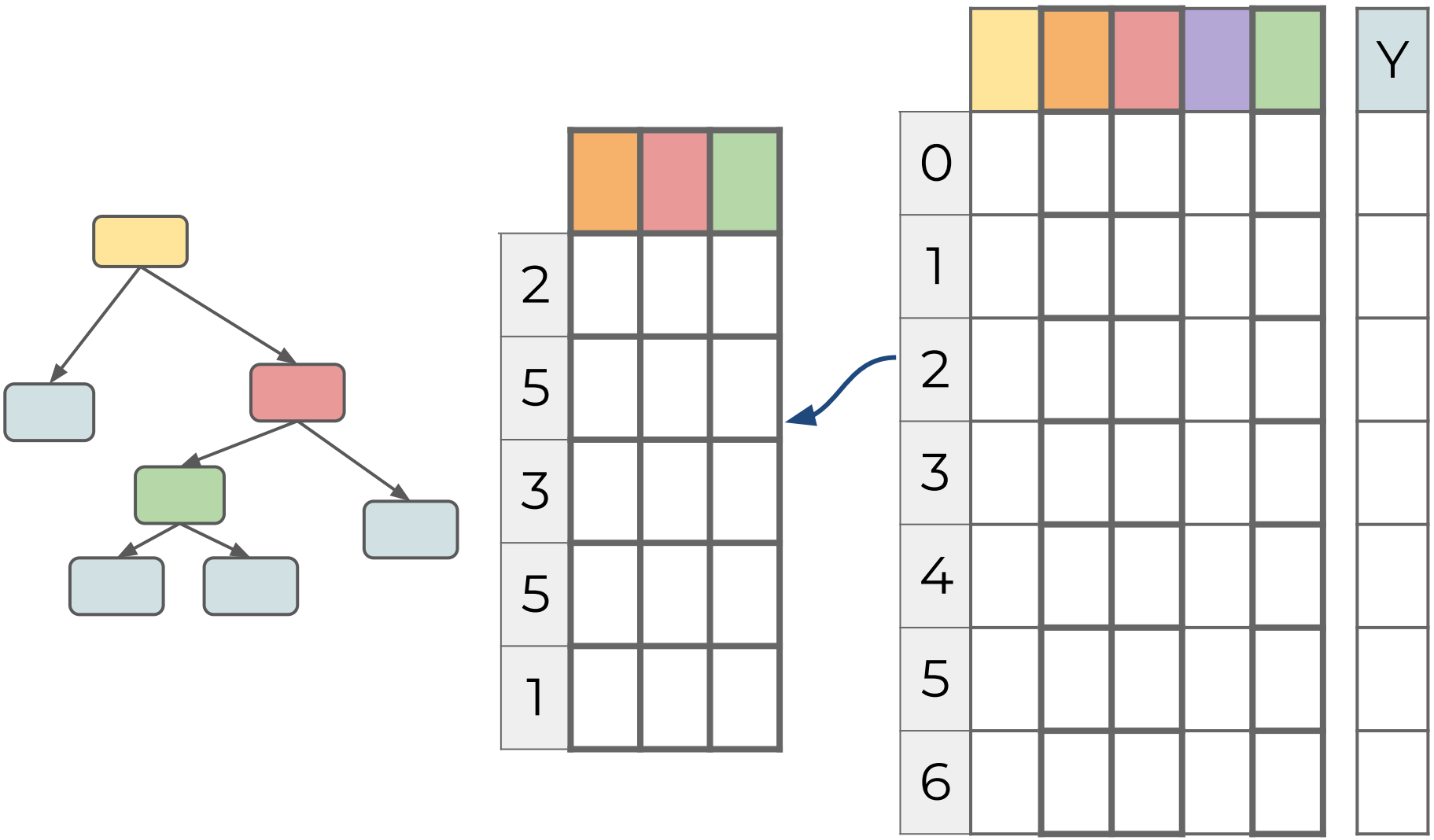
- Random Forest Hyperparameters:
  - Out-of-Bag Error
    - *Calculate OOB error during training?*

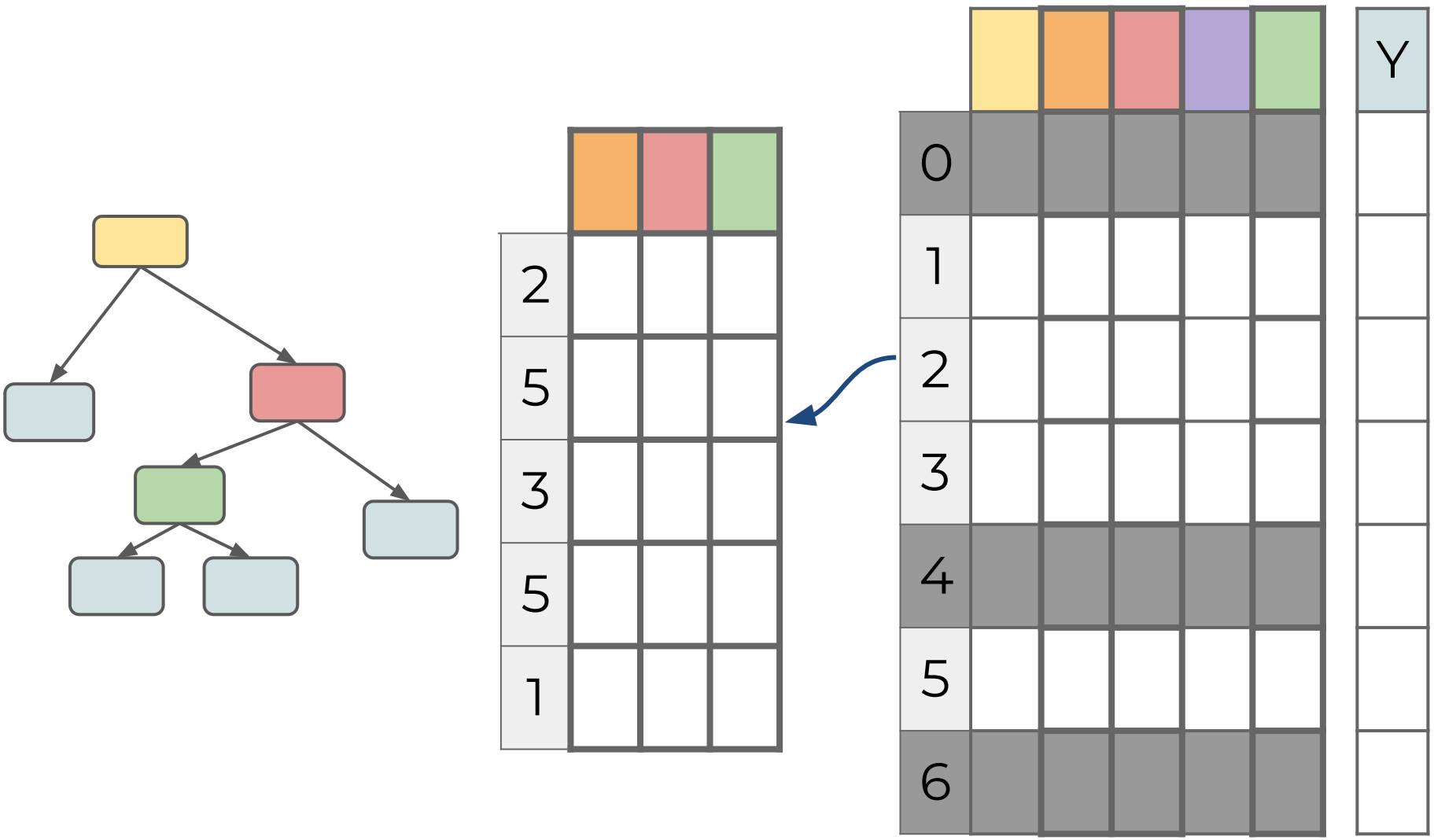
- **What is Bagging?**
  - Recall to actually use a Random Forest, we use **b**ootstrapped data and then calculate a prediction based on the **ag**gregated prediction of the trees:
    - Classification: Most Voted Y Class
    - Regression: Average Predicted Ys
- If we performed bootstrapping when building out trees, this means that for certain trees, certain rows of data were not used for training.

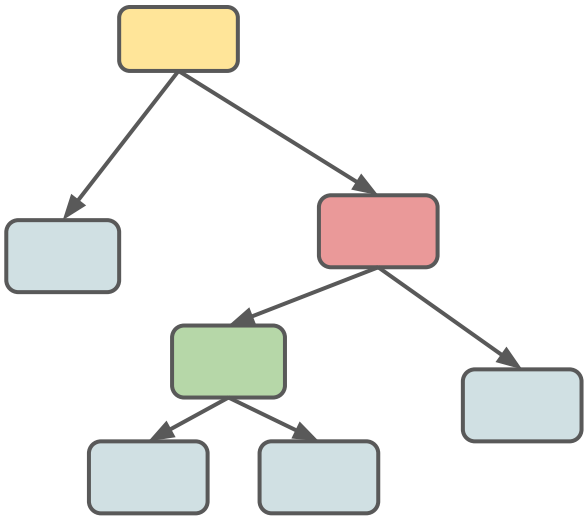

Y
0
1
2
3
4
5
6
2
5
3
5
1



						Y
0						
1						
2						
3						
4						
5						
6						

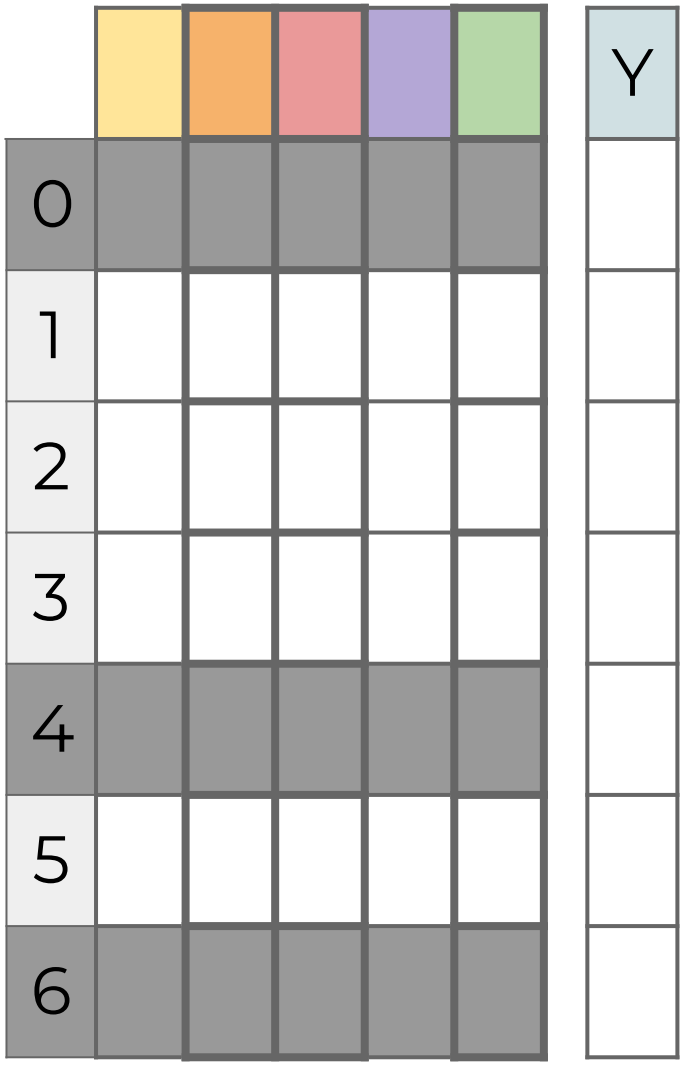

Y
0
1
2
3
4
5
6
2
5
3
5
1

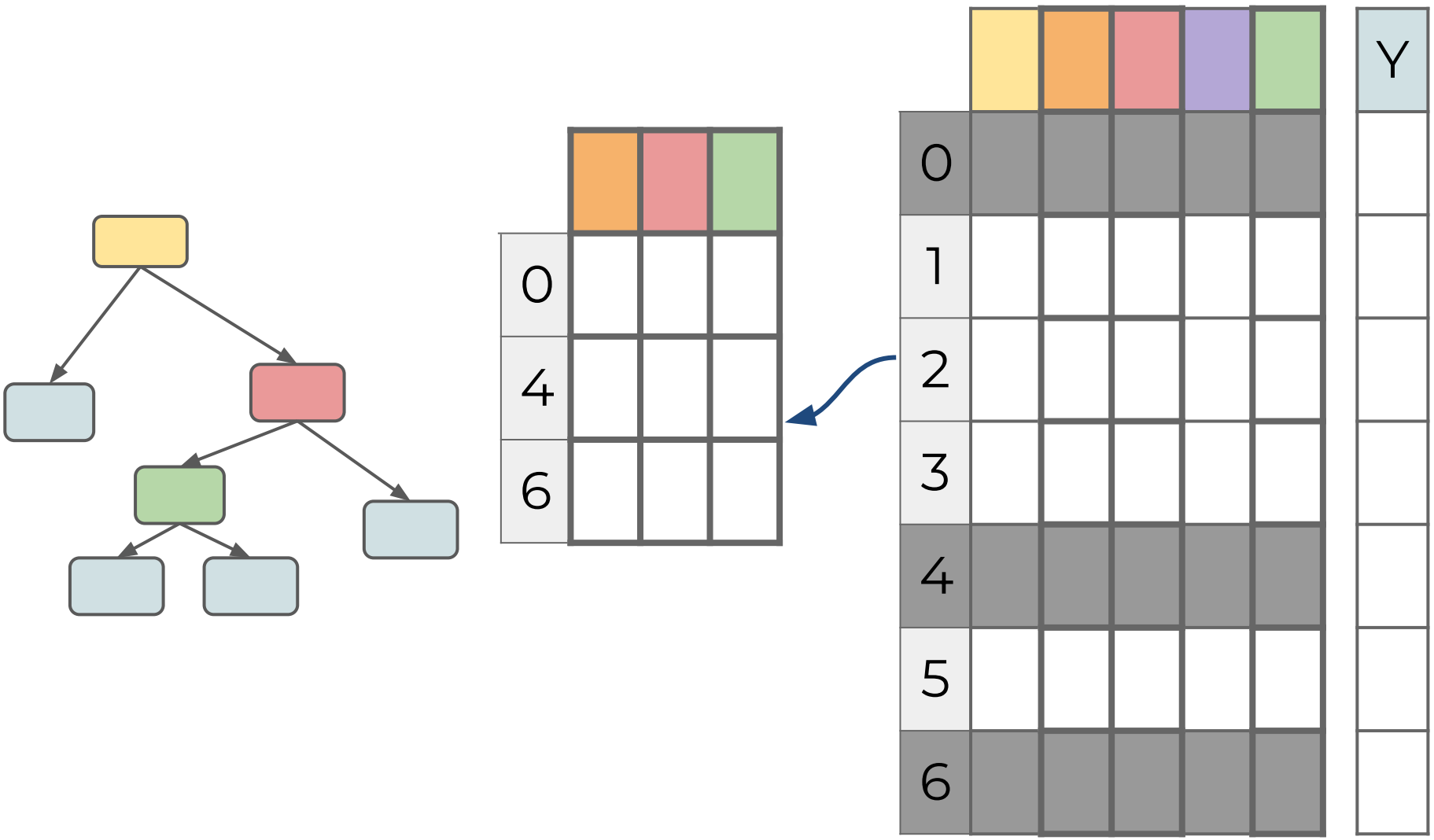

Y
0
1
2
3
4
5
6
2
5
3
5
1

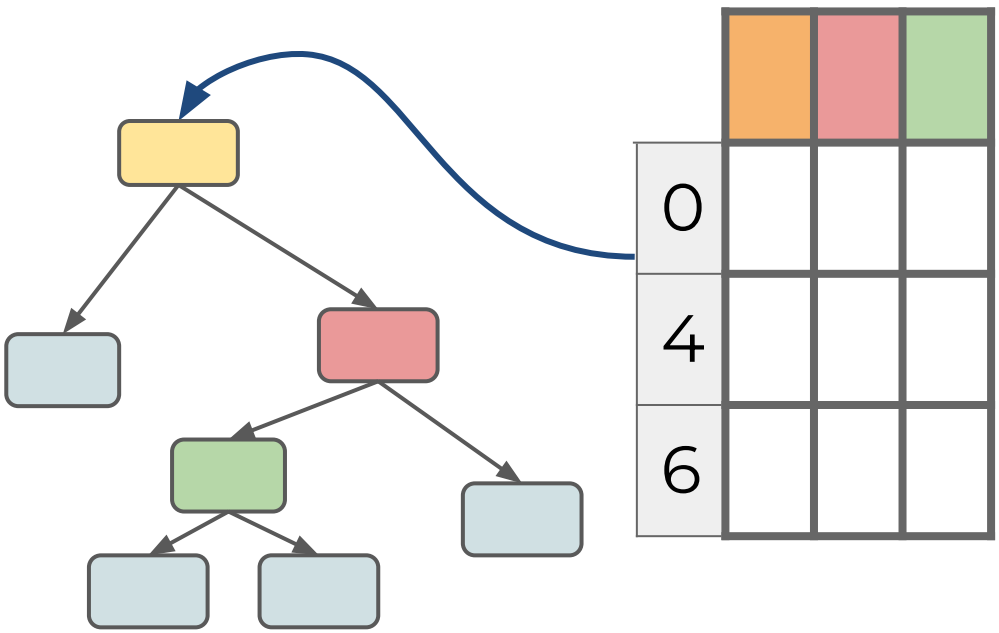


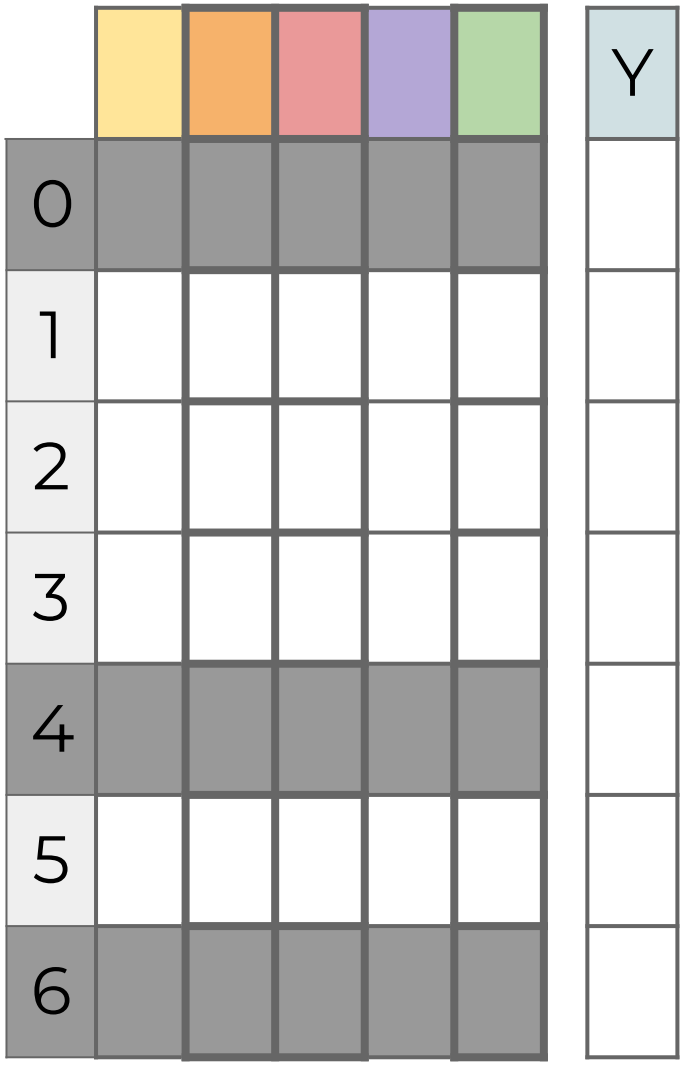
Out-of-Bag Samples

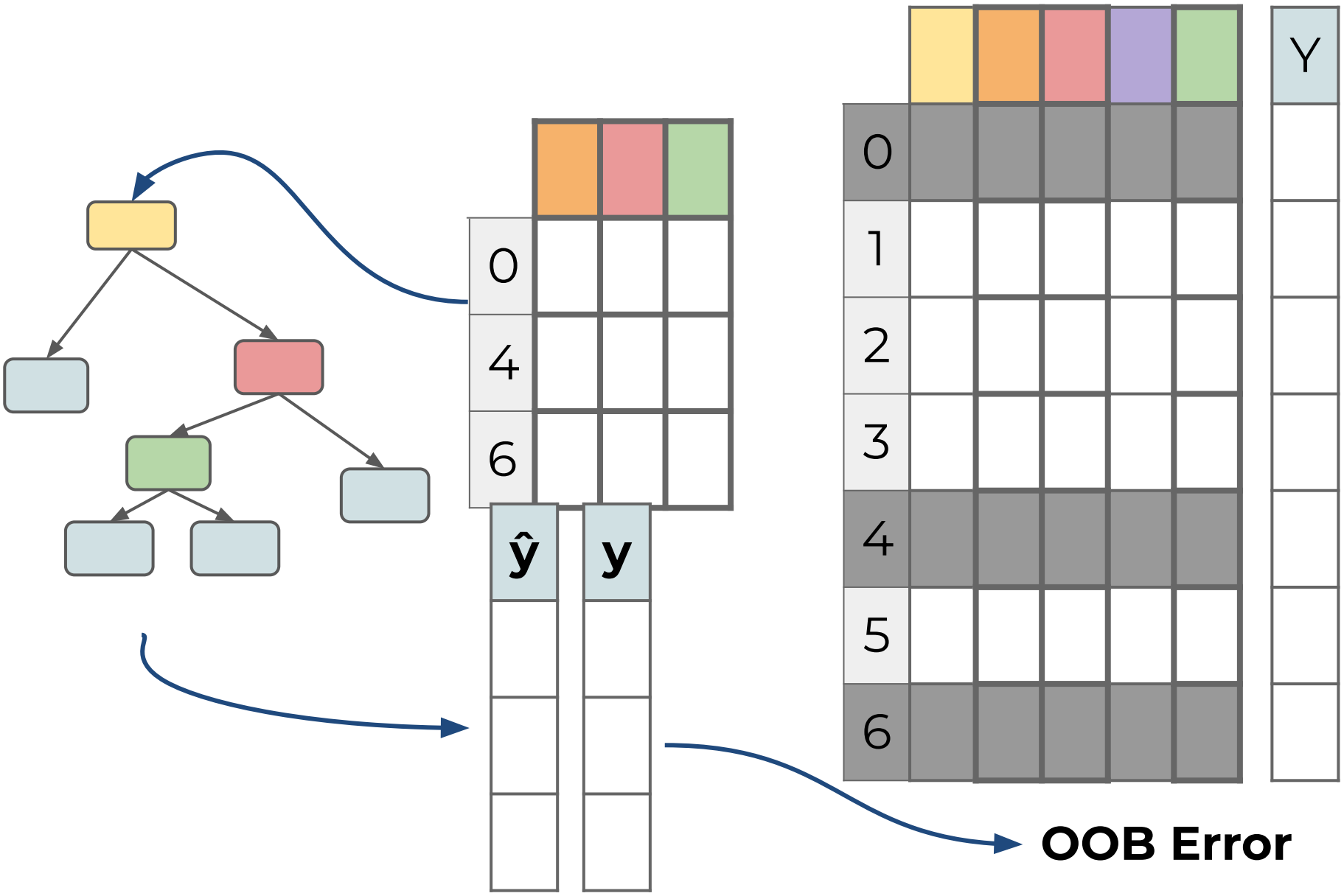
- Not used for constructing some trees.
- We could use these to get performance test metrics on trees that did not use these rows!




0
4
6
0
1
2
3
4
5
6
Y


0
4
6


Y
0
1
2
3
4
5
6


0
4
6
ŷ
y
0
1
2
3
4
5
6
Y
OOB Error

- Note that OOB Score is a hyperparameter that doesn't really affect training process.
- It is separate from bootstrapping, OOB Score is an optional way of measuring performance, an alternative to using a standard train/test split, since bootstrapping naturally results in unused data during training.
- Note that OOB Score is also limited to not using all the trees in the random forest, it can only be calculated on trees that did not use the OOB data!
- Due to not using the entire random forest, the default value of OOB Score hyperparameter is set to False.