Random Forests

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).

Random Forests

- Section Overview:
  - Motivation and History
  - Hyperparameters
  - Random Forest Classification
  - Random Forest Regression
    - Comparing many regression models (SVR, DTR, etc...)

Random Forests

Theory and Intuition: Motivation and History

Random Forests

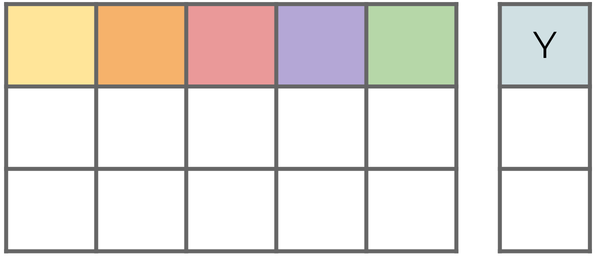
- Why not just continue to use Decision Trees?
- What is the motivation behind Random Forests and how do they improve on Decision Trees?
- Let’s think back to the construction of a single decision tree…

Random Forests

- Imagine a data set with features and label:

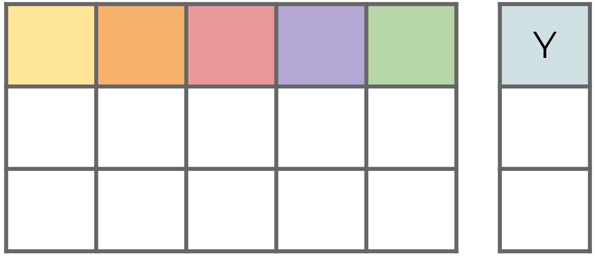
Random Forests

- Imagine a data set with features and label:



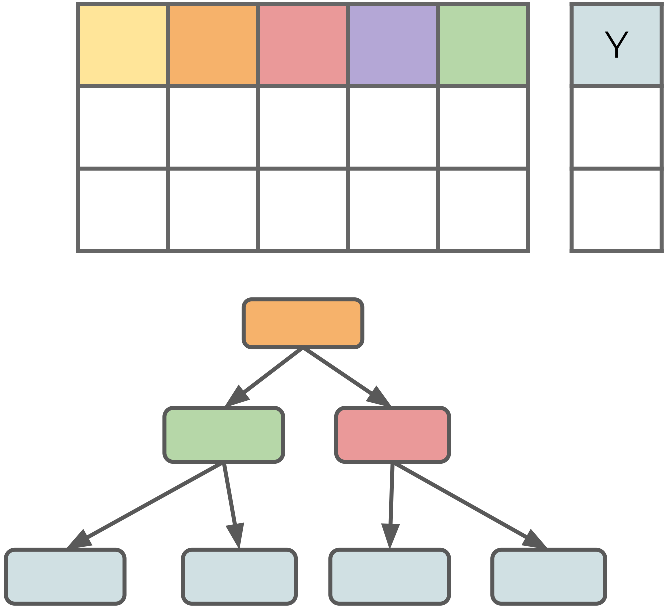
Random Forests

- Decision Tree restricted by gini impurity:



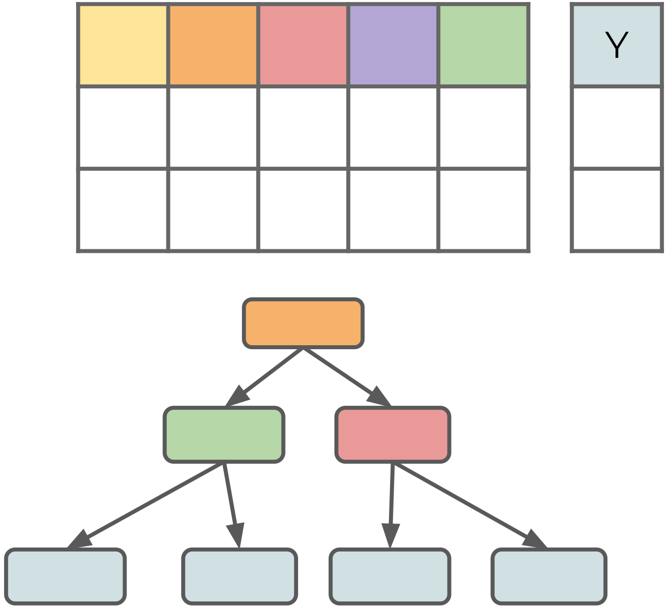
Random Forests

- Decision Tree restricted by gini impurity:



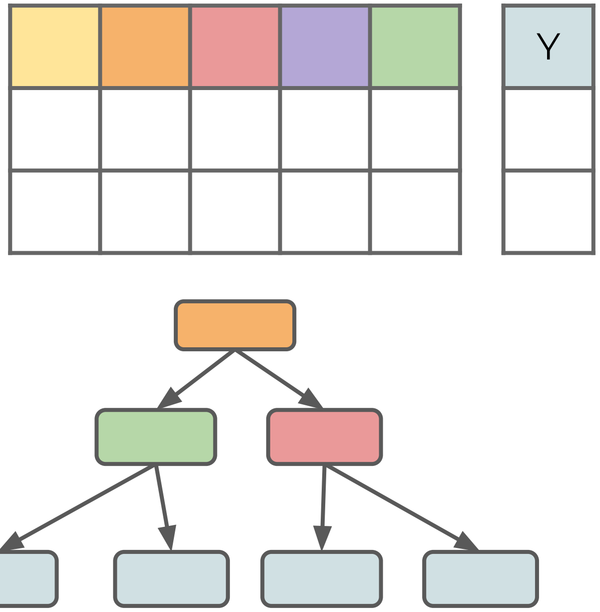
Random Forests

- No guarantee of using all features!



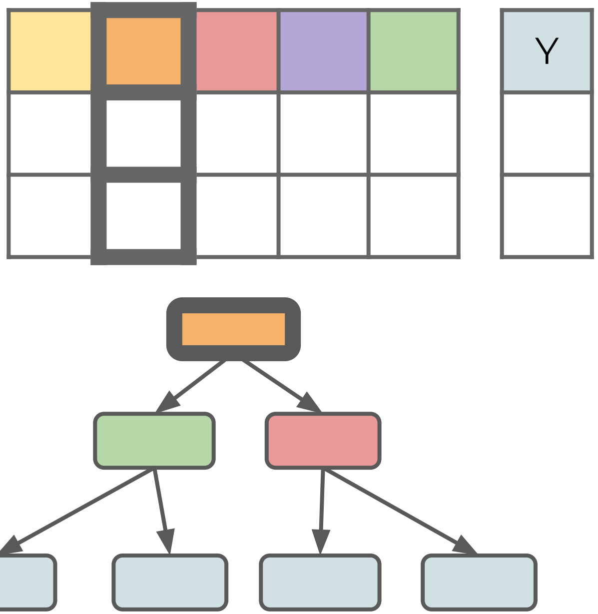
Random Forests

- Root node will always be the same!



Random Forests

- 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

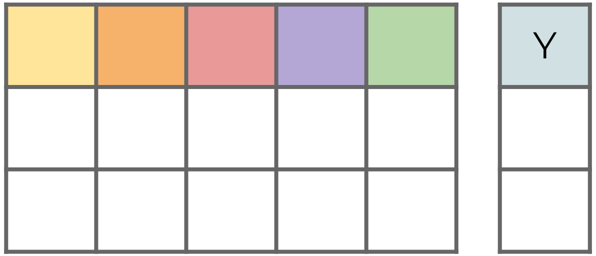
Random Forests

- However 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.

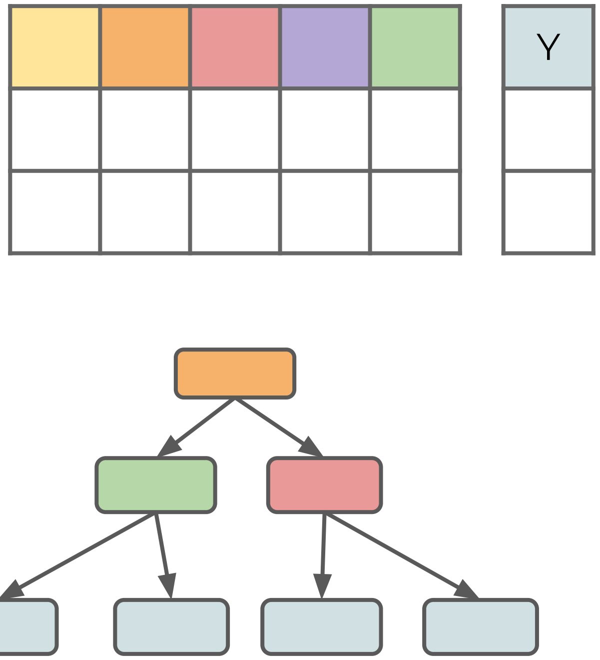
Random Forests

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

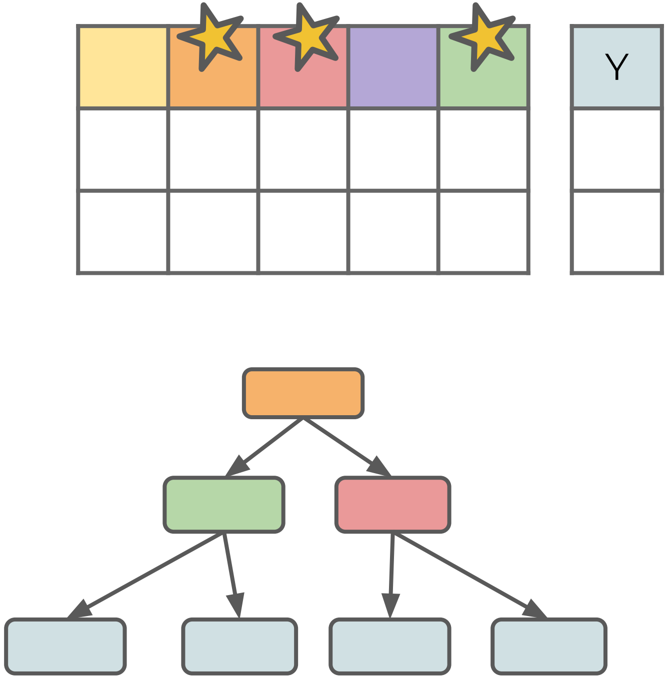
Random Forests



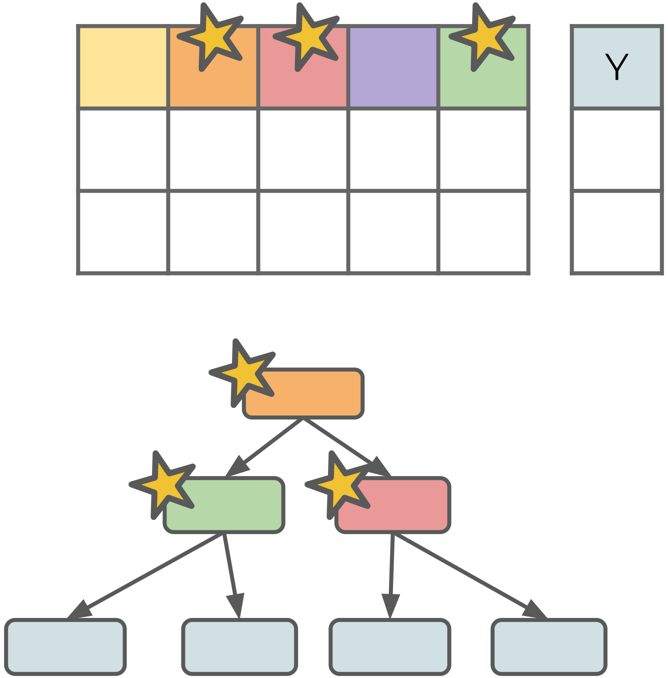
Random Forests



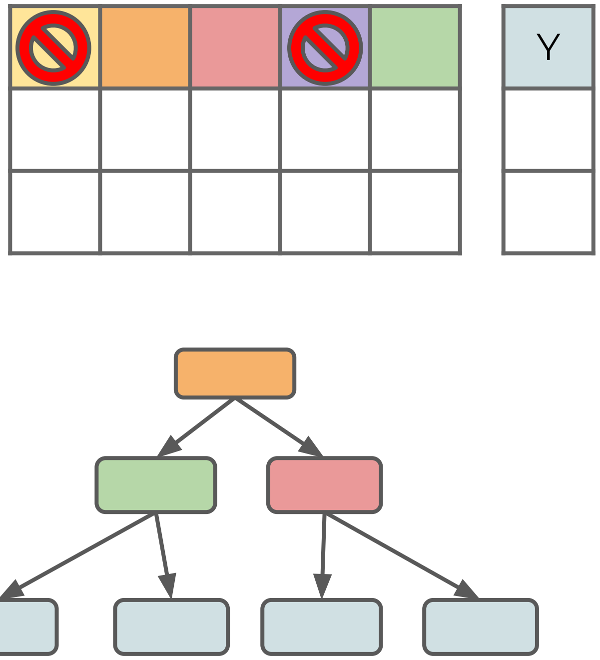
Random Forests



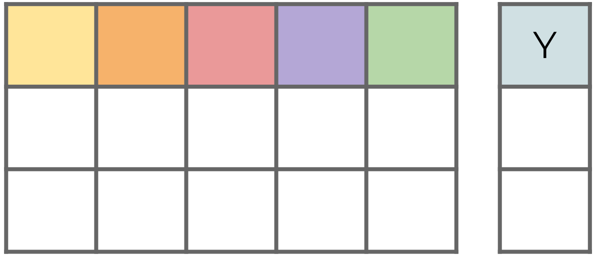
Random Forests



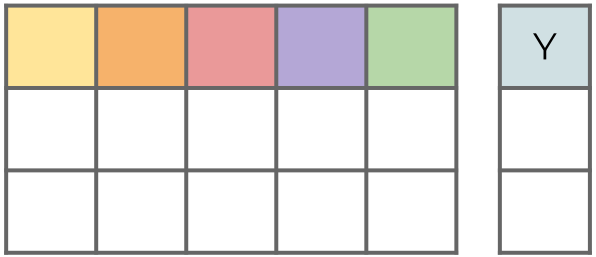
Random Forests



Random Forests

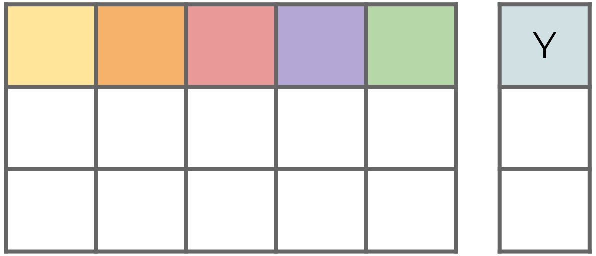


Random Forests

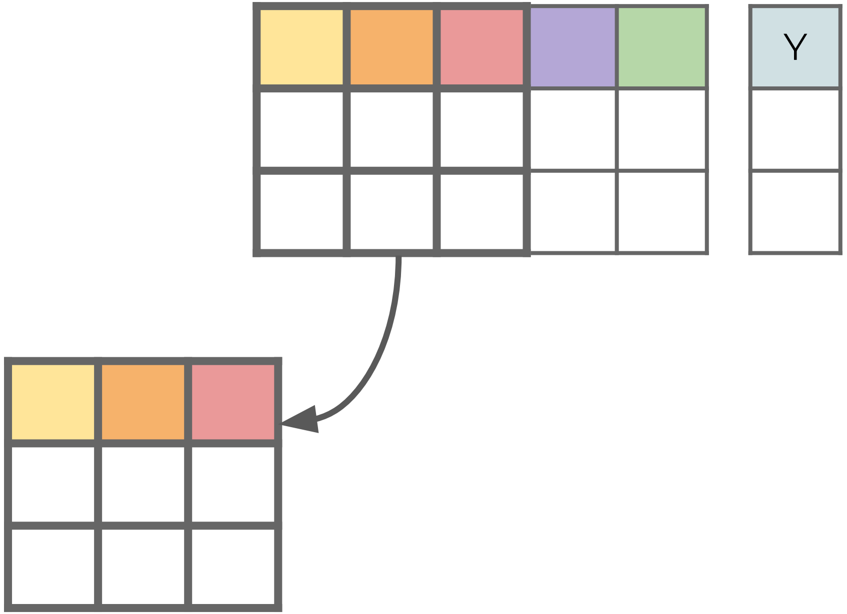


Create subsets of randomly picked features at each potential split

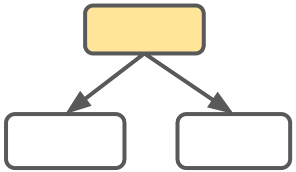
Random Forests

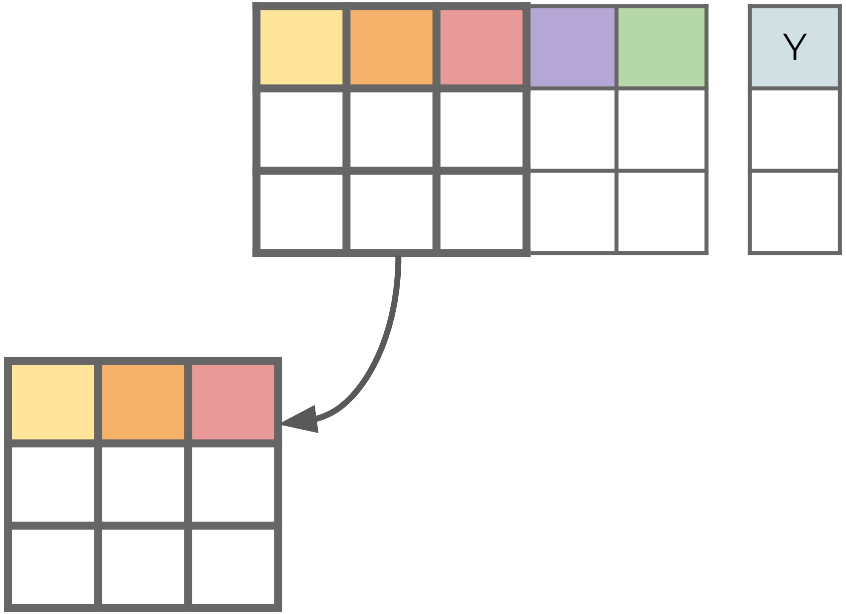


Random Forests

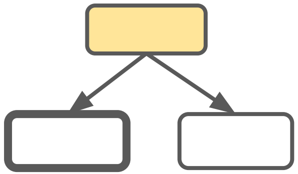


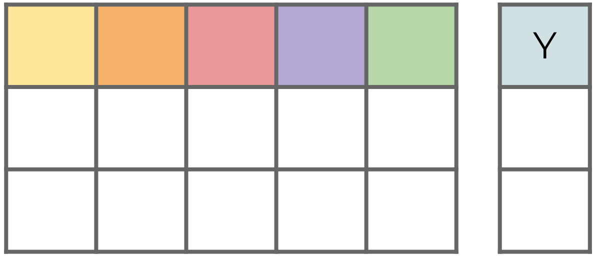
Random Forests



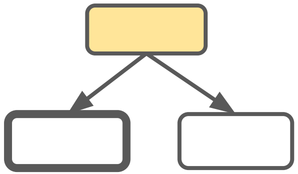


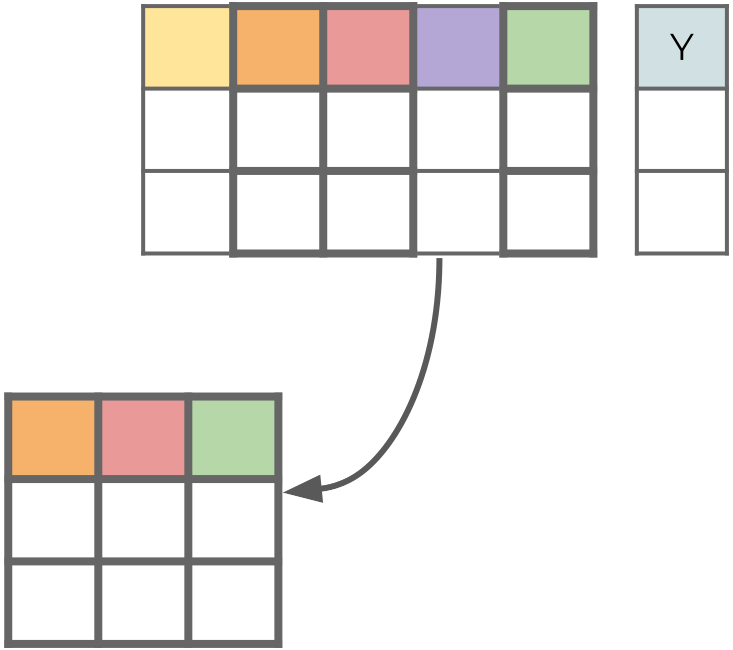
Random Forests



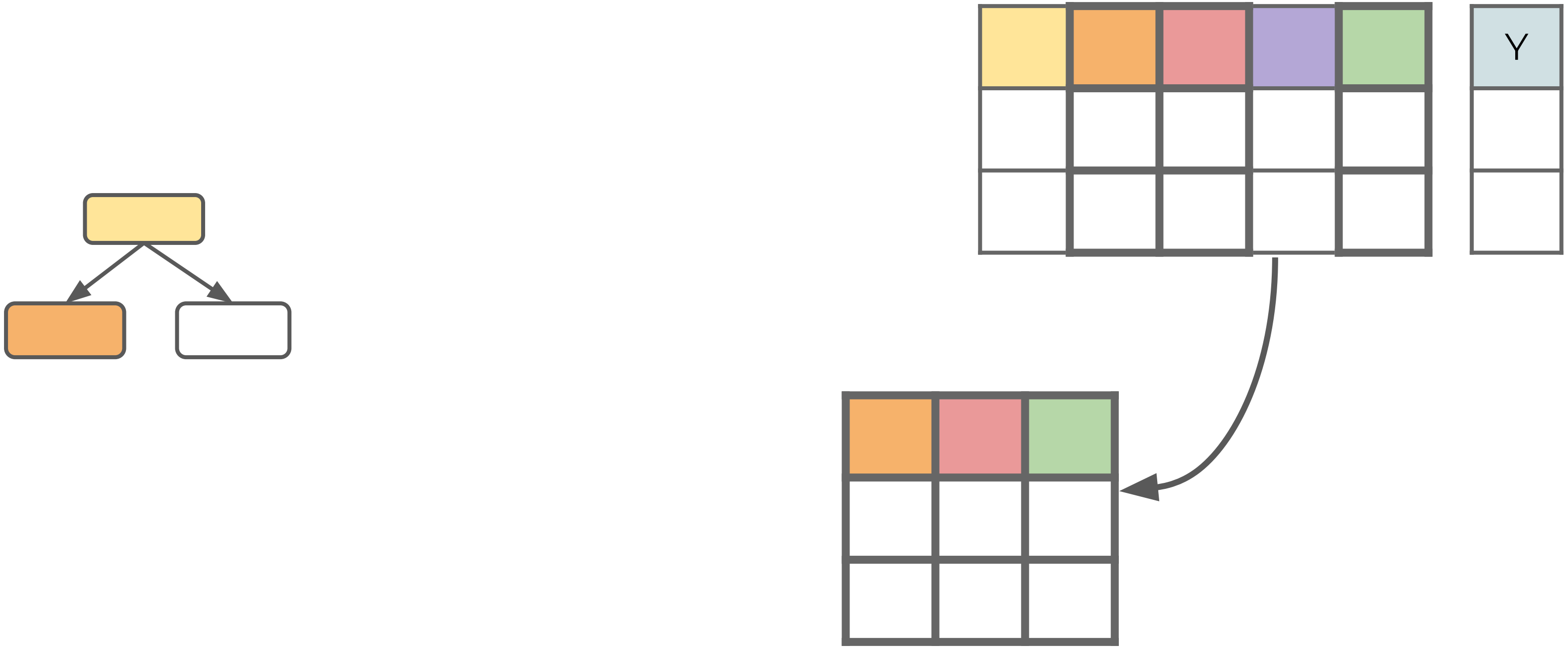


Random Forests

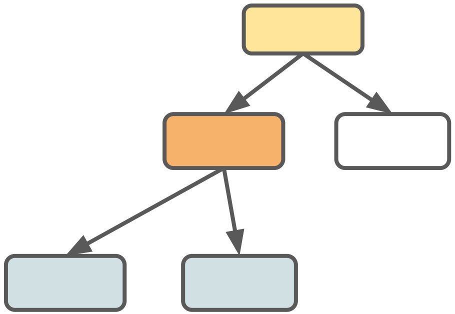


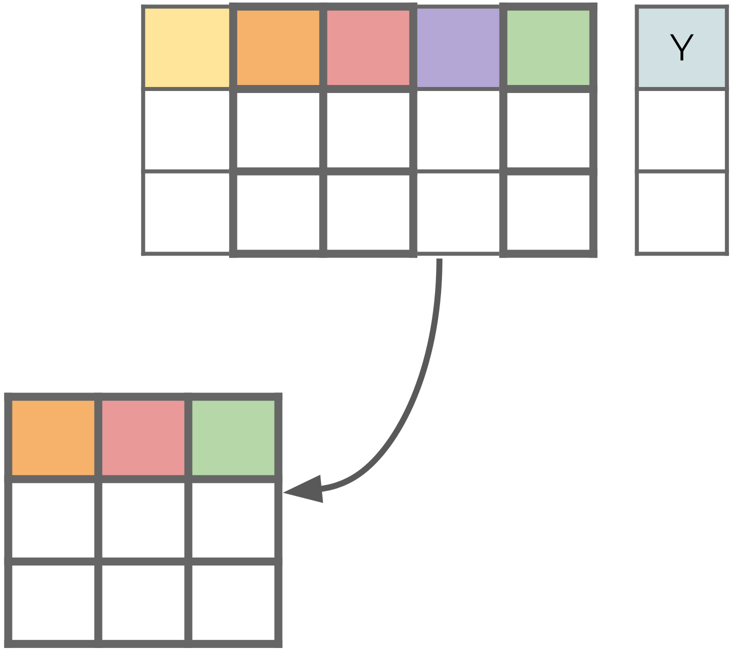


Random Forests

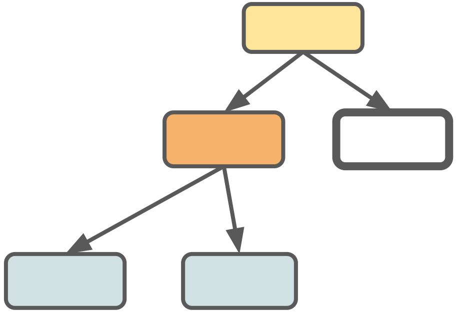


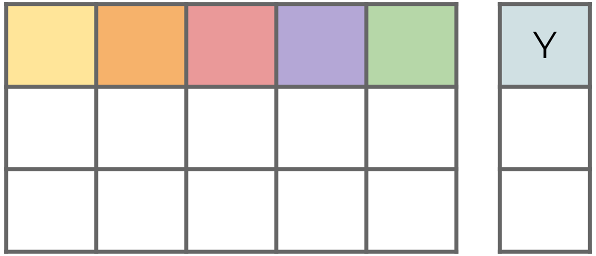
Random Forests



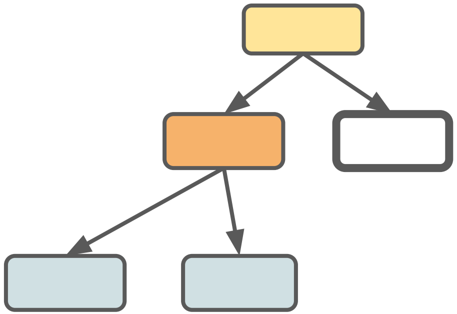


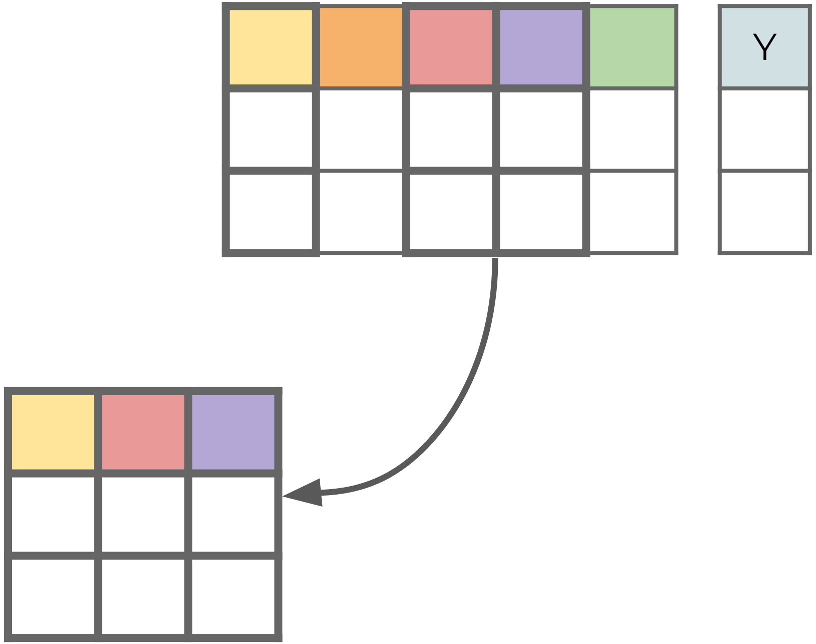
Random Forests



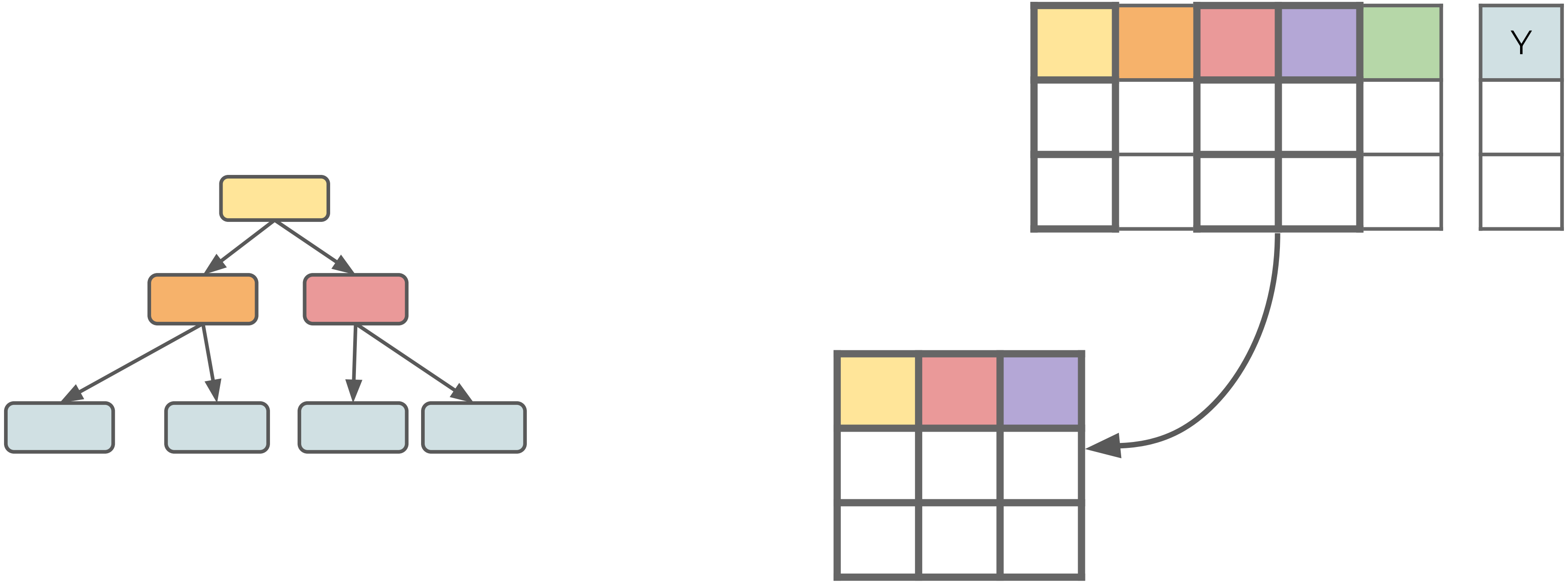


Random Forests

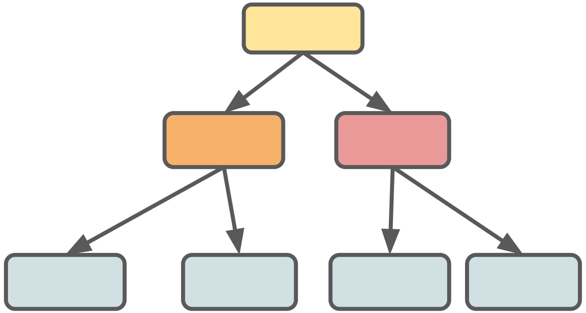


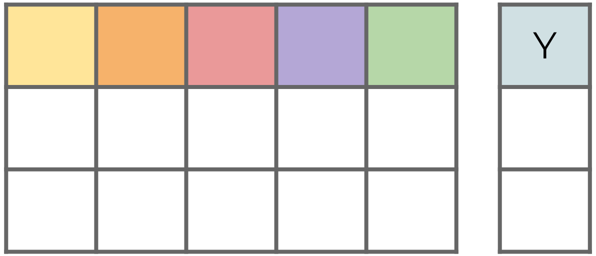


Random Forests

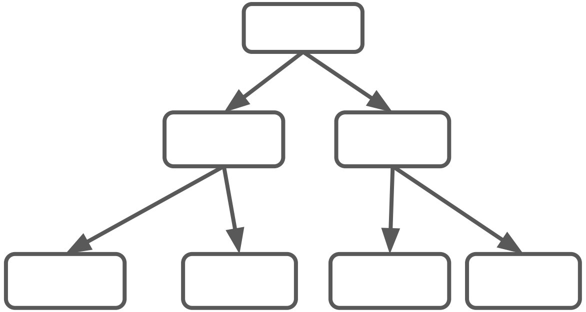


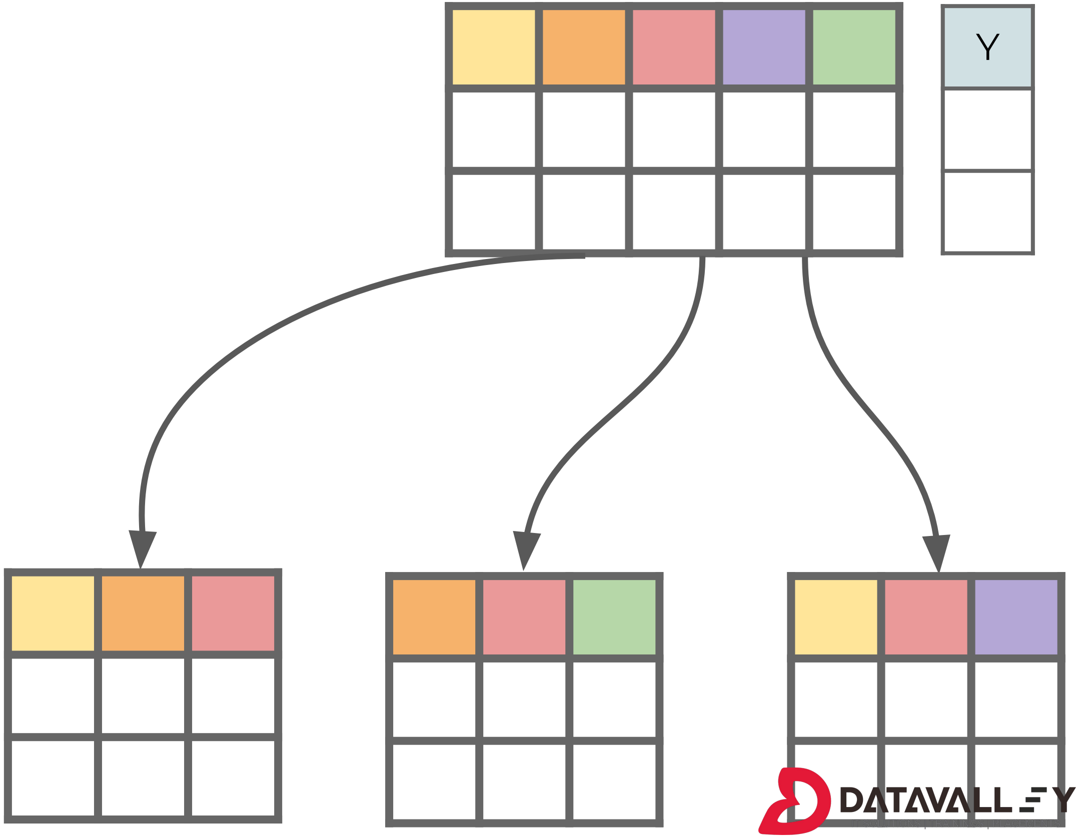
Random Forests



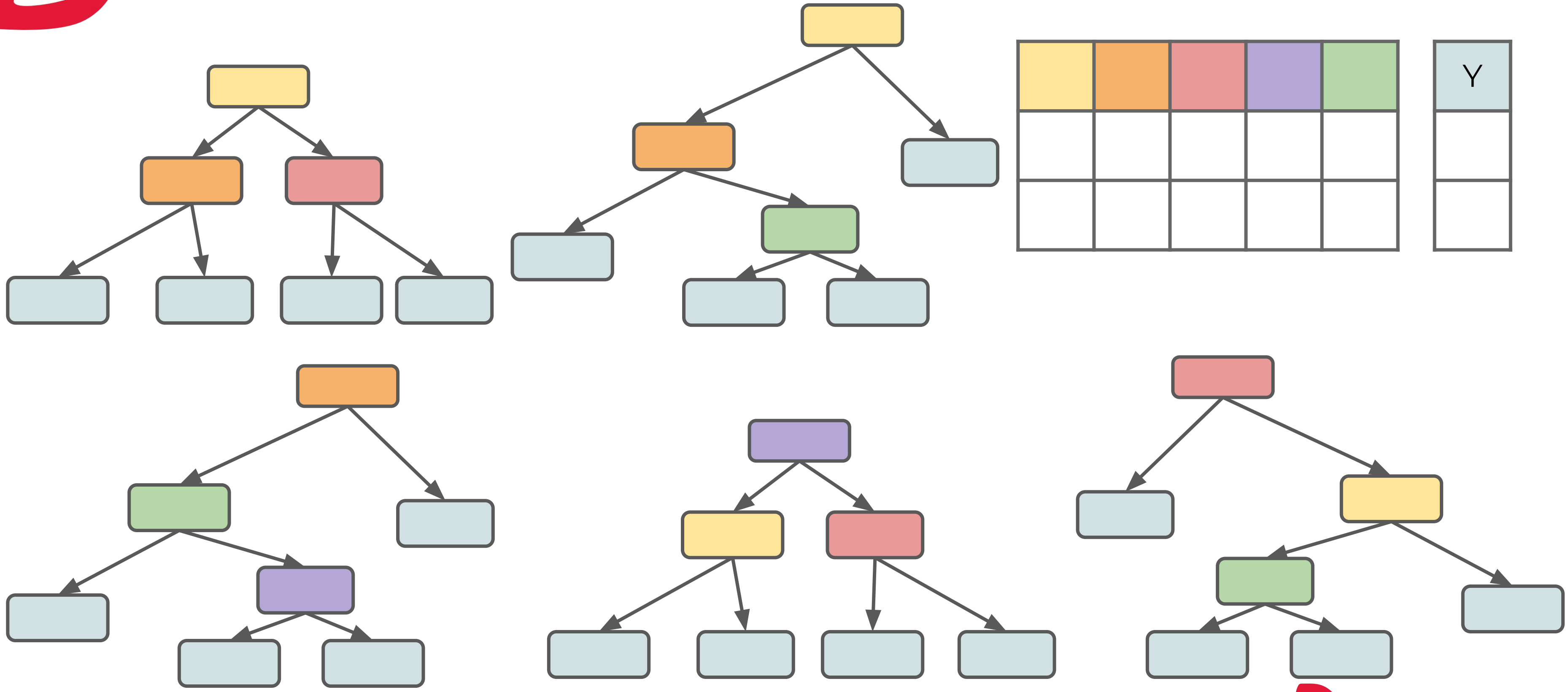


Random Forests

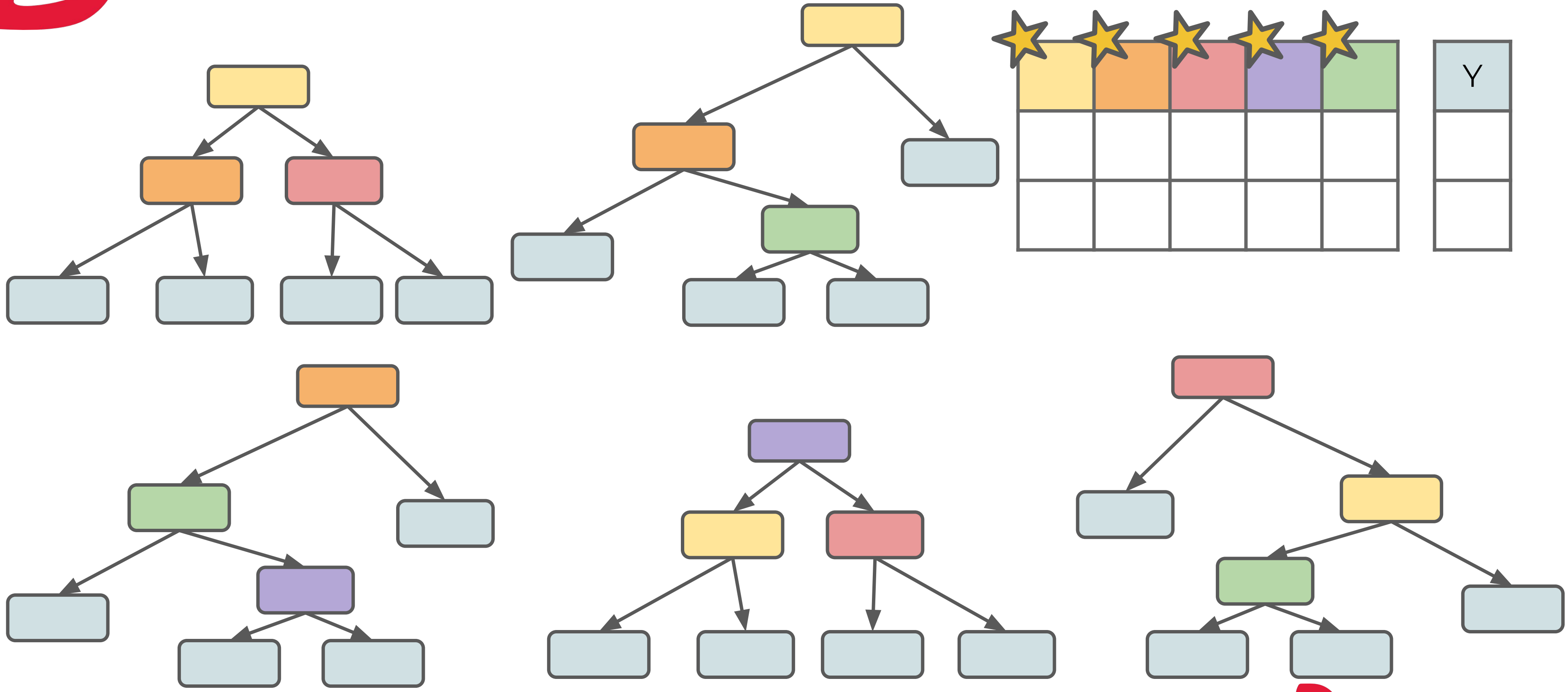




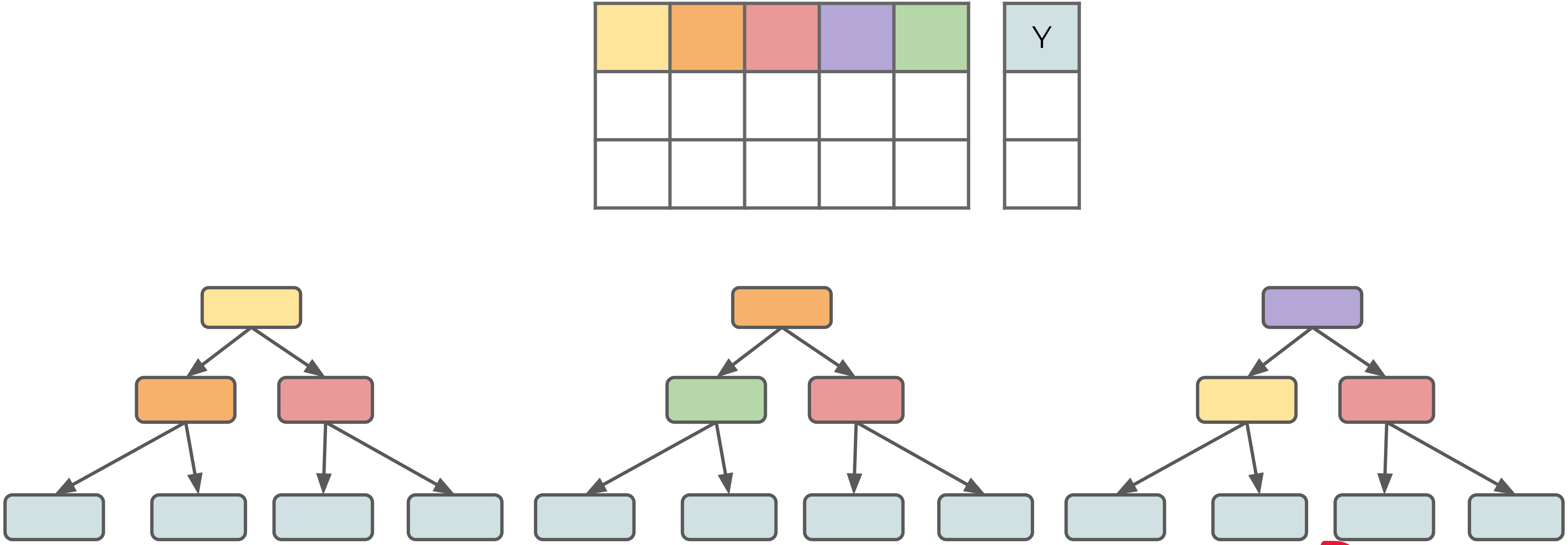
Random Forests



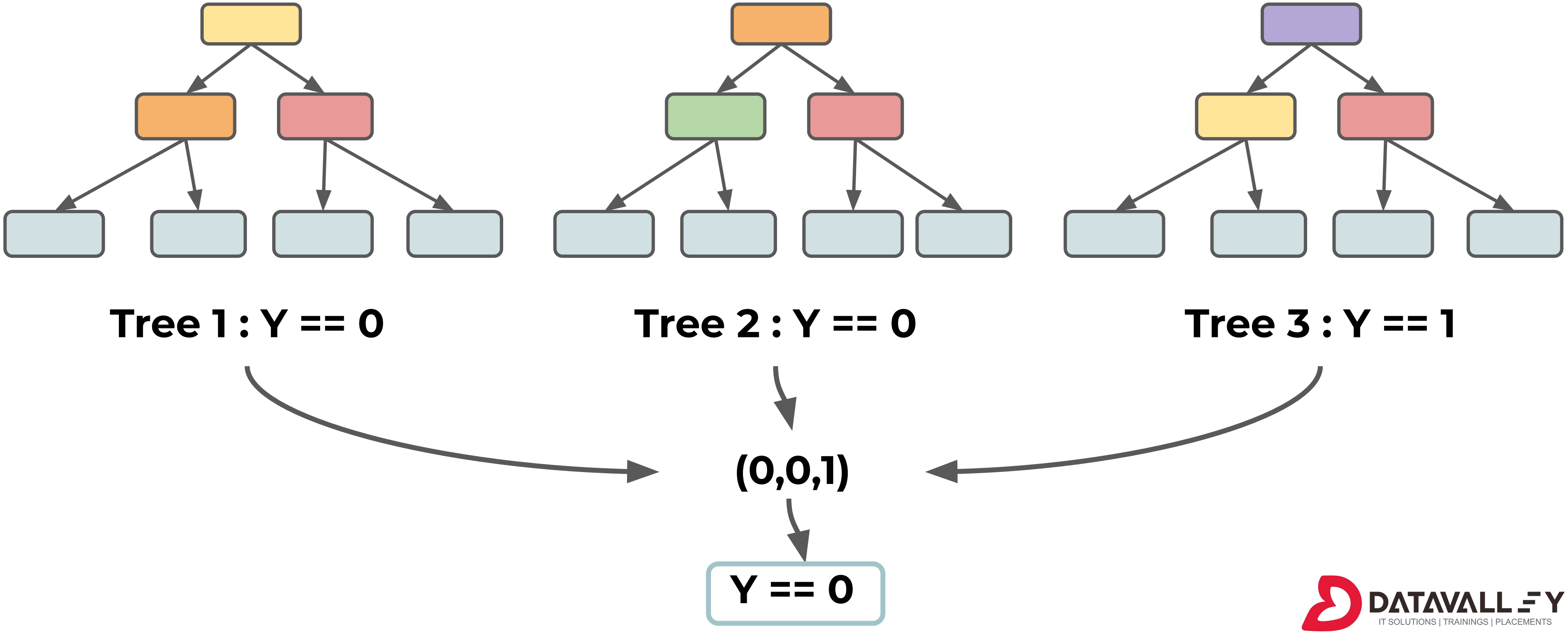
Random Forests



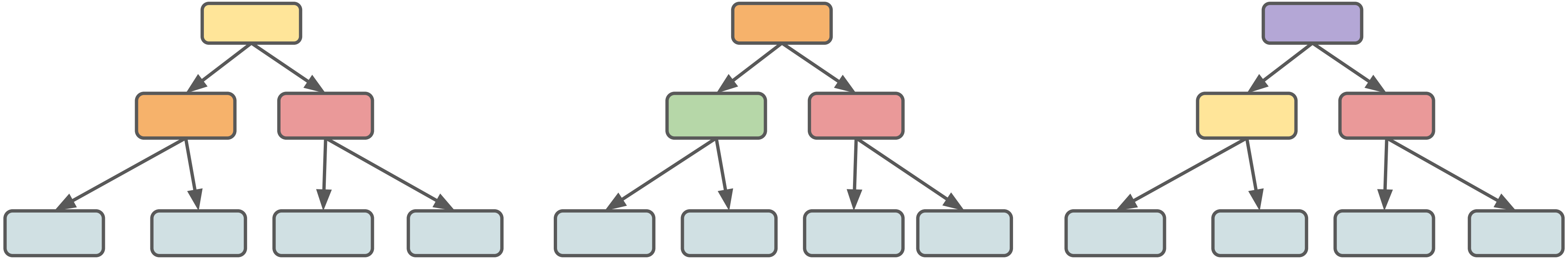
Random Forests



Random Forests



Random Forests



Tree 1 : Y == 10

Tree 2 : Y == 13

Tree 3 : Y == 11

(10,13,11)

Y == 11.3

Random Forests

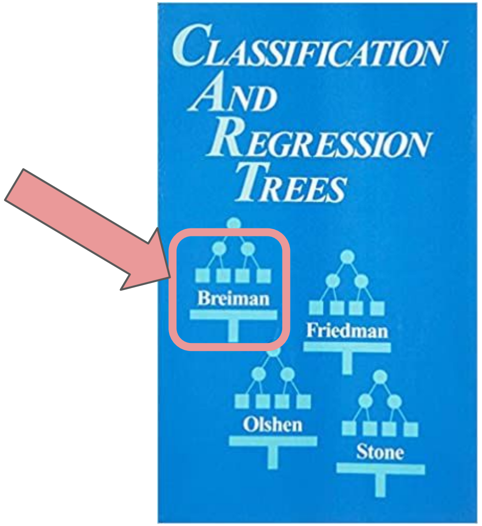
- Ho and other researchers showed that adding the random feature subspace constraint could allow trees to grow much deeper without causing overfitting.
- Mathematical explanation of this was shown by Eugene Kleinberg's theory of stochastic discrimination.

Random Forests

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

Random Forests

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



Random Forests

- Let's further explore how 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.

Random Forests

Theory and Intuition: Hyperparameters

Random Forests

- Since a Random Forest is an ensemble of many decision trees, many of the hyperparameters between both models are shared.
- Let’s explore the important hyperparameters unique to a Random Forest.

Random Forests

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

```
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

```
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

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
    - Number of Features
    - Bootstrap Samples
    - Out-of-Bag Error

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?*

Random Forests

- Random Forest Hyperparameters:
  - Bootstrap Samples
    - *Allow for bootstrap sampling of each training subset of features?*
  - Out-of-Bag Error
    - *Calculate OOB error during training?*

Random Forests

Theory and Intuition: Hyperparameters
Number of Estimators and Features

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?*

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?

Random Forests

- 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."*

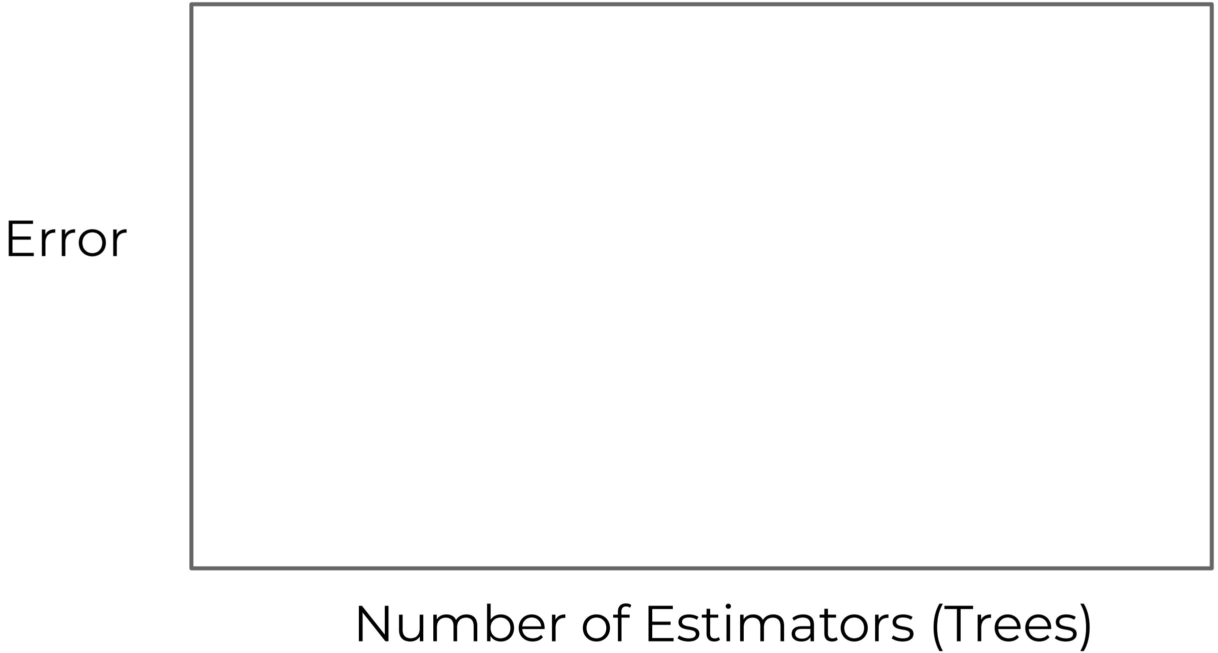
- **www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks**

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.

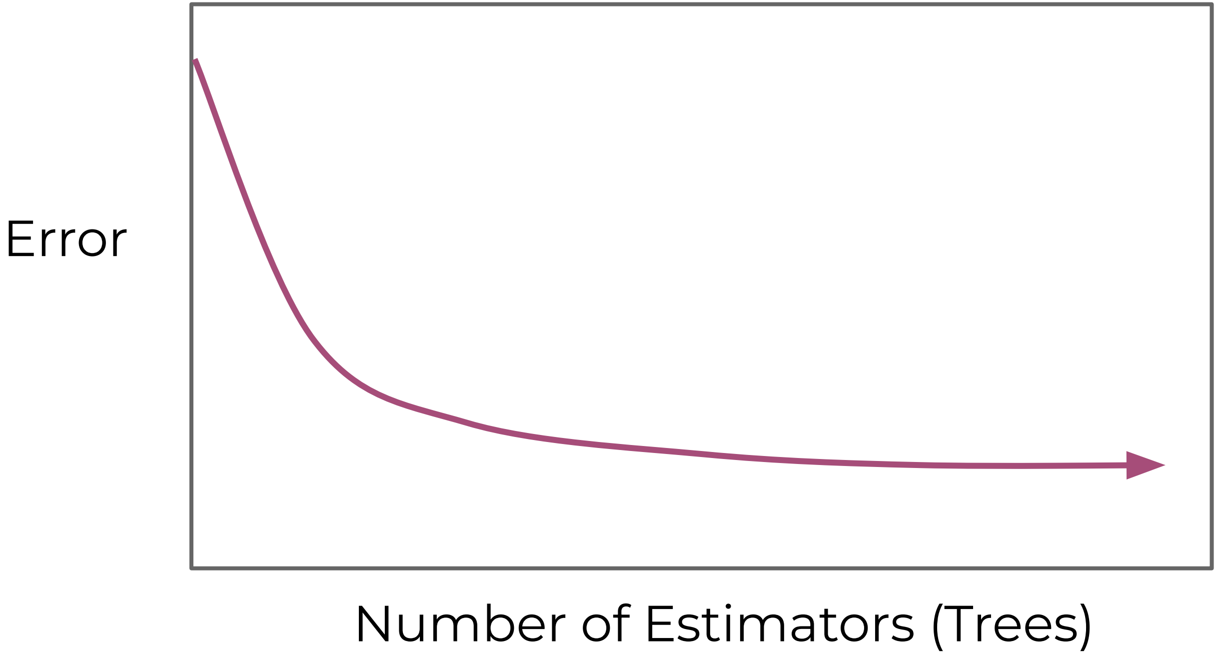
Random Forests

- Error vs. Trees



Random Forests

- Error vs. Trees



Random Forests

- 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.

Random Forests

- 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 Forests

- Random Forest Hyperparameters:
    - Number of Features
        - *How many features to include in each subset when splitting at a node?*

Random Forests

- 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.

Random Forests

- 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**

Random Forests

- 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)**.

Random Forests

- Number of Features in Subset?
  - 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.

Random Forests

- 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 Forests

- Now let's move on to exploring Bootstrapping and Out-of-Bag Error!

Random Forests

Theory and Intuition: Hyperparameters
Bootstrap Samples and OOB Error

DATAVALLEY
IT SOLUTIONS | TRAININGS | PLACEMENTS

Random Forests

- Random Forest Hyperparameters:
  - Bootstrap Samples
    - *Allow for bootstrap sampling of each training subset of features?*
  - Out-of-Bag Error
    - *Calculate OOB error during training?*

Random Forests

- Bootstrap Samples
    - *Allow for bootstrap sampling of each training subset of features?*
- First, let's understand "bootstrapping" in general terms...

Random Forests

- What is Bootstrapping?
  - A term used to describe “random sampling with replacement”.
  - Let’s see a quick example given a set of letters...

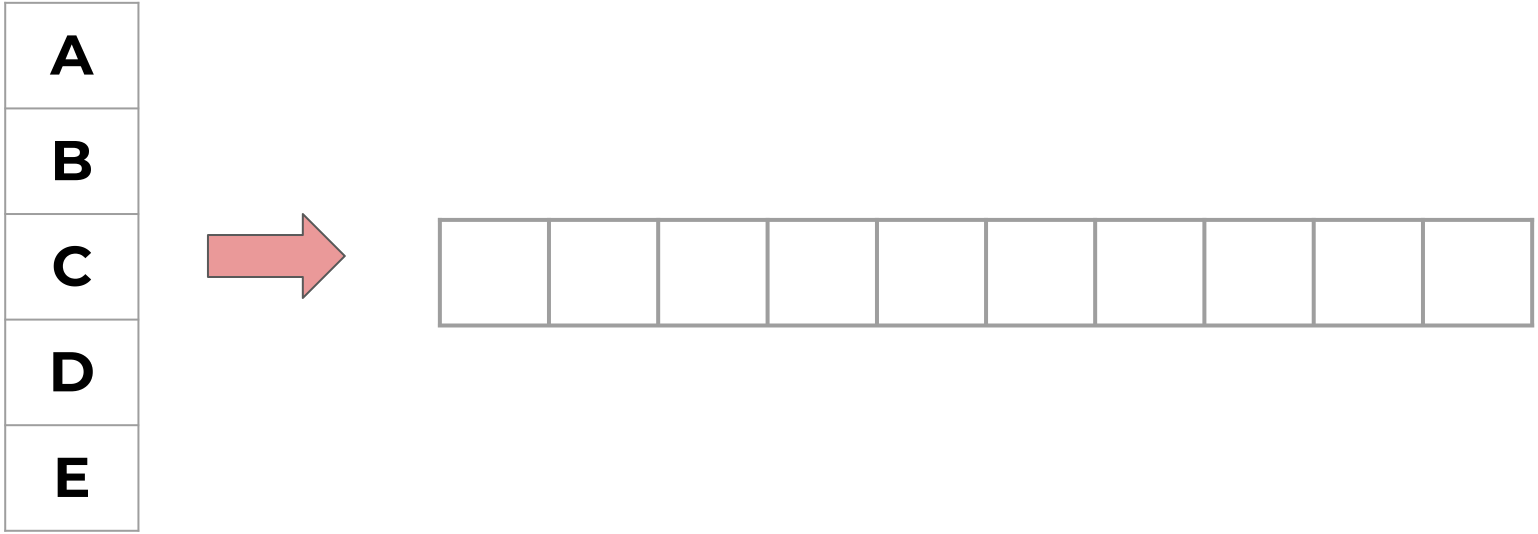
Random Forests

- What is Bootstrapping?

A
B
C
D
E

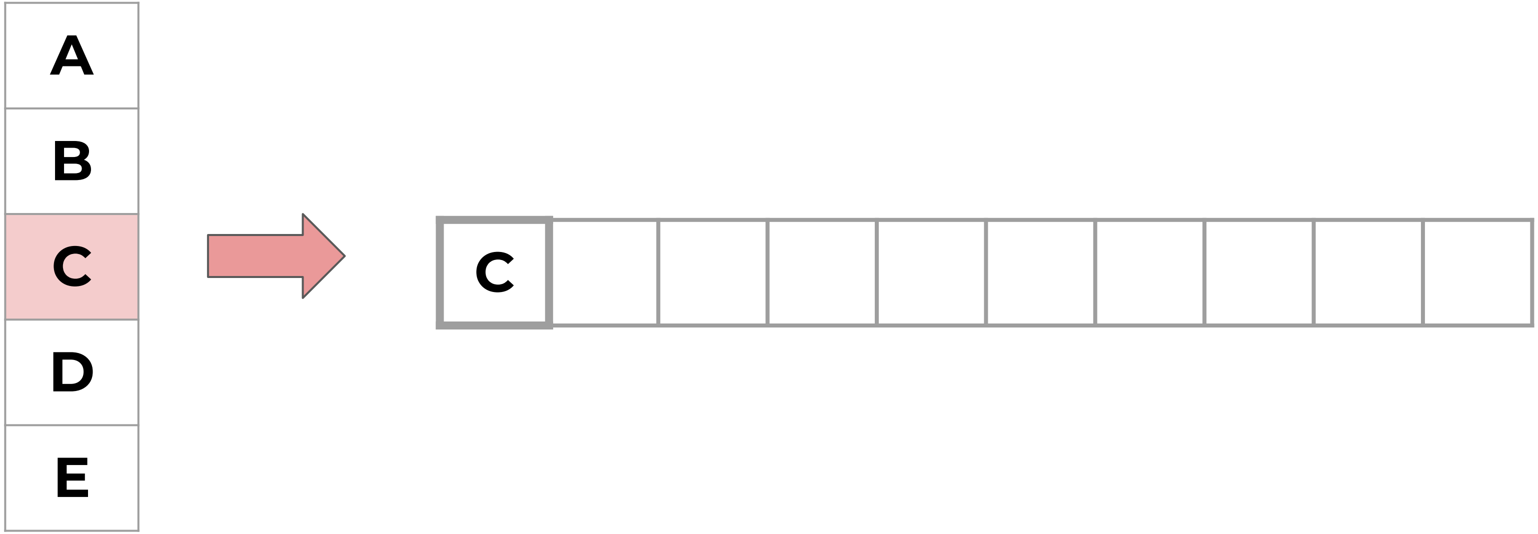
Random Forests

- What is Bootstrapping?



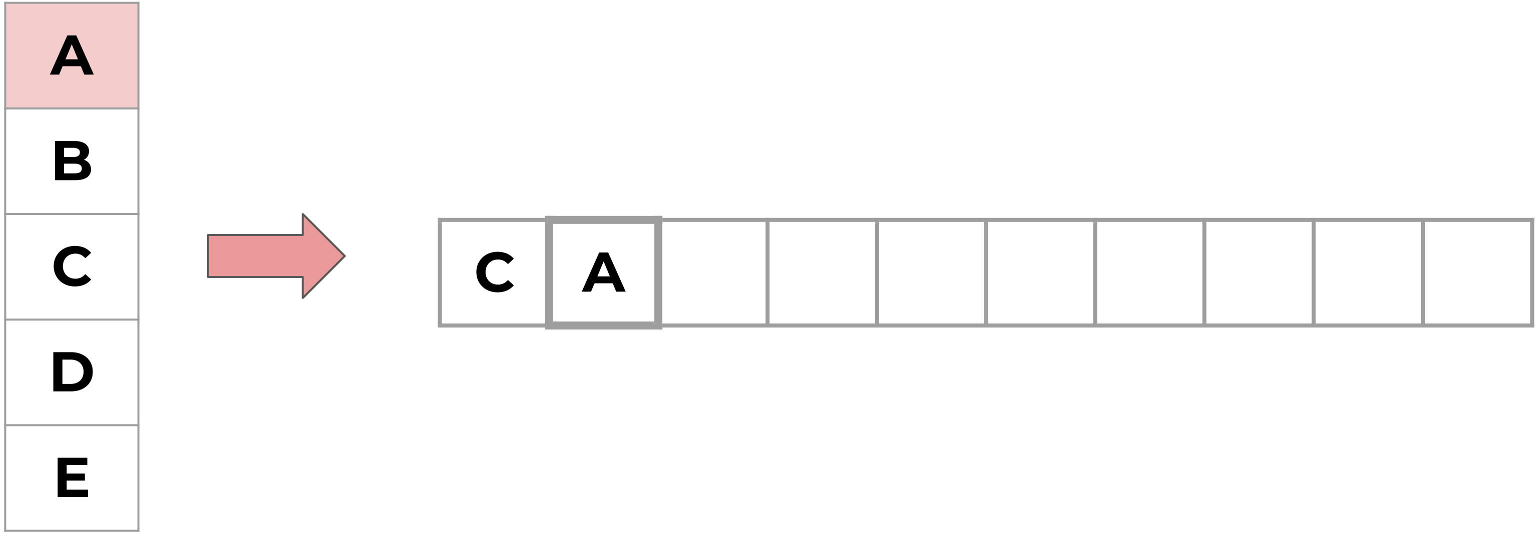
Random Forests

- What is Bootstrapping?



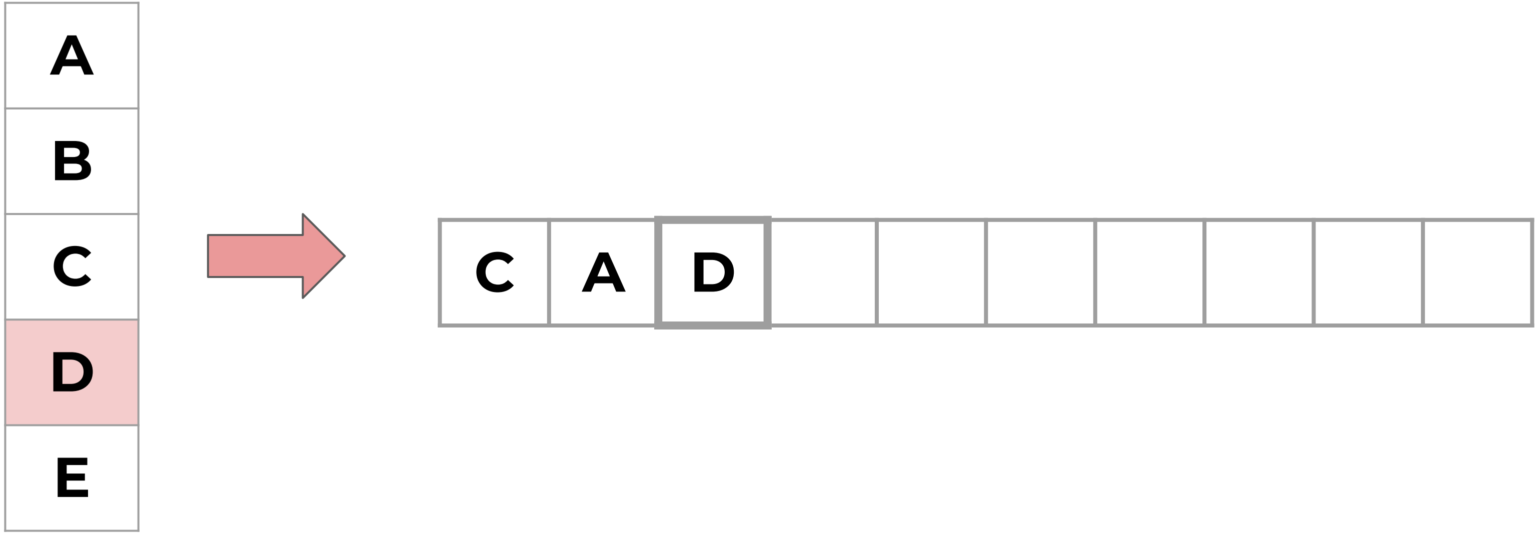
Random Forests

- What is Bootstrapping?



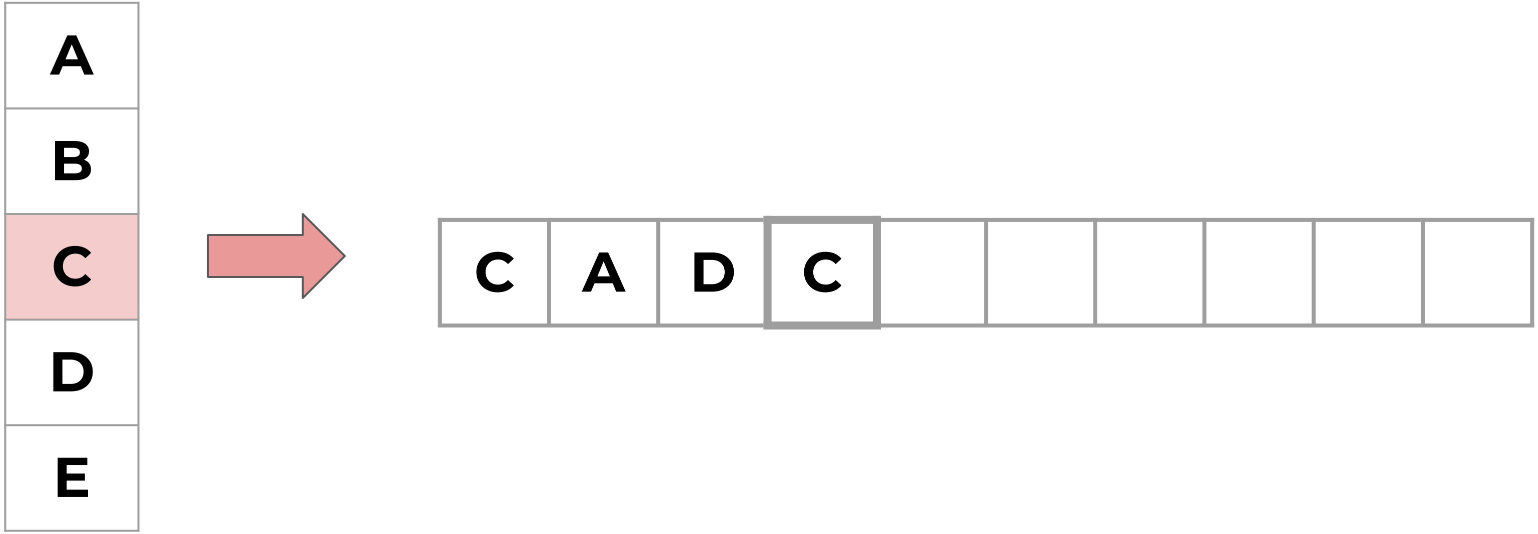
Random Forests

- What is Bootstrapping?



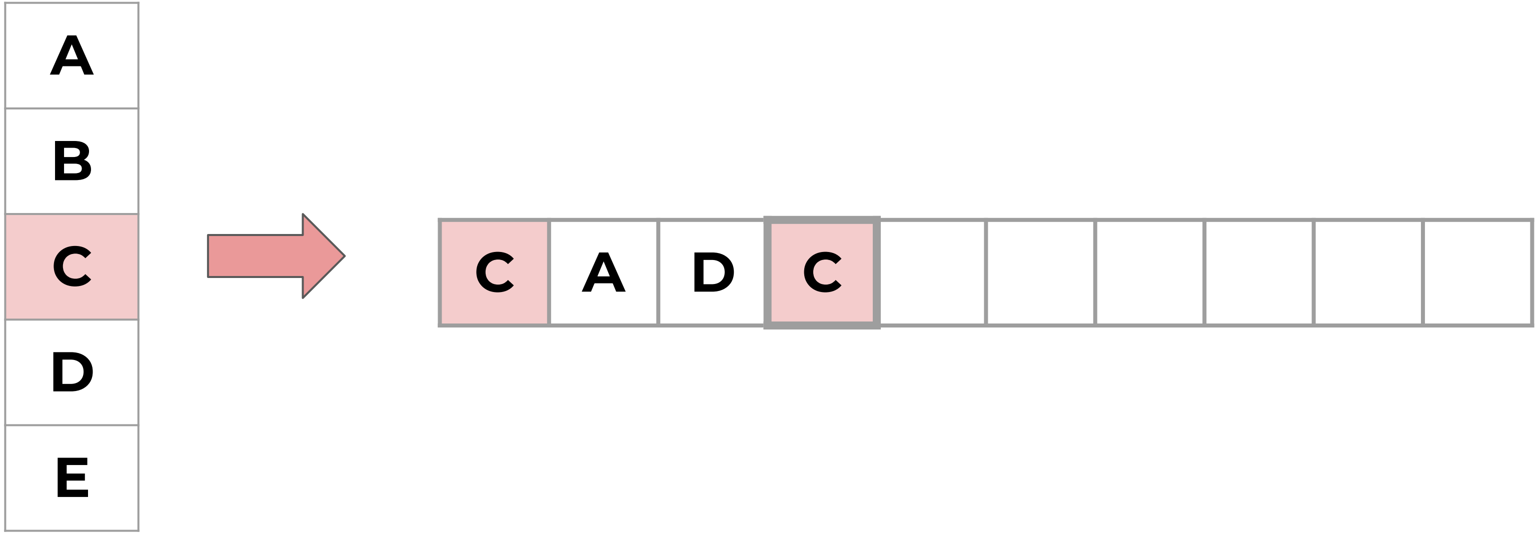
Random Forests

- What is Bootstrapping?



Random Forests

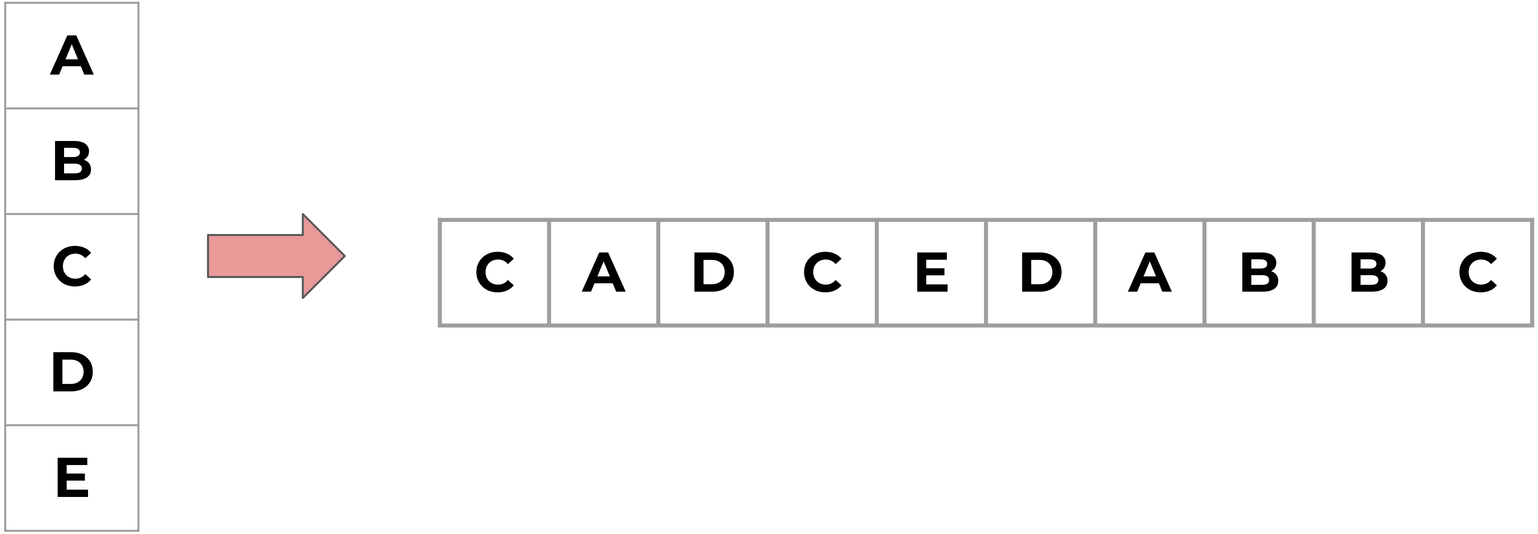
- What is Bootstrapping?



DATAVALLEY
IT SOLUTIONS | TRAININGS | PLACEMENTS

Random Forests

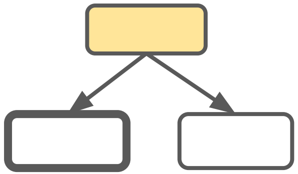
- What is Bootstrapping?

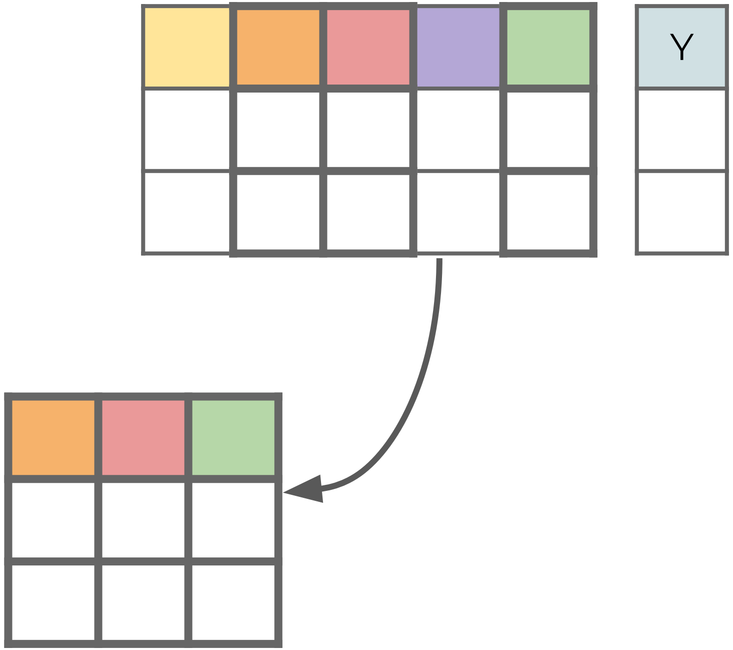


Random Forests

- 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.

Random Forests

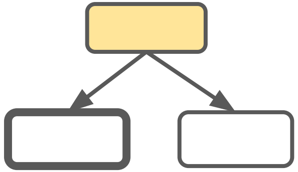




Random Forests

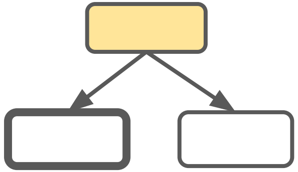
- Bootstrapping in Random Forest
  - To further differentiate trees, we could bootstrap a selection of rows for each split.
  - This results in two randomized training components:
    - Subset of Features Used
    - Bootstrapped rows of data

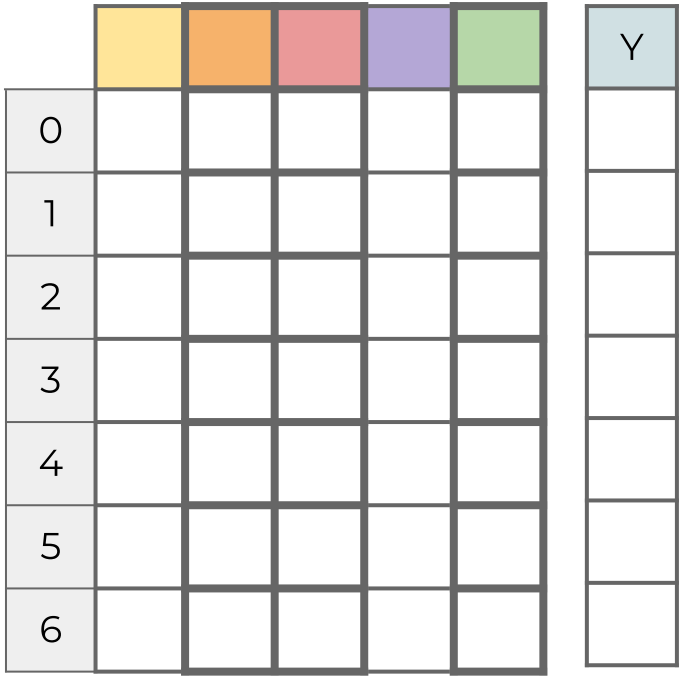
Random Forests



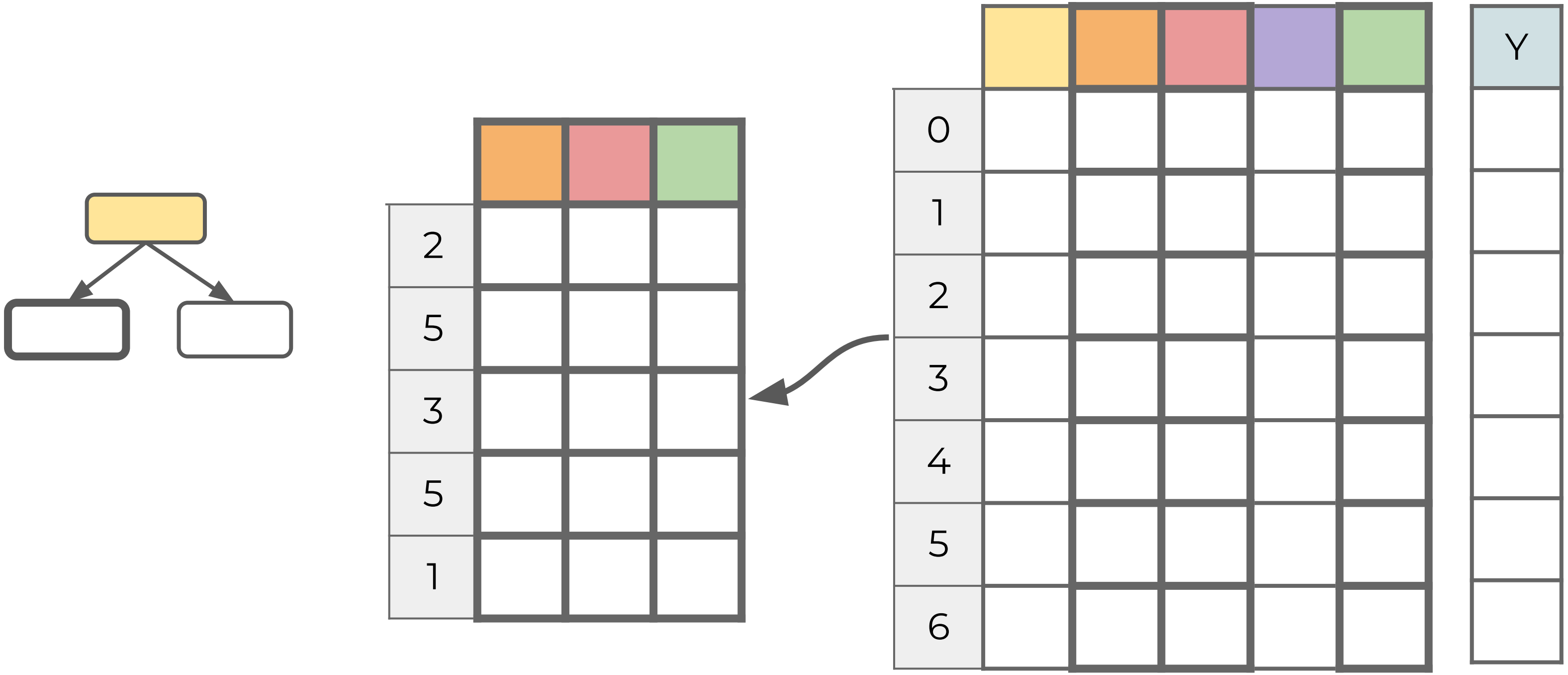
						Y
0						
1						
2						
3						
4						
5						
6						

Random Forests





Random Forests



Random Forests

- 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 Forests

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

DATAVALLEY
IT SOLUTIONS | TRAININGS | PLACEMENTS

Random Forests

- What is Bagging?
  - Recall to actually use a Random Forest, we use bootstrapped data and then calculate a prediction based on the aggregated prediction of the trees:
    - Classification: Most Voted Y Class
    - Regression: Average Predicted Ys

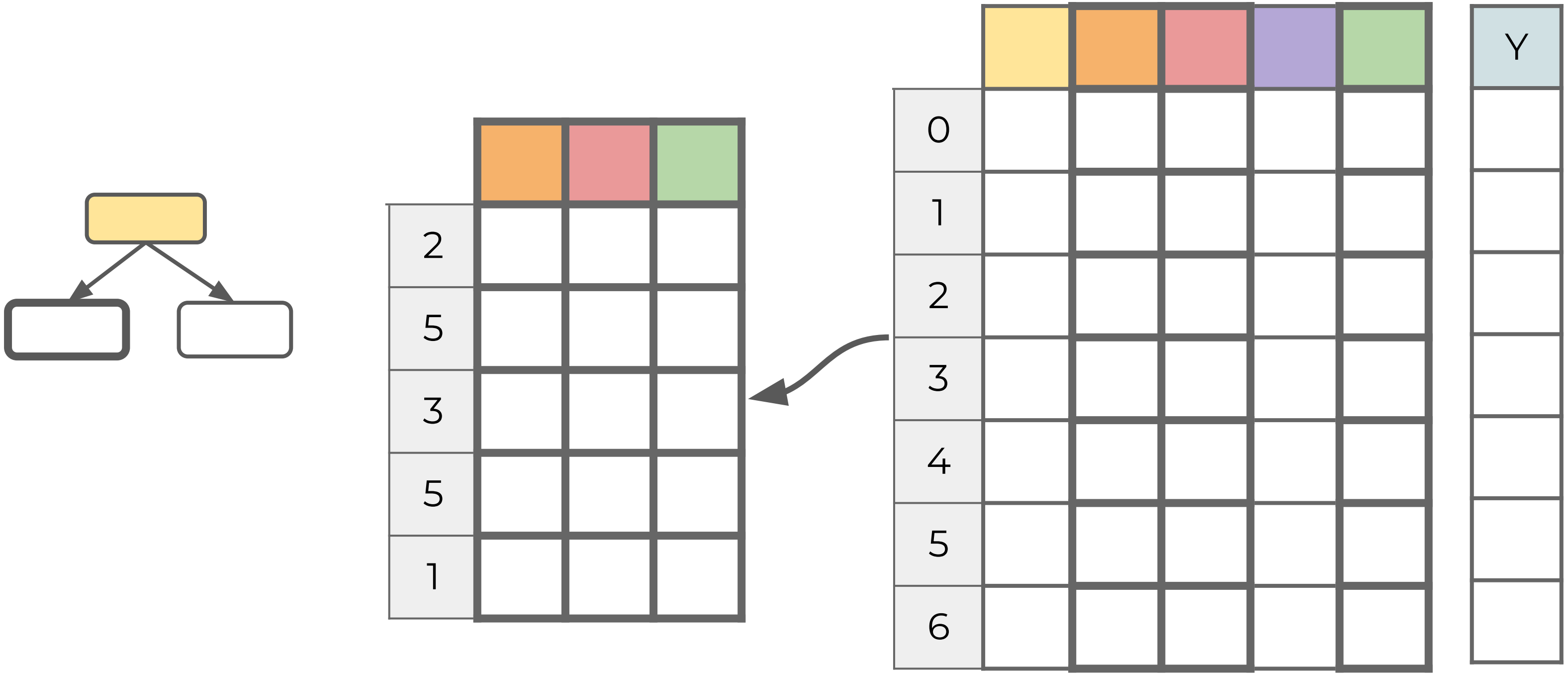
Random Forests

- 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

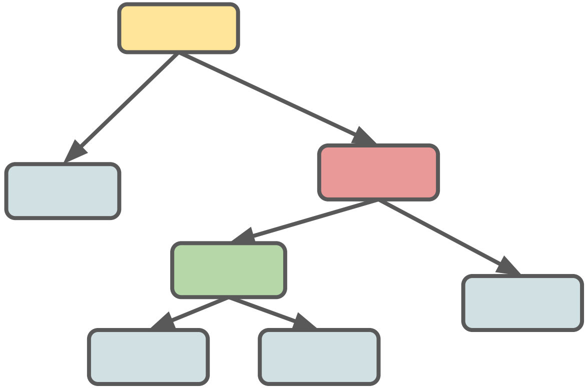
Random Forests

- What is Bagging?
  - If we performed bootstrapping when building out trees, this means that for certain trees, certain rows of data were not used for training.

Random Forests

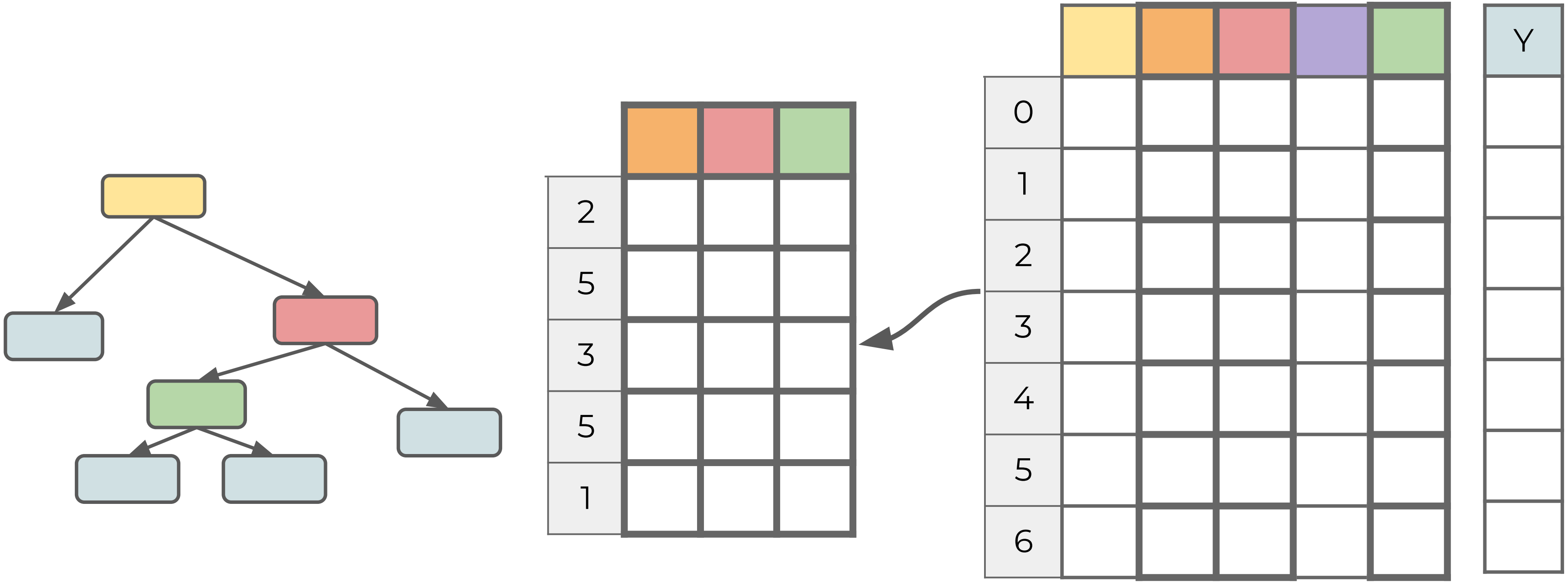


Random Forests

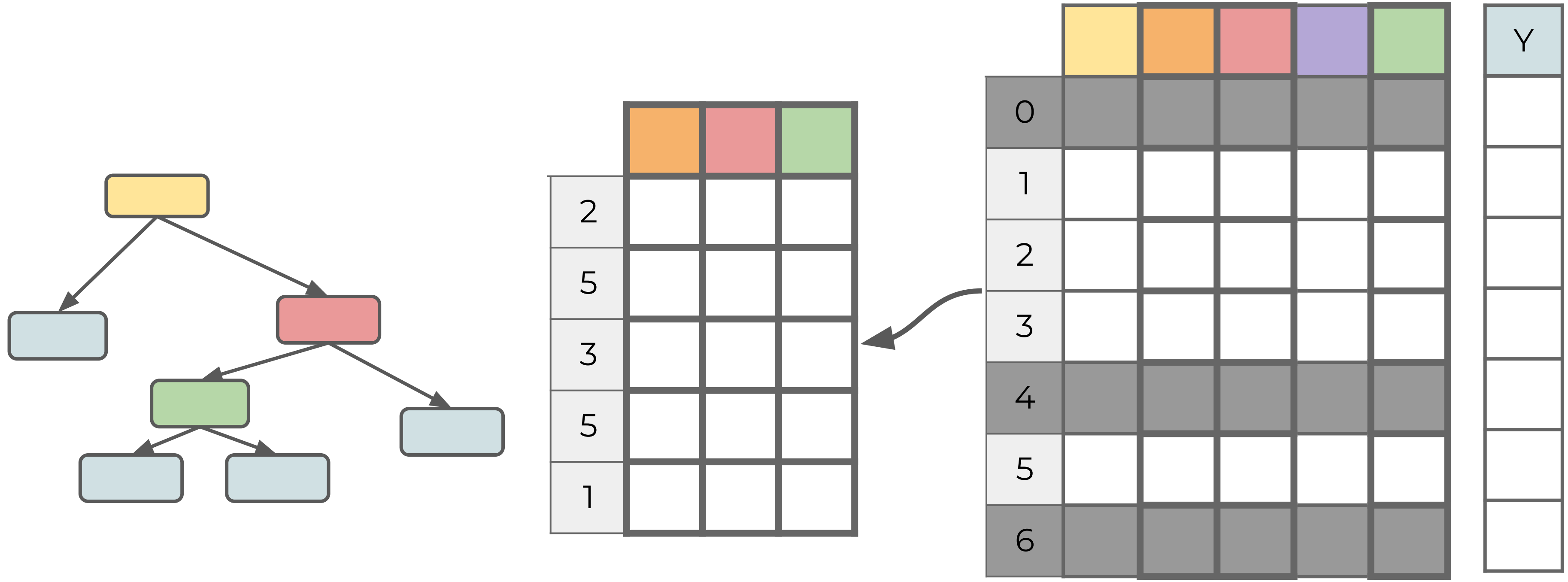


						Y
0						
1						
2						
3						
4						
5						
6						

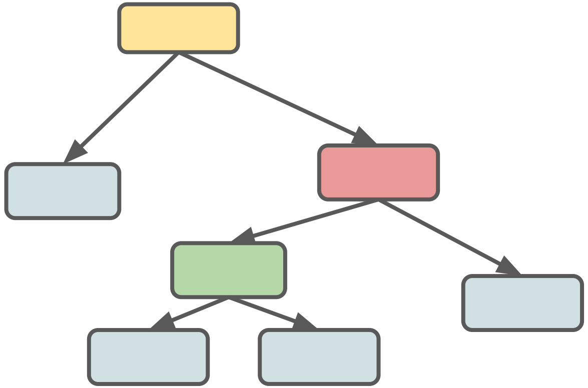
Random Forests



Random Forests

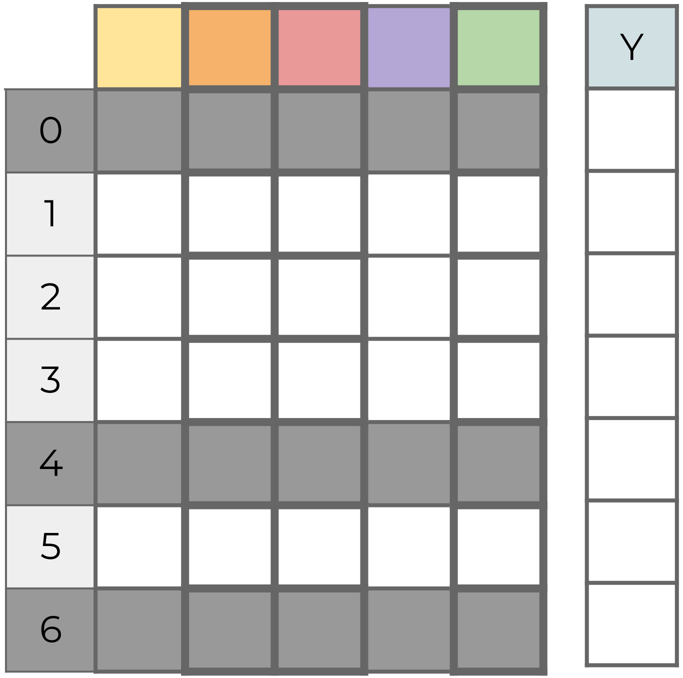


Random Forests

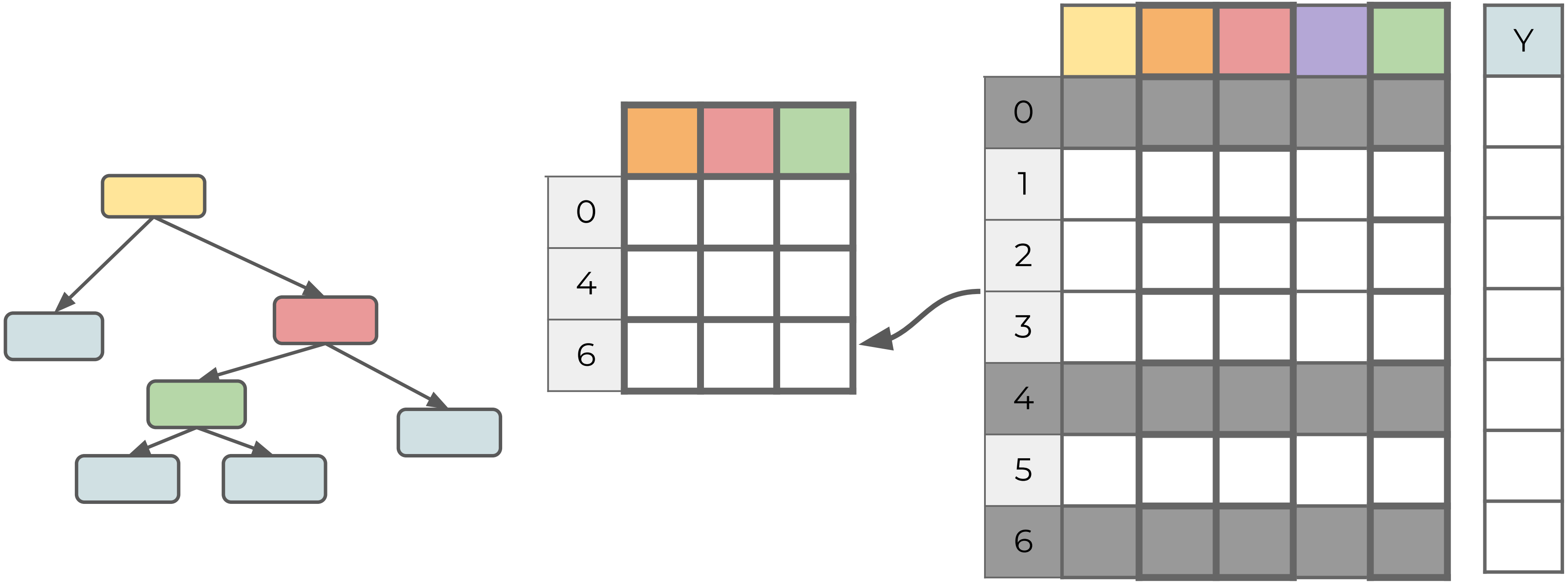


Out-of-Bag Samples

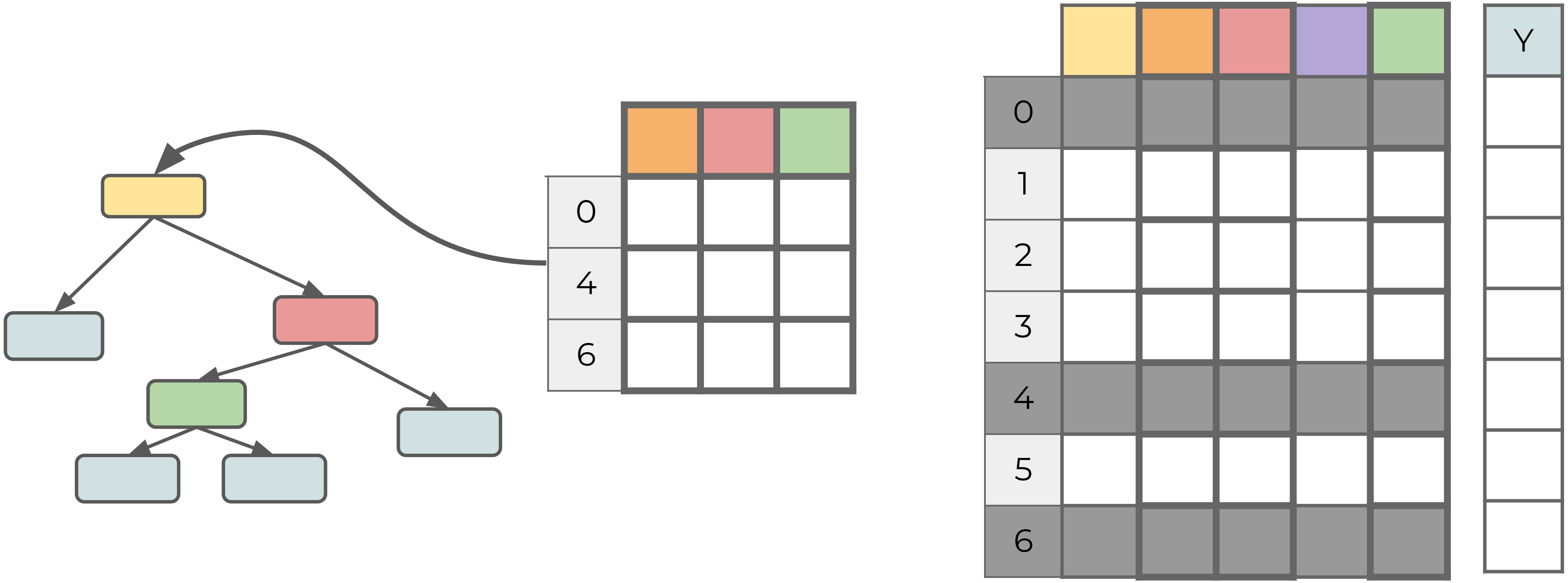
- Not used for constructing some trees.
- We could use these to get performance test metrics on trees that did not use these rows!



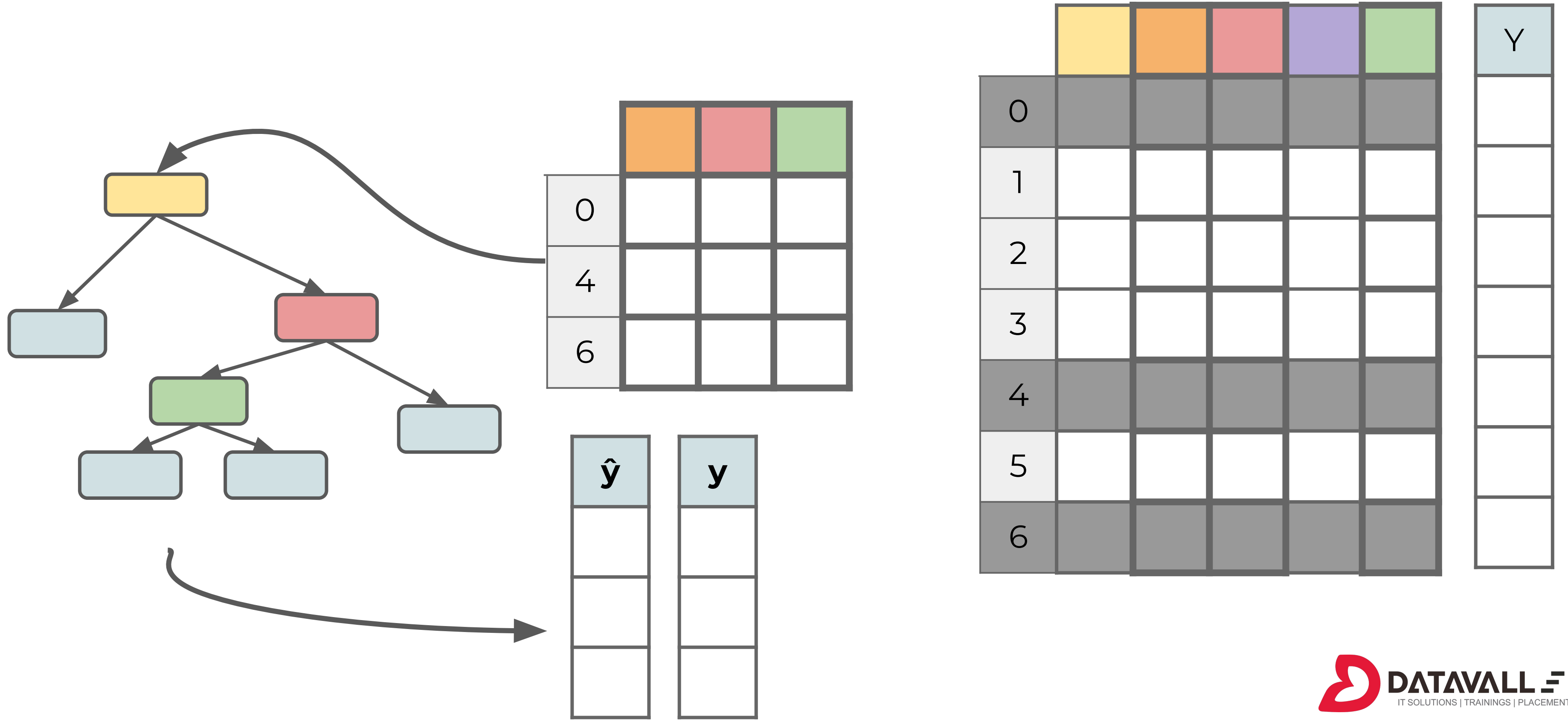
Random Forests



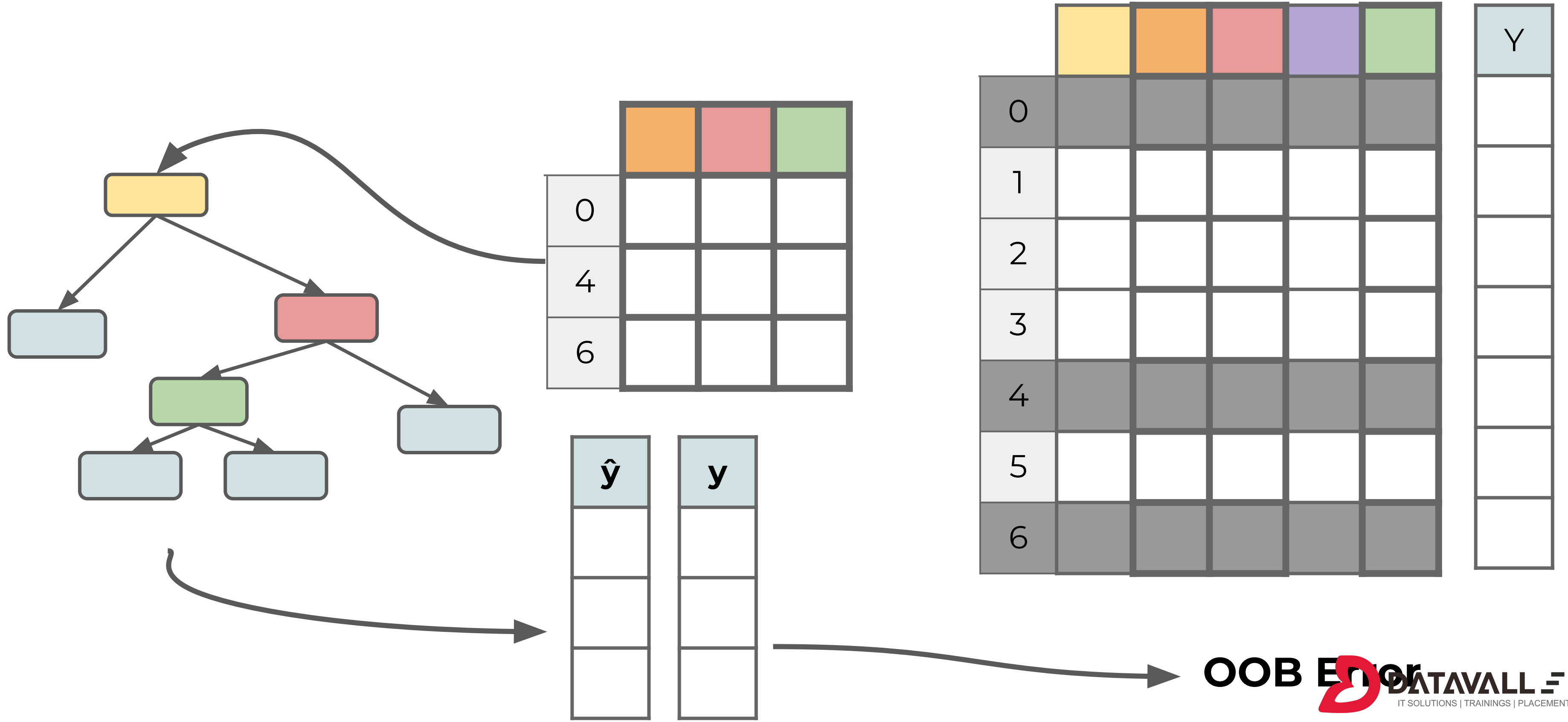
Random Forests



Random Forests



Random Forests



Random Forests

- 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.

Random Forests

- 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.

Random Forests

- We now understand the theory and intuition behind Random Forests and the specific hyperparameters we can adjust.
- Let's move on to exploring how to use Random Forests for Classification and Regression!

Random Forests

Coding Classification - Part One
Simple Random Forest Example

Random Forests

- Let’s revisit our penguin data set, but now use the Random Forest classifier.
- Afterwards, we’ll move on to a larger banknote authentication data set to explore hyperparameters like number of estimators and bootstrapping through a grid search.

Random Forests

Coding Classification - Part Two
Random Forest with Grid Search

Random Forests

- Now we'll explore a larger data set to walk through a more realistic work flow of using Random Forest which would involve a GridSearch with Cross-Validation.
- We will use the popular banknote authentication data set from the UC Irvine ML Data Repository.

Random Forests

- Banknote Authentication Dataset
  - Real and fake bills scanned and a wavelet transformation was performed on the image.

Random Forests

- Banknote Authentication Dataset Features:
    - Variance of Wavelet Transformation
    - Skewness of Wavelet Transformation
    - Curtosis of Wavelet Transformation
    - Entropy of Image

Random Forests

Regression Lecture Series Overview

Random Forests

- Let’s now explore a regression task (continuous label) with Random Forest.
- We will also compare Random Forest Regression with a wide variety of other regression models!

Random Forests

- Data - Tunnel Boring
  - We will be working with some artificial data modelling X-ray signal versus rock density.
  - Let’s get a quick idea of what situation this is modelling.

Random Forests

- Boring a new tunnel into rock

Random Forests

- Using large tunnel boring machine

Random Forests

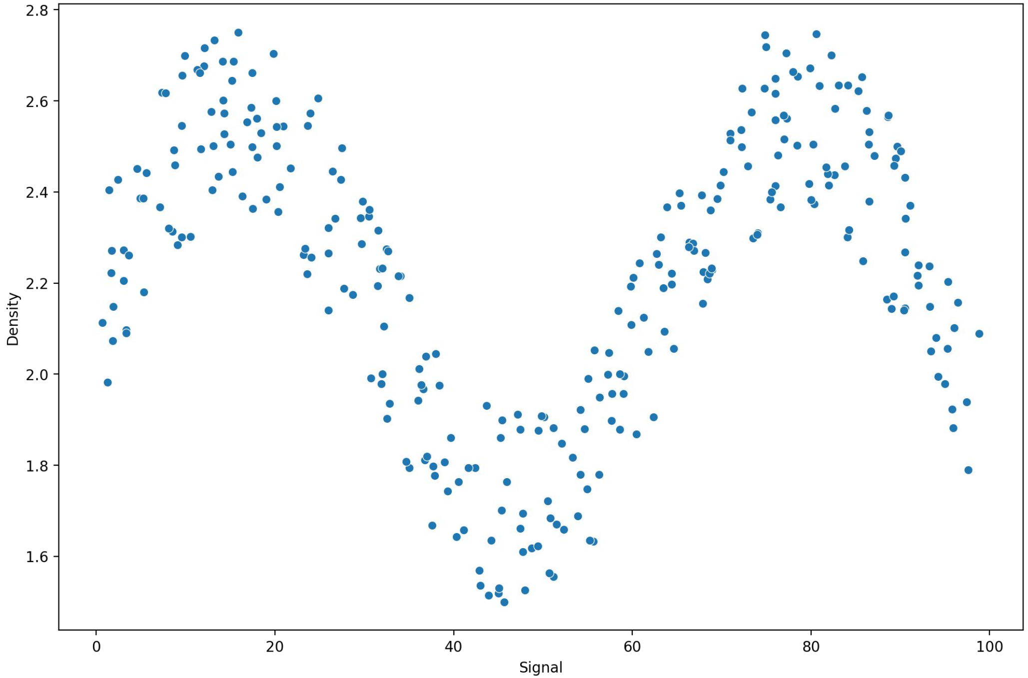
- Boring machines can have different cutting shields depending on the rock density:

Random Forests

- Use X-ray signals to determine rock density.
- Based on a rebound signal strength in nHz, we can estimate a density of rock in kg/m^3.
- We have some experimental results based on lab tests on a variety of rock samples.

Random Forests

- Resulting data looks like this:

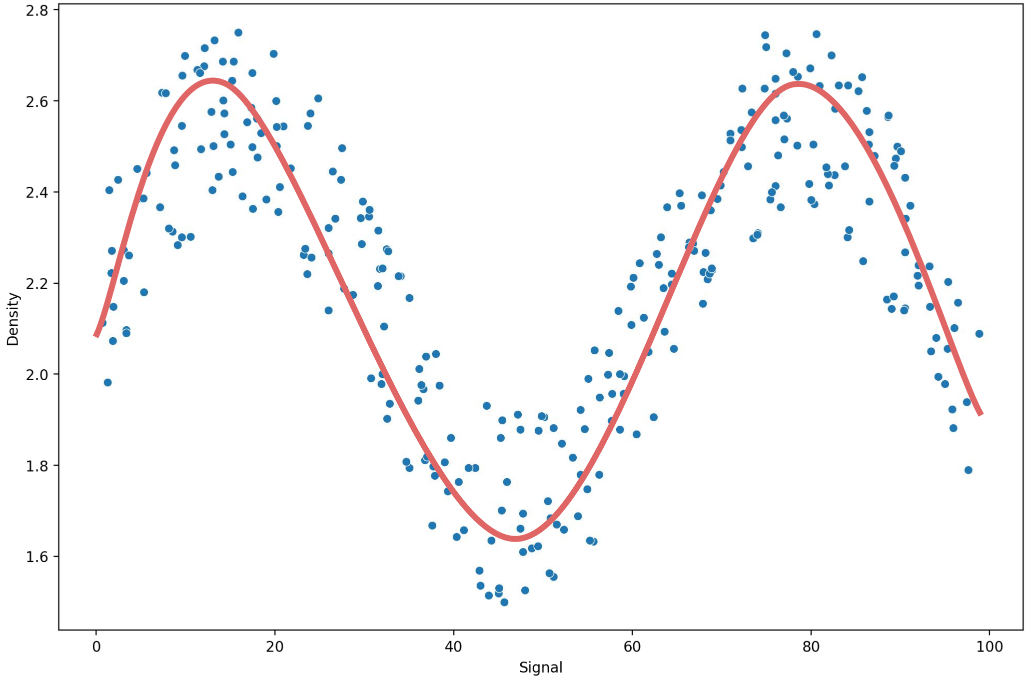


Random Forests

- Our goal is to build a generalized model, which we could then use to accept a rebound signal, and output an expected rock density.

Random Forests

- Example model on top of data:



DATAVALLEY
IT SOLUTIONS | TRAININGS | PLACEMENTS

Random Forests

- This generalized model would allow us to go out in the field, take a rebound signal measurement, and then report back an expected rock density value.
- For this lecture series, we will be comparing a variety of regression models against each other.

Random Forests

- Models Explored:
  - Linear Regression
  - Polynomial Regression
  - KNN Regression
  - Decision Tree Regression
  - Support Vector Regression
  - Boosted Trees Regression
  - Random Forest Regression

Random Forests

- *Things to keep in mind:*
  - Similar to our first studies on Linear Regression, we simply need our model to predict output for the expected range of rebound signal.
  - We'll create prediction outputs for an expected signal range (**0***nHz*-**100***nHz*)

Random Forests

- *Things to keep in mind:*
  - We will be able to easily visualize model output to compare regression outputs.
  - Keep this in mind as you compare outputs (e.g. smoothed output versus step/jagged output)