Machine Learning

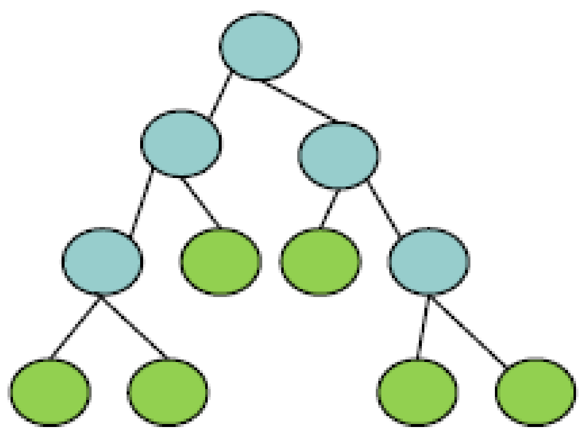
Decision Tree

Decision Tree

- One of the widely used and practical methods for classification

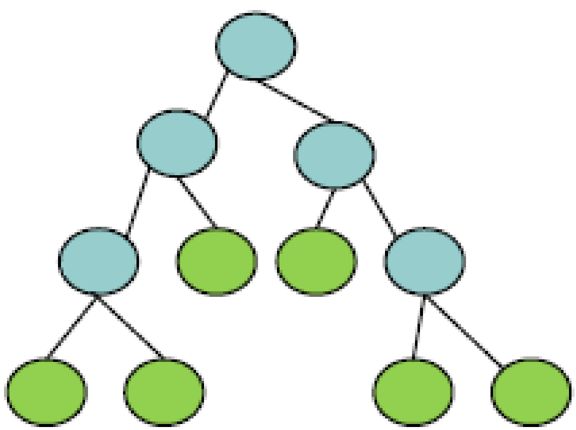
- Utilizes a tree structure to model relationships among the features and the potential outcomes (target attribute)

- Decision trees consist of nodes and branches.



Decision Tree

- Nodes are decision points
- Branches are the result of the decision function.
- The nodes are of three types:
  - Root Node (representing the original data) and a decision function
  - Branch Node (representing a decision function)
  - Leaf Node (holds the result of all the previous functions that flow to it)



Decision Tree

- Goal of a decision tree is to classify or predict an outcome based on a set of predictors

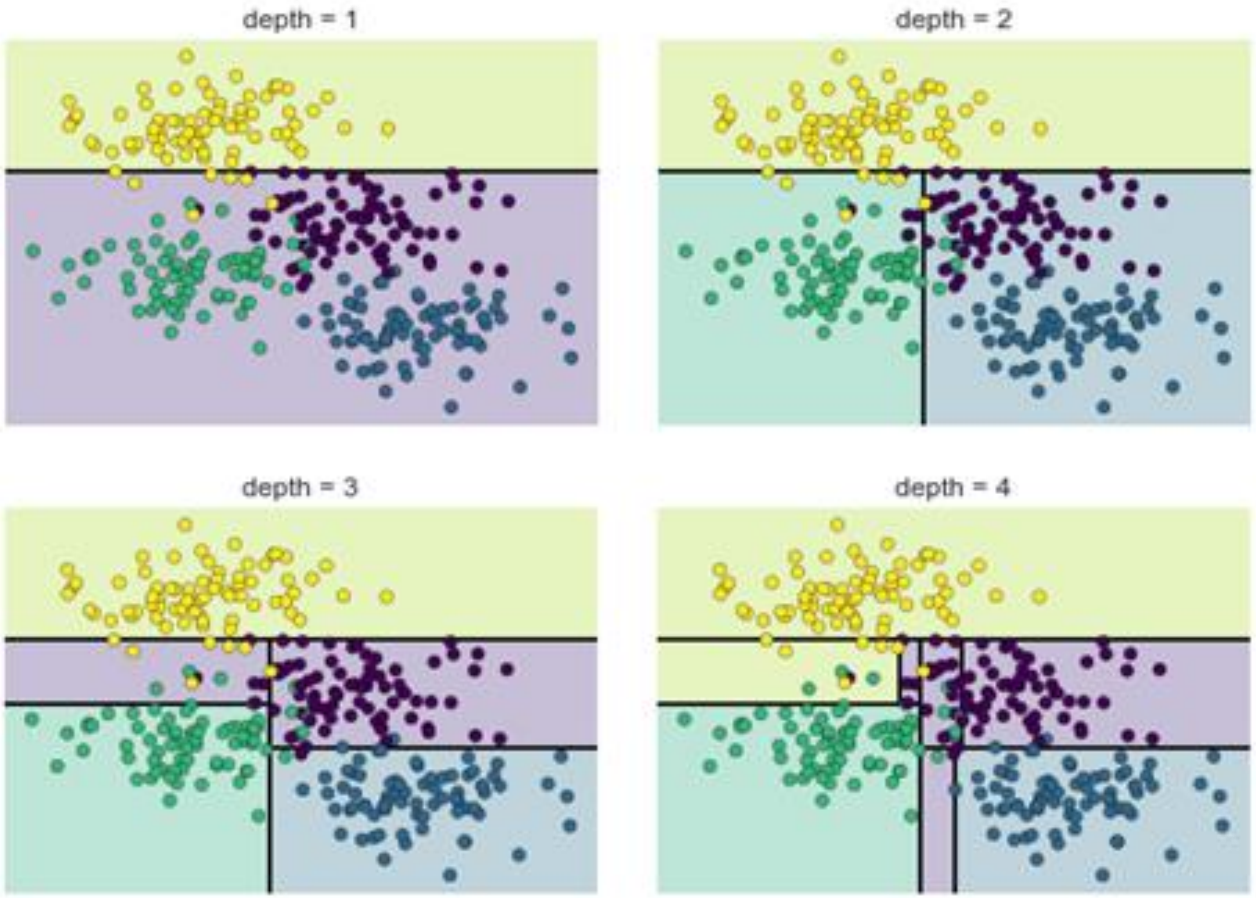
- For example: to Predict whether a customer will buy a product or not. Predictors: age of customer, credit rating etc

- Tree creation splits data into subsets and subsets into further smaller subsets.

- The algorithm stops splitting data when data within the subsets are sufficiently homogenous or some other stopping criterion is met

Decision Tree

- Decision Tree – visualize the increasing depth of tree



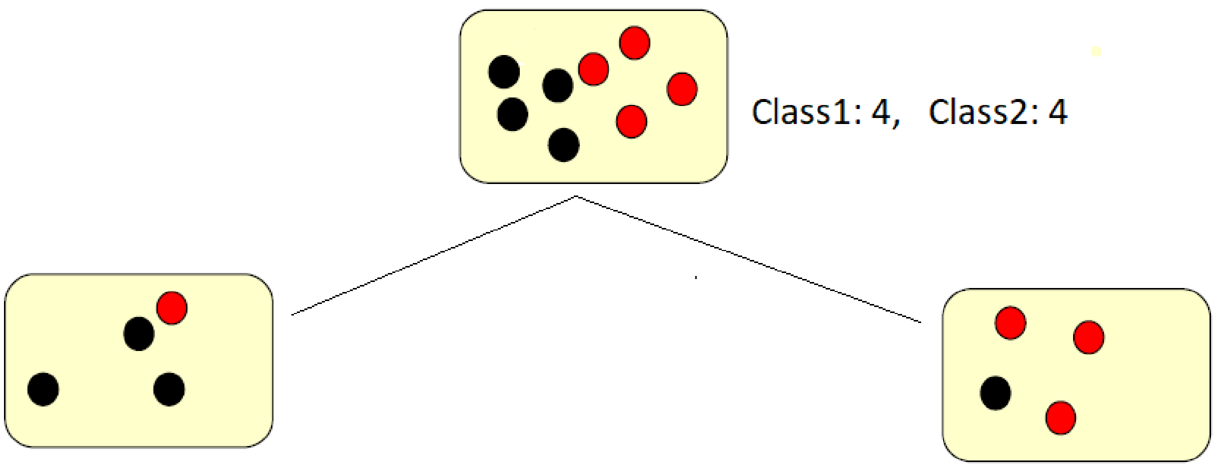
https://share.cocalc.com/share/8b892baf91f98d0cf6172b872c8ad6694d0f7204/PythonDataScienceHandbook/notebooks/05.08-Random-Forests.ipynb?viewer=share

Decision Tree

- After executing all the decision functions from Root Node to Leaf Node, the class of a data point is decided by the leaf node to which it reaches
- The leaf node may not contain all data points of same class. The Leaf Node belongs to the majority class
- Once a decision tree has been constructed, it is a simple matter to convert it into an equivalent set of rules that can be applied to a new data point and predict its class

Decision Tree Training

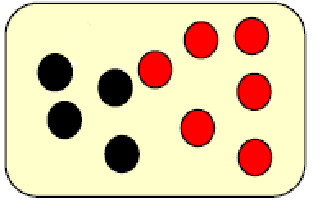
- Decision tree algorithm learns through the measure of impurity of data in a node
- Which of the following node has the most impurity?



- Impurity at a node is measured based on mixture of different classes in the target column of a node
- The objective is to minimize the impurity as much as possible at the leaf nodes

Measuring Impurity

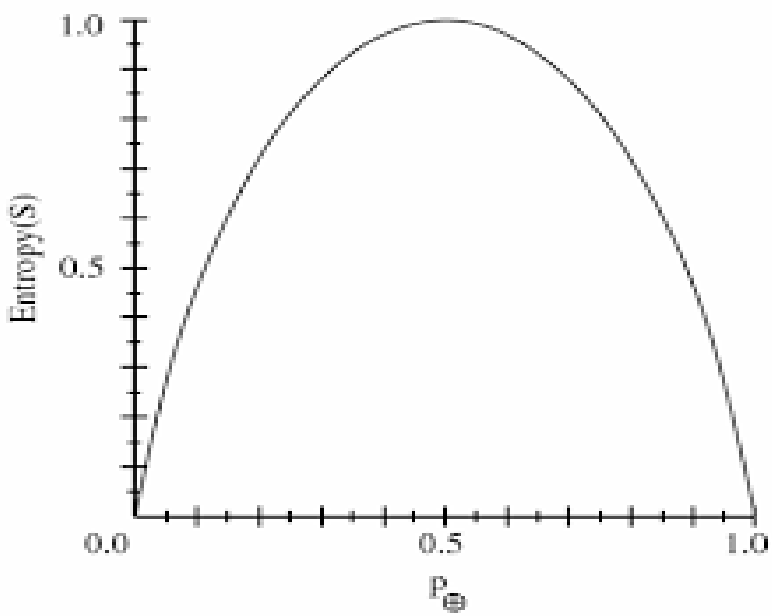
- **Entropy**
- A box contains 6 red and 4 black balls.
- (Imagine that these represent data points of two different classes)



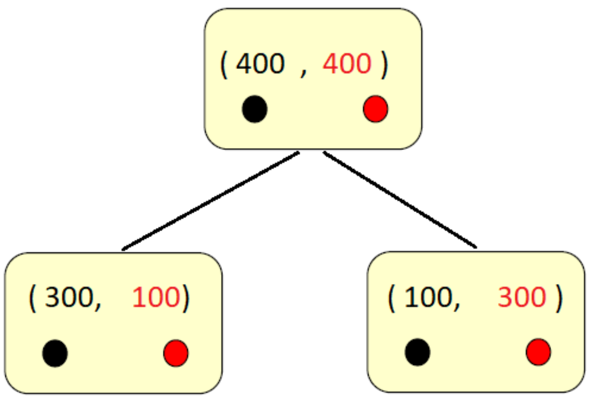
- Entropy of the box is calculated as : $Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$
- Entropy = - (0.6 * $\log_2$(0.6)) - (0.4 * $\log_2$(0.4)) = 0.9709506
- If we remove all red balls from the bag and then entropy will be
- Entropy = - 1.0 *$\log_2$(1.0) – 0.0 * $\log_2$(0) = 0
- What do you think is the interpretation of Entropy = 0?

Entropy

- Entropy ranges from 0 to 1.
- Entropy 0 means 100% information
- Entropy 1 mean maximum uncertainty
- Entropy values for a two-class variable are as follows



Entropy – Information Gain



$$I_H\left(D_p\right) = -\left(0.5\ \log_2\left(0.5\right) + 0.5\ \log_2\left(0.5\right)\right) = 1$$

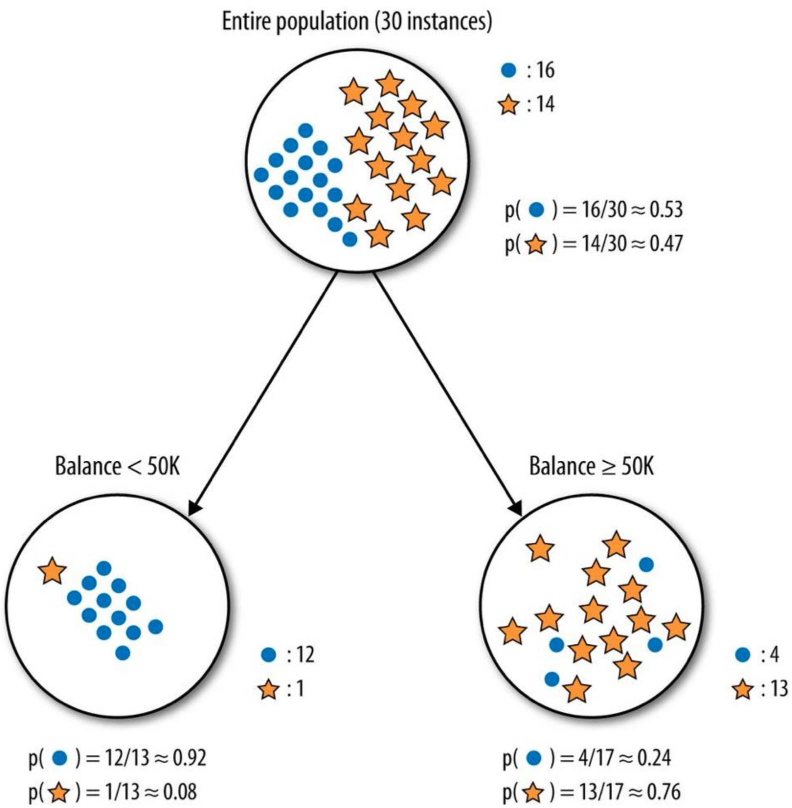
$$-\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.81$$

$$-\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) = 0.81$$

- Information Gain = $1 - \frac{4}{8}0.81 - \frac{4}{8}0.81 = 0.19$

Entropy - example

- Decision tree to predict whether a loan given to a person would result in a write-off or not. Data consists of 30 instances. 16 belong to the write-off class and the other 14 belong to the non-write-off class.



$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

$$E(Balance) = \frac{13}{30}\times 0.39 + \frac{17}{30}\times 0.79 = 0.62$$

Information Gain:

$$IG(Parent, Balance) = E(Parent) - E(Balance) = 0.99 - 0.62 = \mathbf{0.37}$$

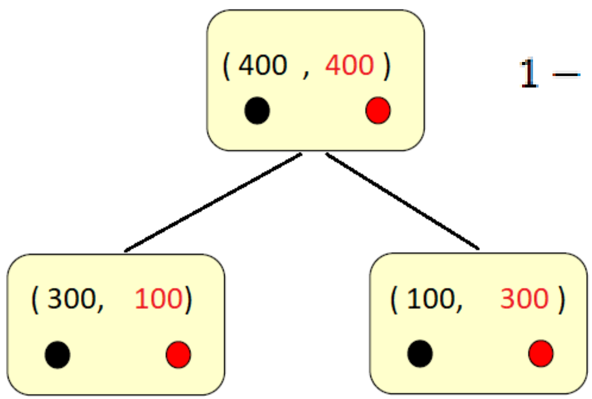
Gini Index

- Gini index is another measure which gives similar results as Entropy
- It is calculated by subtracting the sum of the squared probabilities of each class from one

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

- When uncertainty is highest, i.e. when data is evenly distributed in a node, value will be 0.5
- In perfectly classified node, values will be 0

Gini Index – Information Gain



$$1-(0.5^2+0.5^2)=0.5$$

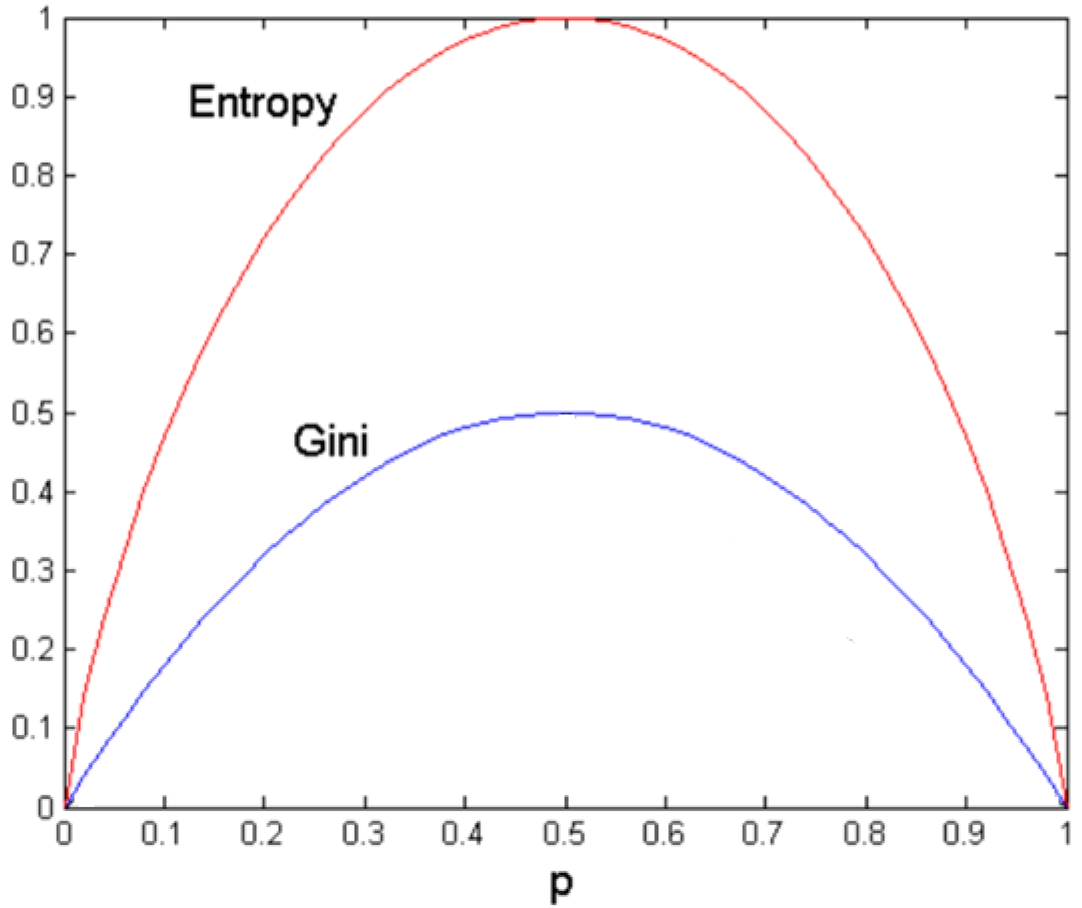
$$1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)=\frac{3}{8}=0.375$$

$$1-\left(\left(\frac{1}{4}\right)^2+\left(\frac{3}{4}\right)^2\right)=\frac{3}{8}=0.375$$

Information Gain = $0.5-\frac{4}{8}0.375-\frac{4}{8}0.375=0.125$

Entropy and Gini Index

- Comparison for 2-class problem



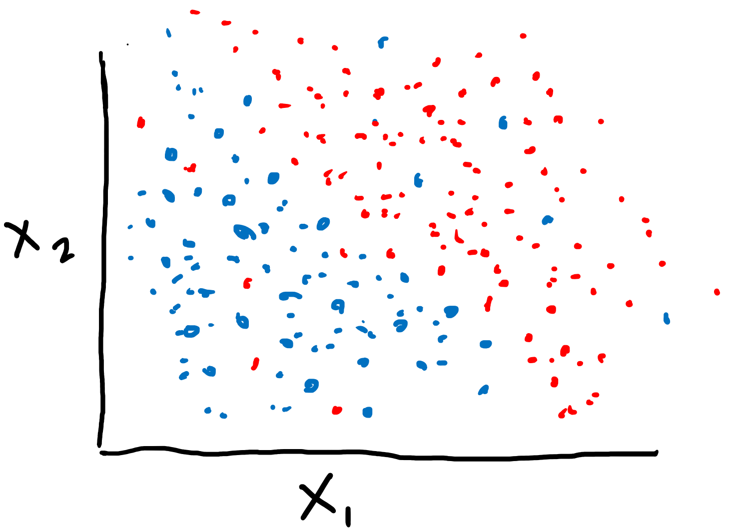
Decision Tree Algorithms

- ID3 (Iterative Dichotomiser 3) was developed in 1986 by Ross Quinlan. The algorithm creates a multiway tree, finding for each node (i.e. in a greedy manner) the categorical feature that will yield the largest information gain for categorical targets

- C4.5 is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute

- C5.0 is latest version release under a **proprietary license**. It uses less memory and builds smaller rulesets than C4.5 while being more accurate

- CART (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables. It creates binary tree. Available in Scikit-learn


x2
x1

Decision Tree

- Advantages :
  - Simple to understand and to interpret
  - Trees can be visualized
  - Able to handle both numerical and categorical data.
  - Requires less data preparation. Does well with missing data
  - Does not assume relationship between features and target variables

- Disadvantages:
  - Decision trees can be unstable and tend to overfit. (Mitigation: Use decision trees within an ensemble)
  - Decision tree learners create biased trees if some classes dominate
  - Divide feature space in axis parallel boundaries which may not be optimum

Decision Tree - Parameters

- max_depth – Is the maximum length of a path from root to leaf (in terms of number of decision points. The leaf node is not split further. It could lead to a tree with leaf node containing many observations on one side of the tree, whereas on the other side, nodes containing much less observations get further split

- min_sample_split - A limit to stop further splitting of nodes when the number of observations in the node is lower than this value

- min_sample_leaf – Minimum number of samples a leaf node must have. When a leaf contains too few observations, further splitting will result in overfitting (modeling of noise in the data)

- max_leaf_nodes – maximum number of leaf nodes in a tree

Machine Learning

Ensemble Techniques

Ensemble - Background

- What is Ensemble?
  - Do not predict using a single classifier but learn a set of classifiers
  - An ensemble of classifiers is created by combining predictions of multiple classifiers for improving prediction performance
- Why Ensemble?
  - Combining the outputs of several classifiers may reduce the risk of selecting a poorly performing classifier
  - The errors made while classifying instances by one classifier are generally averaged out by the correct classification of another classifier, so that the overall classification accuracy is improved

Ensemble - Background

- In some parts of the feature space, the different classifiers produce similar results
- In regions where the data points from different classes overlap, the classifiers give different results
- By using information from multiple classifiers, the result may be better than an individual classifier

Ensemble

- For Ensemble to work successfully, we need to ensure each learner (classifier, in this case) is slightly different

- How to achieve this?
    - Provide different data to different classifiers
    - Perform random sampling with replacement of rows
    - Provide different features as input to different classifiers
    - By adjusting the weights assigned to each data point to force an instance to focus on certain data points more

Bagging

- Bagging term is made from - **B**ootstrap **Agg**regation
- It is used to reduce the variance of a decision tree
- It creates several subsets of data from training sample chosen randomly with replacement.
- Each subset data is used to train a decision tree
- As a result, we have an ensemble of different models.
- For classification, bagging is used with voting to decide the class of an input while for regression average or median values are calculated
- This concept can be extended for other models too, but is commonly used for Decision Tree

Bootstrap sampling

- Generate new training sets using sampling with replacement. It is called bootstrap sampling

Datasets created using Bootstrap Sampling

Original dataset	
Row no	
1	....
2	....
3	....
4	....
5	....
6	....
7	....
8	....
9	...
10	...

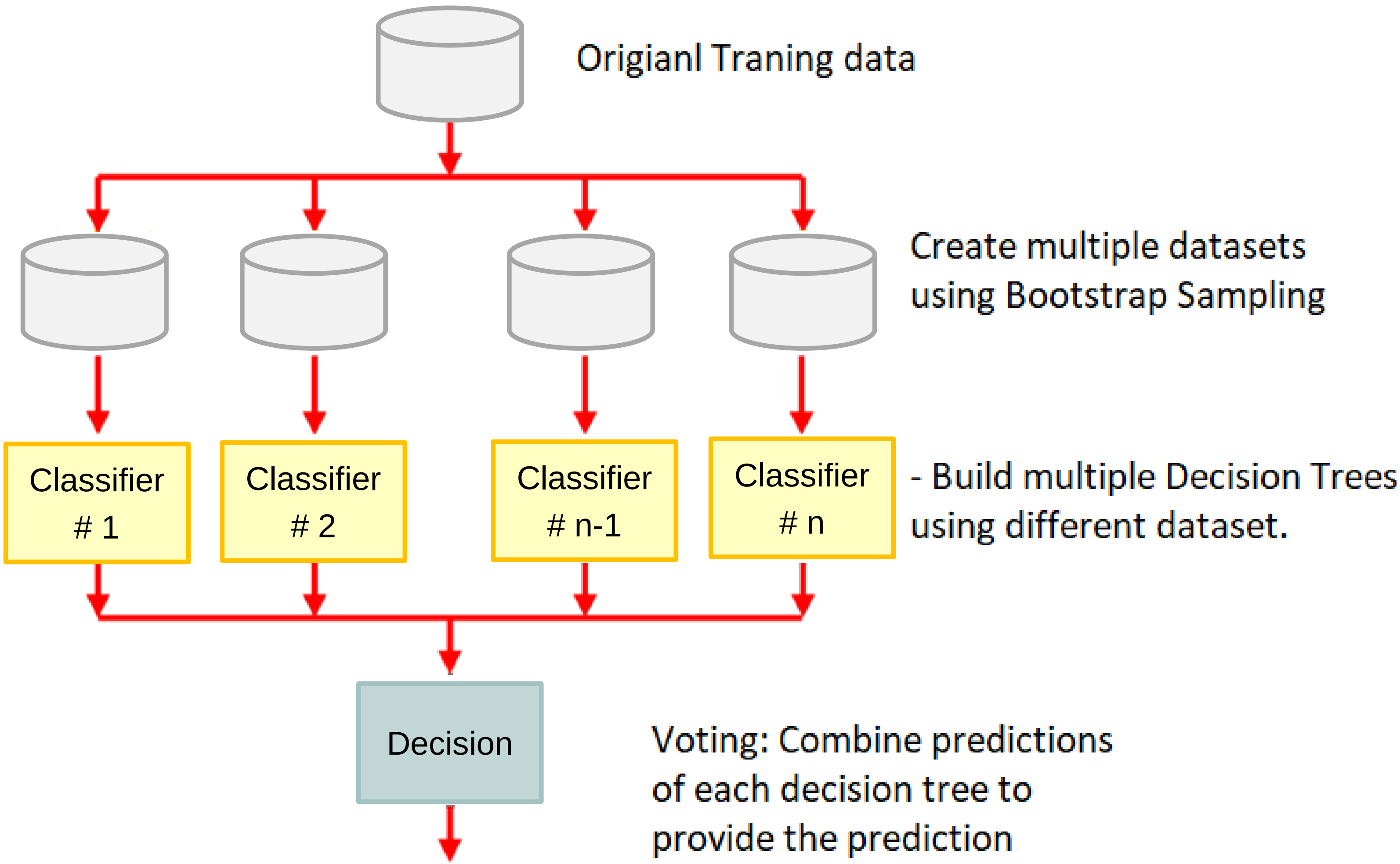
Dataset1	
Row no	
4	....
2	....
9	....
5	....
2	....
10	....
3	....
2	....
8	...
6	...

Dataset2	
Row no	
1	....
3	....
8	....
4	....
10	....
2	....
3	....
10	....
9	...
3	...

Dataset2	
Row no	
9	....
1	....
8	....
3	....
9	....
2	....
1	....
5	....
2	...
1	...

- Some rows will appear more than once in a dataset while some rows will be missing in a dataset

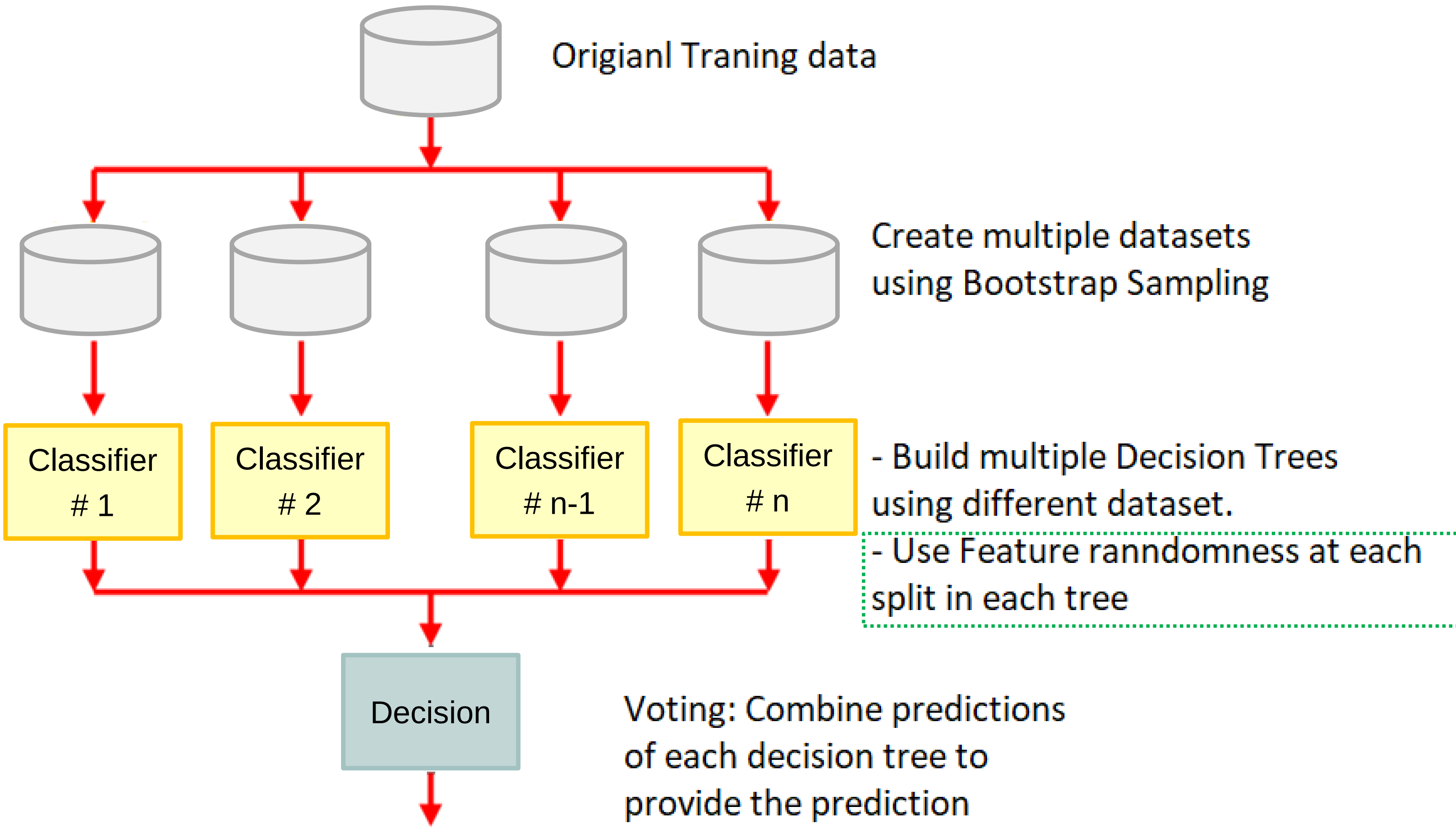
Bagging


Origianl Traning data
Create multiple datasets using Bootstrap Sampling
Classifier # 1
Classifier # 2
Classifier # n-1
Classifier # n
- Build multiple Decision Trees using different dataset.
Decision
Voting: Combine predictions of each decision tree to provide the prediction

Random Forest

- Build each tree using a sample drawn with replacement (bootstrap) from the training set
- When splitting a node during the construction of a tree, the split that is chosen is no longer the best split among all the features
- Instead, the split is picked is the best split among a random subset (say, k number of subsets) of the features
- As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree)
- Due to averaging, its variance decreases, usually more than compensating the increase in bias, hence yielding overall a better result

Random Forest



Random Forest

- Advantages
  - The process of averaging or combining the results of different decision trees helps to overcome the problem of overfitting
  - Less variance than a single decision tree
  - Higher accuracy than a single decision tree
- Disadvantages
  - The advantage of Decision Tree, interpretability, is lost. No interpretability
  - May require very large number of trees resulting in very slow training of model

Exercise

- Ensemble Techniques

Machine Learning

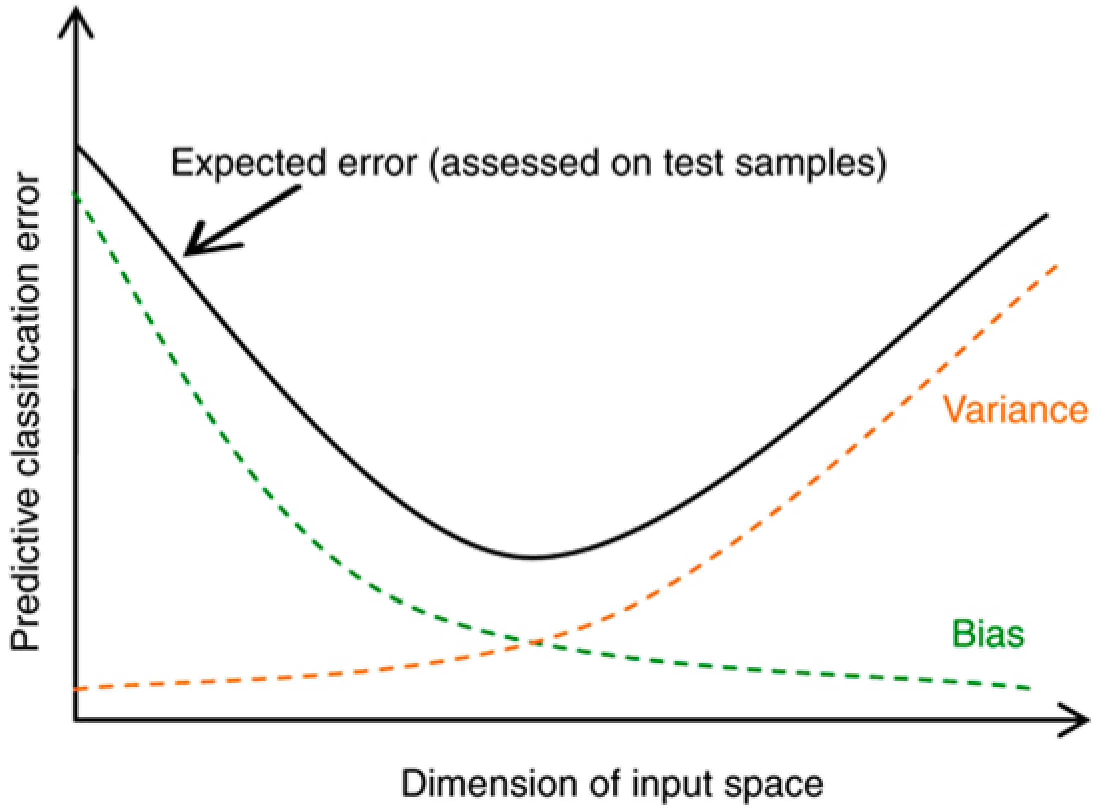
Bias Variance

Bias Variance Errors

- Bias
  - Caused by our selection of the attributes and our interpretation of their influence on each other
  - The real model in the universe / population may have many more attributes and the attributes interacting in different ways not reflected in our model

- Variance
  - Different test data – gives very different scores. Variance is the amount that the estimate of the target function will change if different training data was used
  - Caused by overfitting of model

Bias – Variance trade-off



- Select the right complexity model to trade=off Bias and Variance errors

Thank you

- Prashant Koparkar