Random Forests

Prof. G Panda, FNAE,FNASc,FIET(UK)
IIT Bhubaneswar

Outline

- Introduction
- What is Random Forest
- Why Random Forest?
- How Random Forest Works
- Example
- Advantages and Disadvantages.
- Applications

Introduction

Random Forests is the supervised machine learning algorithm used for both Regression and Classification.

Classification : It is problem of identifying to which set of categories a new observation belongs to.

Regression : Establishing relation between dependent variable(target value) and independent variable(s)(input variables or predictor variables)

What is Random Forest

- As the name suggests, this algorithm creates forest of trees by randomly selecting decision trees.
- Forest which means collection of trees(here Decision trees) and these trees are trained on subsets(equal to the size of training set) selected at random.
- The decision of majority of trees is chosen as the final decision.

What is Random Forest

- Most powerful Supervised Machine learning algorithm.
- Random Forest is an ensemble classifier made using many decision tree models.
- Ensemble models combine results from different models.
- Ensemble is like a divide and conquer approach used to improve performance where a group of weak learners can form a strong learner.
- In general the more trees in the forest the more robust the prediction and thus higher accuracy.

Why Random Forest?

- No Overfitting(use of multiple trees reduce the risk of overfitting)
- High accuracy(Runs efficiently on large database)
- Estimates missing data(maintain accuracy even if large proportion of data is missing).

Example:If A want to watch a movie based on the review of it.

Ask best friend (Analogous to decision tree): A get review from the friend who might me more biased towards the movie.

Ask all friends(Analogous to Random Forest) : A gets reviews from all friends and summarizes from them to watch or not. This is more generalized.

How does it work

- Bagging method is used to build the forest which is the collection of Decision trees.
- Multiple trees are built, to classify a new object based on attributes, each tree gives a classification and the class which has higher votes is chosen for the final classification,
- In case of regression average of all outputs of different is considered.

How does it work

Bagging(Bootstrap aggregating)

- Bootstrapping the dataset and making the decision from aggregation is called bagging.
- It Is a machine learning ensemble technique designed to improve the stability, accuracy, reduce variance(helps to avoid overfitting)
- **Bootstrapping**: It is an estimation method used to make predictions on a dataset by re-sampling it.

How does it work

- To create a bootstrapped data set, we must randomly select samples from the original data set. A point to note here is that we can select the same sample more than once.
- **Out of bag dataset** : entries that did not make into the bootstrap dataset
- Then run the out of bag sample through all the other trees which were built without it .
- We can measure how accurate our random forest is by the proportion of random forests that were correctly classified by the random forest.
- The proportion of out-of-bag samples incorrectly classified is “out-of bag error”

How does it work

- If there are M variables(features) in the data set, at each node in the decision tree only m(<< M) variables are considered to choose the best split.
- If all predictor variables are selected each decision tree will be same and the model will not learn something new.
- If randomly m features are chosen every time we get new decision tree, and classification will be more intelligent and generalized

How does it work

To find the optimal value of m:

Find out-of -bag error for different number of variables(different values of m)and choose the one with the most accurate random forest .

Steps:

1)Build the Random forest.

2)Estimate the accuracy.

3)repeat 1 and 2 till the optimal number of variables found.

How does it work

Grow each tree on an independent bootstrap sample from the training data. At each node:

1. Select m variables at random out of all M possible variables (independently for each node).

2. Find the best split on the selected m variables. Grow the trees to maximum depth (classification)

Random Forest:Training Phase

Step-1: create a bootstrapped dataset

To create a bootstrapped data set, we must randomly select samples from the original data set. A point to note here is that we can select the same sample more than once.

Step-2 : create Decision tree

This can be done using CART or ID3 algorithms which uses Gini or Entropy as measure of impurity.

Step-3: Repeat step 1 and 2 to built forest of trees.

Algorithm

1. Assume number of cases in training set is N. Then sample of these N cases is taken at random but with replacement.
2. If M is the number of input variables or features, a number m<<M is specified such that at each node, m variables are selected at random out of M. The value of m is held constant while the forest is grown.
3. The best split on these m variables is used to split the node.
4. Each tree is grown to the largest extent possible by repeating steps 2 and 3 .
5. Build forest of n trees by repeating steps 1 , 2 and 3.
6. Predict new data by aggregating the predictions of n trees

Example

Consider the dataset with four predictor variables Blood Flow, Blocked Arteries, Chest pain and weight.

Blood Flow	Blocked Arteries	Chest pain	Weight	Heart Disease
Abnormal	No	No	130	No
Normal	Yes	Yes	195	Yes
Normal	No	Yes	218	No
Abnormal	Yes	Yes	180	Yes

Example: Training phase

Step-1: Create a Bootstrapped dataset.(where rows are selected at random)

Bootstrapped dataset-1

Blood Flow	Blocked Arteries	Chest pain	Weight (in kg)	Heart Disease
Normal	Yes	Yes	195	Yes
Abnormal	No	No	130	No
Abnormal	Yes	Yes	180	Yes
Abnormal	Yes	Yes	180	Yes

Example: Training phase

Step-2 : Creating decision tree

Since number of features M = 4, let m = 2(features chosen at each node randomly)

To find that variable with best split, Entropy measure can be used.

$$\boxed{Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)}$$

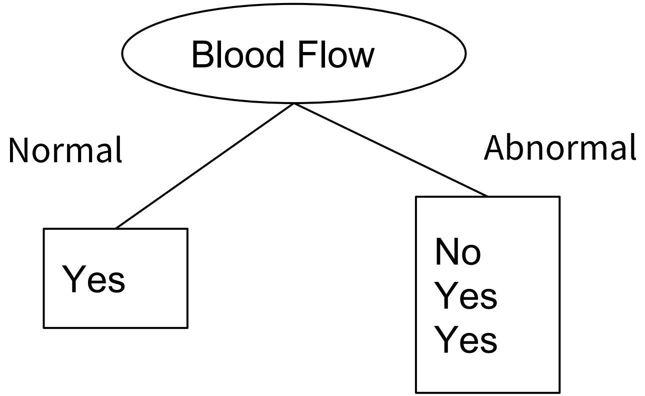
$$\boxed{H(p) = \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)}$$

Where, H(p) is uncertainty at node p and k is number of partitions, n is total records in node p and n_i = records in partition i.

Smaller the value of H(p) lesser is the uncertainty.

Example: Training phase

Let's say Blood Flow and Blocked arteries are two randomly selected variables.



Entropy(Blood Flow = Normal)

$= -1x\log(1) - 0x\log(0) = 0$

Entropy(Blood Flow = Abnormal)

$= -\frac{1}{3}\log(\frac{1}{3}) - \frac{2}{3}\log(\frac{2}{3}) = 0.918$

H(Blood Flow) = ¼xEntropy(Blood Flow = Normal) + ¾xEntropy(Blood Flow = Abnormal)

= 0.689

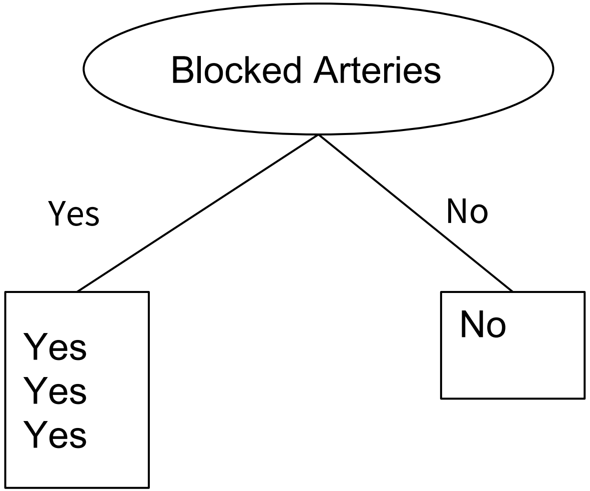
Example: Training phase

Entropy(Blocked Arteries = Yes)

= -3/3xlog(3/3) - 0xlog(0) = 0

Entropy(Blocked Arteries = No)

= -1xlog(1) - 0xlog(1)

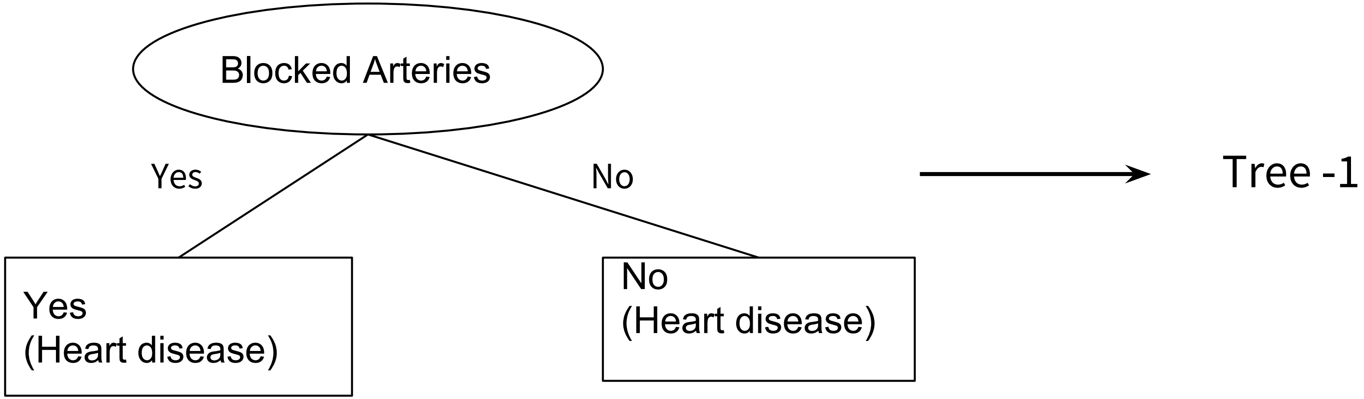


H(Blocked Arteries) = ¾ xEntropy(Blocked Arteries = Yes) + ¼ Entropy(Blocked Arteries = No)

= 0

Example: Training phase

SInce H(p) is less for Blocked arteries choose it as root node.



Since this decision tree gives the final classification, this is the final decision tree for Bootstrap dataset-1

Example: Training phase

Step-3: repeat step 1 and 2 to build forest of trees.

Bootstrapped dataset-2

Blood Flow	Blocked Arteries	Chest pain	Weight (in kg)	Heart Disease
Normal	Yes	Yes	195	Yes
Normal	Yes	Yes	195	Yes
Normal	No	Yes	218	No
Abnormal	Yes	Yes	180	Yes

Example: Training phase

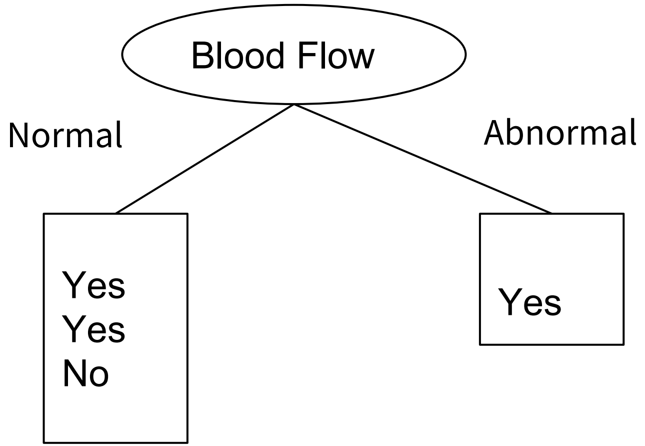
Let Blood Flow and chest pain be two randomly selected variables

Entropy(Blood Flow = Normal)

= -⅔ xlog(⅔) - ⅓ xlog(⅓) = 0.918

Entropy(Blood Flow = Abnormal)

= -1xlog(1) - 0xlog(0) = 0



H(Blood Flow) = ¾ xEntropy(Blood Flow = Normal) +¼ Entropy(Blood Flow = Abnormal)

= 0.689

Example: Training phase

Let Blood Flow and chest pain be two randomly selected variables

Entropy(Chest Pain = Yes)

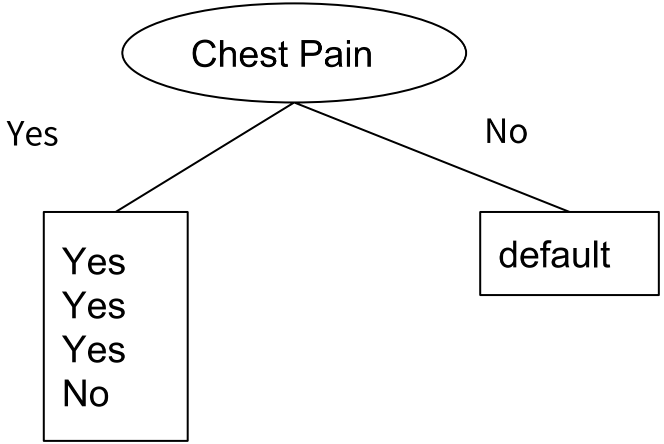
= -¾ xlog(¾) - ¼ xlog(¼) = 0.811

Entropy(Chest Pain = No)

= -0xlog(0) - 0xlog(0) = 0

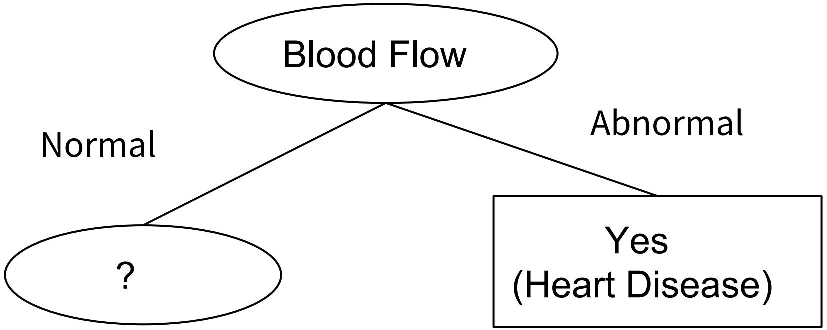
H(Chest Pain) = 4/4 xEntropy(Chest Pain = Yes) + 0/4 x Entropy(Chest Pain = No)

= 0.811



Example: Training phase

Since H(s) of Blood Flow is less, it is chosen as root



Now at node Blood Flow = Normal, we need to choose two random variables ,

Let chest pain and weight be those two variables.

Example: Training phase

Given Blood Flow = Normal

Entropy(Chest Pain = Yes)

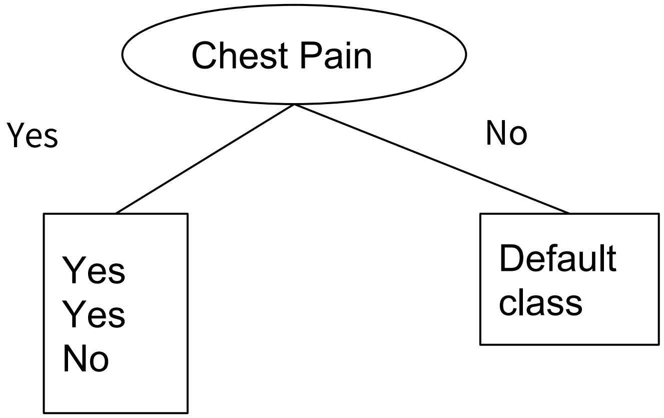
$= -\frac{2}{3} \text{xlog}(\frac{2}{3}) - \frac{1}{3} \text{xlog}(\frac{1}{3}) = 0.918$

Entropy(Chest Pain = No)

$= -0\text{xlog}(0) - 0\text{xlog}(0) = 0$

H(Chest Pain) = 3/3 xEntropy(Chest Pain = Yes) + 0/3 x Entropy(Chest Pain = No)

= 0.918



Example: Training phase

Given Blood Flow = Normal

Entropy(Weight < 200)

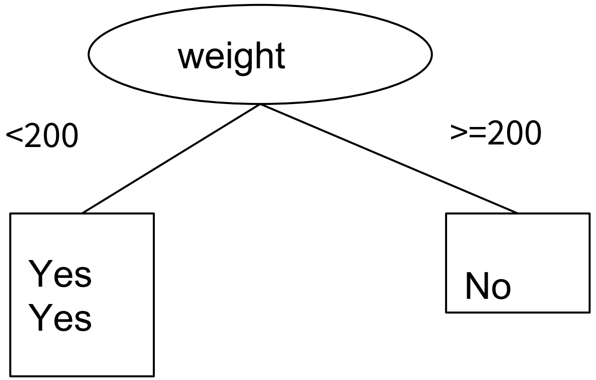
= -2/2 xlog(2/2) - 0 xlog(0) = 0

Entropy(Weight >= 200)

= -1/1xlog(1/1) - 0xlog(0) = 0

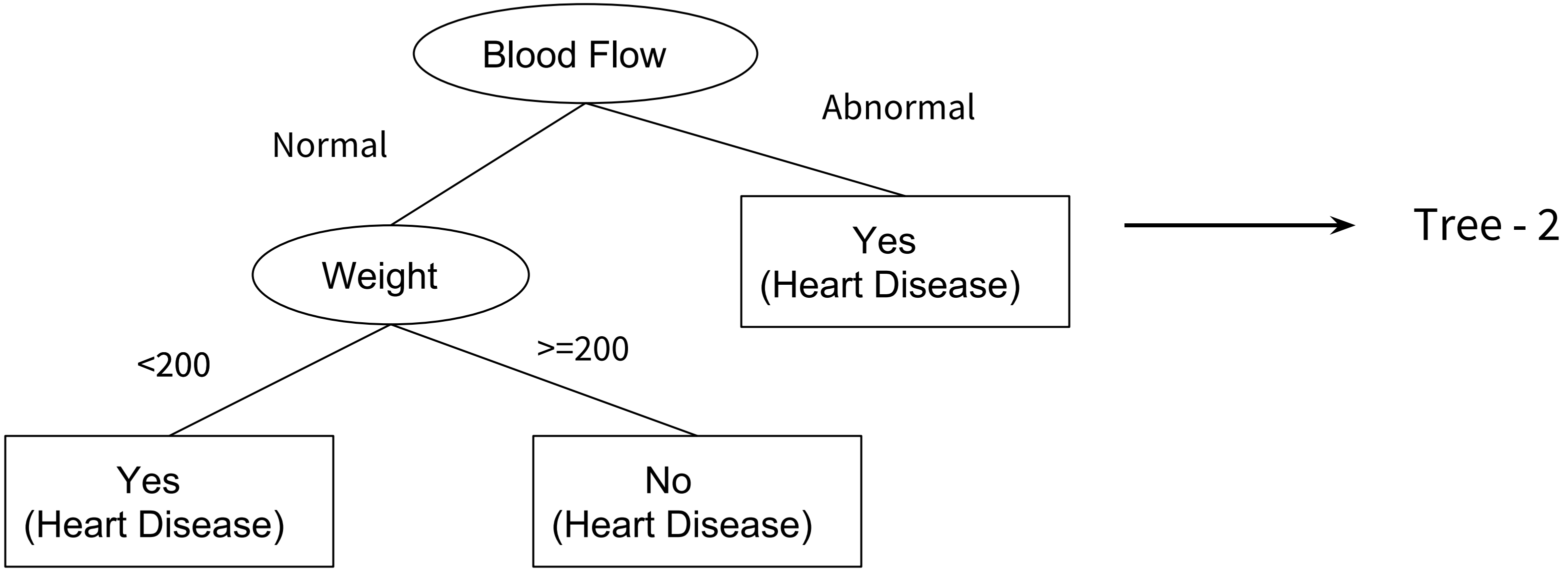
H(Weight) =2/3 xEntropy(Weight < 200) +1/3 Entropy(Weight >= 200)

= 0



Example: Training phase

Since H(s) of Weight is less, it is chosen as child of Blood Flow = Normal



Example: Training phase

Step-3: repeat step 1 and 2 to build forest of trees.

Bootstrapped dataset-3

Blood Flow	Blocked Arteries	Chest pain	Weight (in kg)	Heart Disease
Abnormal	No	No	130	No
Normal	Yes	Yes	195	Yes
Normal	No	Yes	218	No
Normal	No	Yes	218	No

Example: Training phase

Let weight and chest pain be two randomly selected variables

Entropy(weight<200)

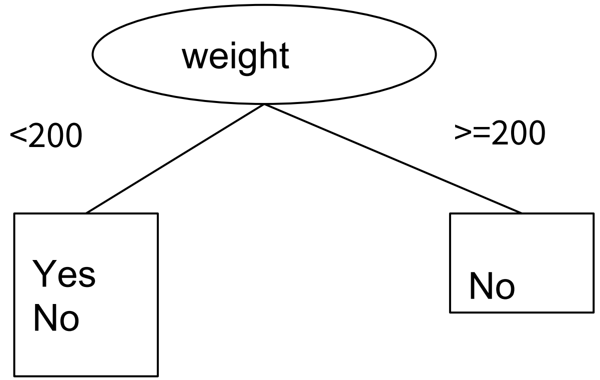
= -½ xlog(½) - ½ xlog(½) = 1

Entropy(weight >= 200)

= -1/1 xlog(1/1) - 0 xlog(0) = 0

H(weight) =2/3 xEntropy(Weight<200) +1/3 Entropy(weight>=200)

= 0.666



Example: Training phase

Entropy(Chest Pain = Yes)

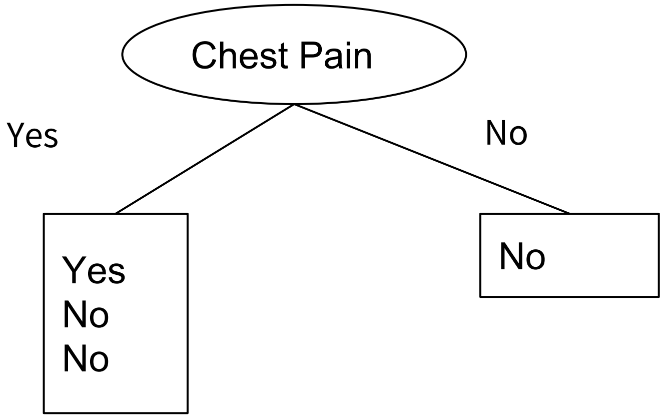
= -⅔ xlog(⅔) - ⅓ xlog(⅓) = 0.918

Entropy(Chest Pain = No)

= -0xlog(0) - 1/1xlog(1/1) = 0

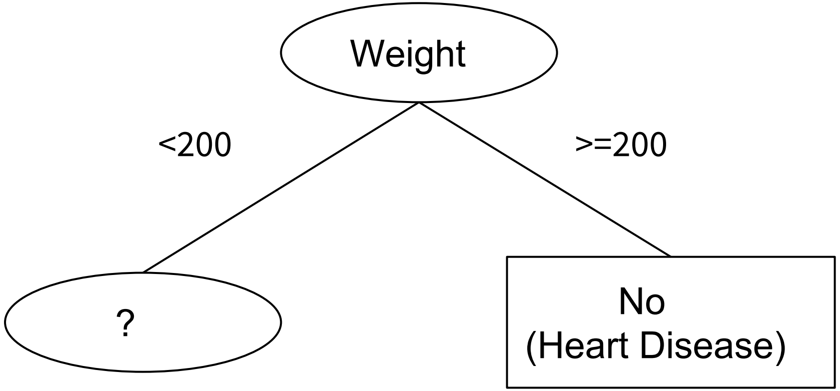
H(Chest Pain) = 3/4 xEntropy(Chest Pain = Yes) + 1/4 x Entropy(Chest Pain = No)

= 0.689



Example: Training phase

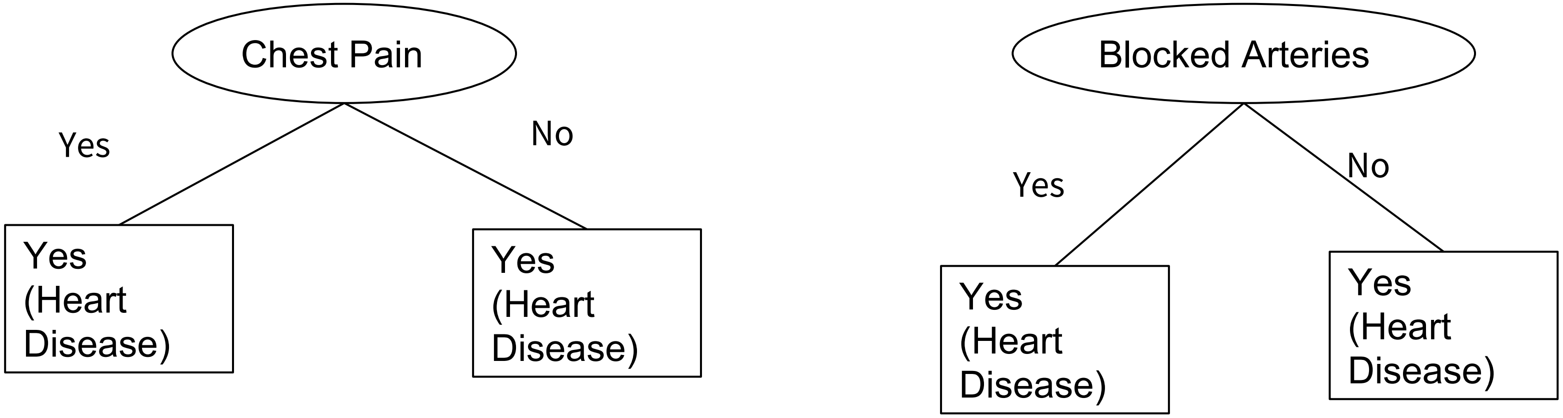
Since H(s) of Weight is less, it is chosen as child root node



At node weight < 200 again select two random variables and repeat step 1 and step 2

Let the two variables be Blocked Arteries and chest pain

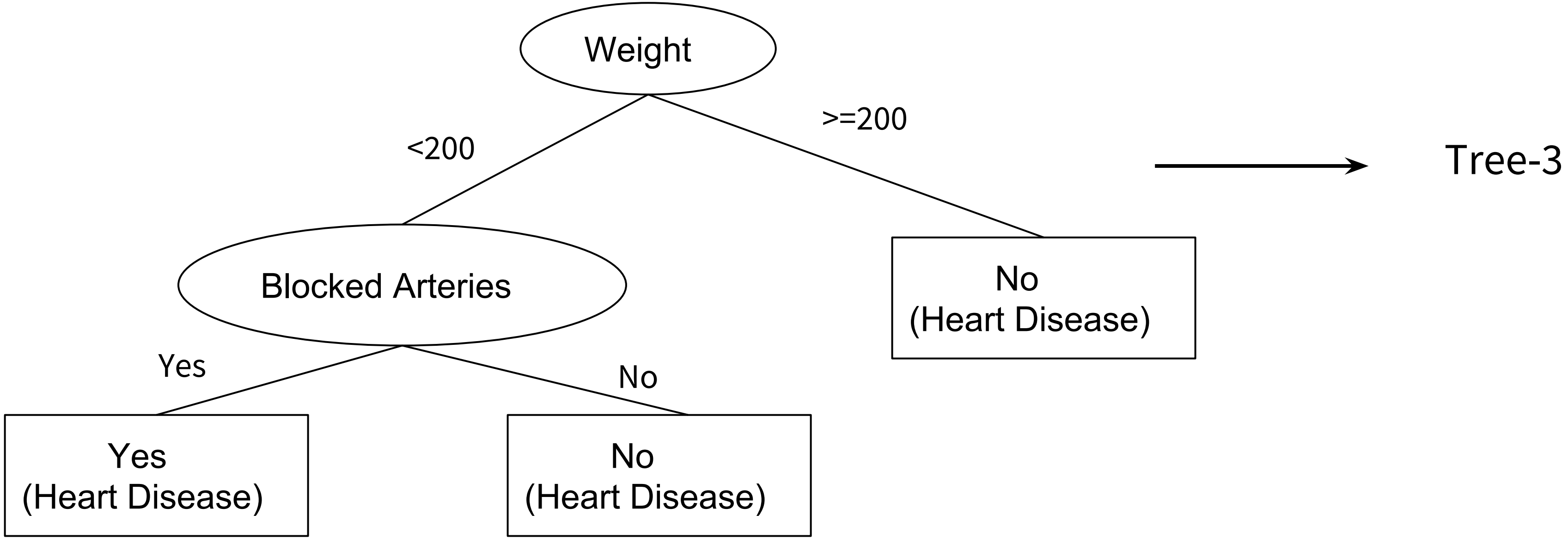
Example: Training phase



Since both variable reach to leaf nodes, we can consider anyone of them say Blocked Arteries.

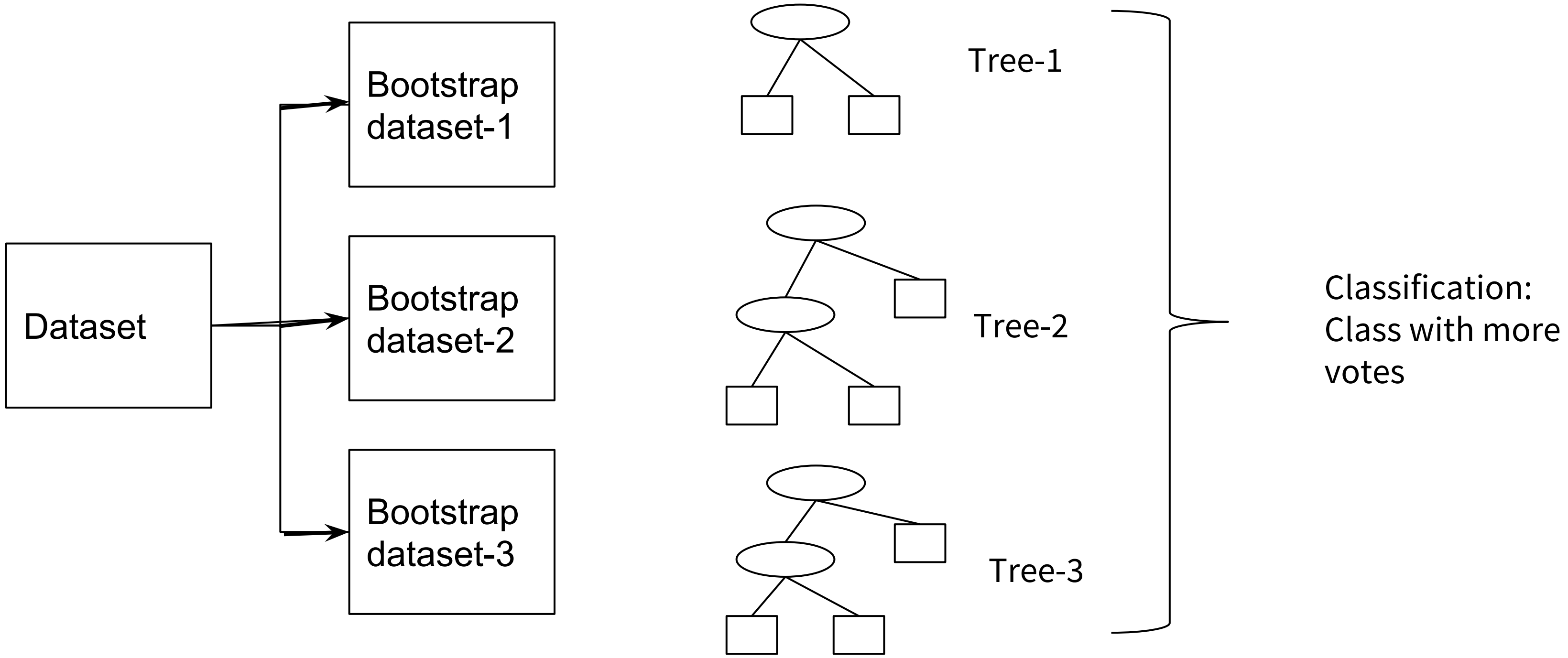
Example: Training phase

So Blocked Arteries is chosen as child node of Weight < 200



Example: Training phase

Considering the forest with previously computed 3 trees.



Example Testing Phase

Given a person with Abnormal Blood Flow, No Blocked Arteries, No chest pain and weight = 120kg , Does the person have heart Disease or not.

Tree-1 gives class ‘No’

Tree-2 gives class ‘Yes’

Tree-3 gives class ‘No’

Since more votes are for ‘No’, Random forest classifies the person as ‘No Heart Disease’.

Advantages

- Most accurate learning algorithm
- Used for both both Classification and regression problems
- Runs efficiently on large databases.
- Can be easily grown in parallel.
- Handles missing data and maintains accuracy for missing data.
- Wont overfit the model(fit data so close to what we have in the sample)
- Handle large dataset with higher dimensionality.

Disadvantages

- Good at classification but not as good as for regression(cannot predict beyond the range)
- Little control on what the model does.

Applications

- In banking sector : Loyal/ fraud customer
- Medicine: Identify disease by analysing patient records
- In stock market used to identify stock behaviour(predict expected profit or loss by purchasing a particular stock)

References

- Decision Trees and Random Forests: A Visual Introduction for Beginners, 4 October 2017, Book by Chris Smith and Mark Koning
- Machine Learning: The Ultimate Beginners Guide for Neural Networks, …,29 July 2017,Book by Ryan Roberts
- Random Forest or Random Decision Forest edureka

Thank you