Random Forests

(Random decision forests)

Random forests is an ensemble method that builds multiple decision trees, and predicts outcome as the class predicted by the most trees.

Random forests are trained using "bagging" method (bootstrap aggregation) that combines multiple learning models that improve the overall result.

Eg:

Ram would like to go on a tour, so he asks his friends for suggestions. He also provides them with the likes and dislikes of previous travels.

- Each friend creates rules (decision tree) to guide his recommendation for Ram's travel.

Finally, Ram chooses the places that most his friends recommend him.

Dataset

Chest Pain	Good blood circulation	Blocked arteries	weight	Heart disease
NO	no	no	125	no
yes	yes	yes	180	yes
yes	yes	no	210	no
yes	no	yes	167	yes

Random Forest Process

(1) create a bootstrap sample
(a bootstrap sample is same size as the original sample, but we randomly select samples from the original dataset)

chest pain	good blood circulation	blocked arteries	weight	heart disease
yes	yes	yes	180	yes
yes	yes	no	210	no

(2) create a random subset of features at each step and build a decision tree for every such feature subset

Bootstrapped dataset

chest pain	good blood circulation	blocked arteries	weight	heart disease
yes	yes	yes	180	yes
yes	yes	no	210	no

DT1

cp	gbc	hd
yes	yes	yes
yes	yes	no

DT2

ba	weight	hd
yes	180	yes
no	210	no

DT3

cp	ba	hd
yes	yes	yes
yes	no	no

(3) Repeat steps 1 and 2 to build more decision trees

Note Using a bootstrapped sample and considering only a subset of variables result in a wide variety of trees.

This variety makes random forests more effective than individual decision trees.

classification

Unseen sample

(chestpain = yes, good blood circulation = no, blocked arteries = no, weight = 168, heartdisease = ?)

① Run down the unseen samples across all the decision trees.

② Accumulate the votes for each prediction. The class label receiving most votes will be the prediction of the random forest.

Hyperparameters in Random Forest

- Number of decision trees in the random forest
- ~~Number~~ method for bootstrapping (sampling with/without replacement)
- no. of features in each tree
- max. allowed depth for each tree
- min. no. of samples in a leaf node
- min. no. of samples required to split a node

(RandomizedSearchCV for finding best values for hyperparameters in sklearn)

Advantages

1, Scalability
2, works for both classification and regression problems
3, works on continuous and categorical variables
4, no feature scaling required
5, no pruning required
6, handles non-linear parameters effectively

disadvantages

1, high complexity
2, more training time

Applications

- credit card fraud detection
- customer Segmentation
- Identification of loan defaulters
- cancer Prediction
- Sentiment Analysis
- Product Recommendation

Extra Reading

- Ensemble learning
  - Bagging
  - Boosting
    - AdaBoost