19-Oct-2023

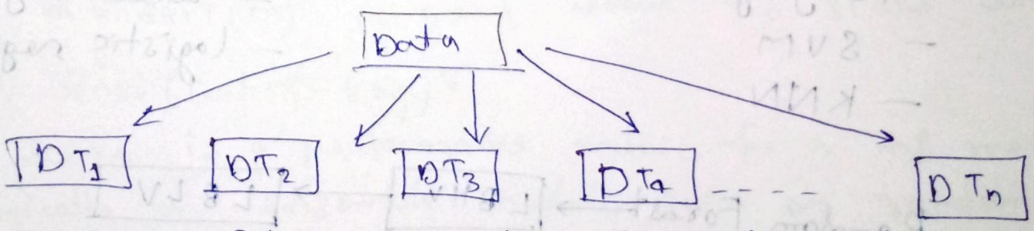
RANDOM FOREST

- Random Forest is the apply for both classification and regression.
- Random Forest is the ensemble technique.
- it is a part of bagging techniques.

Random Forest
↑ Group of tree

Bagging
↙ Bootstrap (sampling)
↘ Aggregation

- If in the Bagging Technique all the model are decision tree then this technique is called Random Forest.



where DTn = Decision Trees (model)

- for each model we provide a subset of data of data (original)
- if assume if original data has 10000 raw then we provide each decision tree to 500 raw.
- these 500 raw is decided by these method

(i) Raw Sampling (Pasting) } with or without replacement
(ii) column sampling (Random subspaces) }
(iii) raw and column sampling (Random patches)
(iv)

Bias Variance Trade-Off :

The error comes in machine learning due to two reasons. (i) Bias
(ii) Variance

- **Bias** → when ML model not provide better result in training data.
- **Variance** → when ML model is perfectly work on the training data (~~Hindi~~ रटता है) but not good perform on test data.

Hence we always need low bias and low variance. ~~or~~ but

$$\text{Bias} \propto \frac{1}{\text{variance}}$$

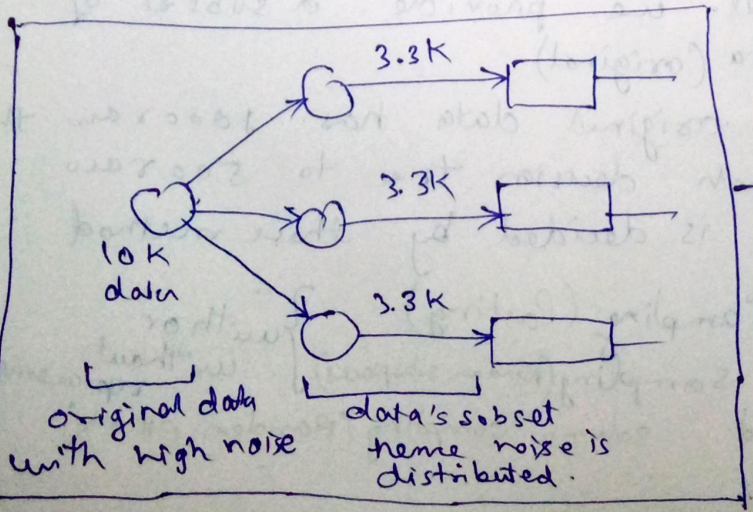
Algorithms types

Low bias high variance
- Fully grown DT
- SVM
- KNN

High bias low variance
- linear regression
- logistic regression

Random Forest → LBHV → LBLV

example How?



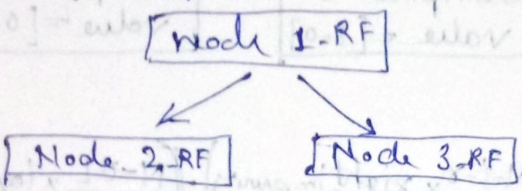
→ Hence Overall model is Low Bias Low variance

Difference between the Bagging and ~~boos~~ Random Forests.

Fundamental difference between bagging and the random forest is that in Random forests, only a subset of features is selected randomly out of the entire and therefore the best split feature from the subset is employed to separate each node during a tree, unlike in bagging where all features are consider for splitting a node

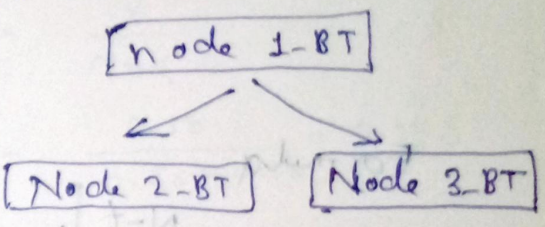
Random forests.

only $m < p$ features considered for each node for split



Bagging Trees

All of p features considered for each node for a split



where m can be selected via out-of-bag error, but $m = \sqrt{p}$ is good value to start with

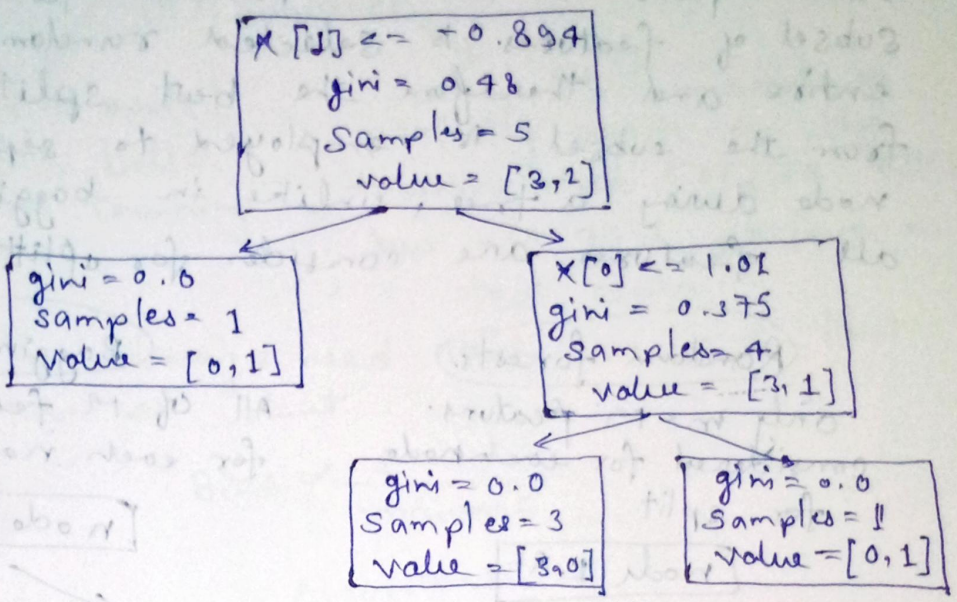
OOB Score (out-of-bag):

OOB score is a performance metric for a ml model, specifically for ensemble models such as random forest. it is calculated using the samples that are not used in the training of the model, which is called out-of-bag samples. These samples are used to provide an unbiased estimate of the model's performance, which is known as a the OOB score

Feature Importance Calculation

(i) For Decision Tree

the addition of all features is always one

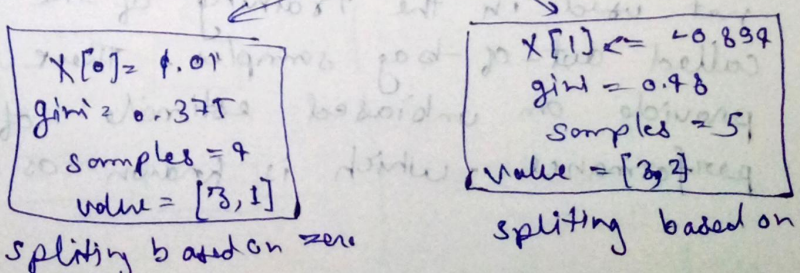


Formula

$$n_i = \frac{N-t}{N}\left[\text{impurity} - \left(\frac{N-t-r}{N-t} \times \text{right\_impurity}\right) - \left(\frac{N-t-L}{N-t} \times \text{left\_impurity}\right)\right]$$

$$f_{ik} = \frac{\sum_{j \in \text{node splitting feature } k} n_i}{\sum_{j \in \text{all nodes}} n_i}$$

In the above tree, column 0 and 1



$$n_i(0^{th}) = \frac{X}{X+Y}$$

$$n_i(1^{th}) = \frac{Y}{X+Y}$$

$$n_i = \frac{N-t}{N}\left[\text{impurity} - \left(\frac{N-t-r}{N-t} \times \text{right\_impurity}\right) - \left(\frac{N-t-l}{N-t} \times \text{left\_impurity}\right)\right]$$

$$Y = \frac{5}{5}\left[0.48 - \left(\frac{4}{5} \times 0.375\right) - \frac{1}{5} \times 0\right] = 0.18$$

$$X = \frac{4}{5}\left[0.375 - (0) - 0\right] = 0.30$$

feature importance of 0^{th} feature $[0^{th}] = \frac{X}{X+Y}$

$$= \frac{0.30}{0.30+0.18} = 0.625$$

feature importance of 1^{st} feature $[1^{st}] = \frac{Y}{Y+X}$

$$= \frac{0.18}{0.30+0.18} = 0.375$$

(ii) For Random Forest.

```
print(rf.estimators_[0].feature_importances_)
print(rf.estimators_[1].feature_importances_)
```

output. [1. 0.]

[0.5555 0.4444]

feature importance of 1st column

print((1+0.5555)/2) → 0.777500.

feature importance of 2nd column

print((0 + 0.4444)/2)

warning: impurity-based feature importances can be misleading for high cardinality features (many unique values) ~~See~~ for alternative we use

sklearn.inspection.permutation_importance

1. What is an OOB error (Out-Of-Bag error) and how is it useful?

- OOB error is a metric used in bagging algorithm like RF. it calculates the prediction error for each data point based on the votes of the decision tree in the ensemble that did not include that data point in their bootstrap sample during training.
- The OOB error serves as a reliable estimate of the model's performance without the need for a seperate validation set, making it useful to assessing model accuracy and preventing overfitting.
- for example in RF with 100 DT, the OOB error is calculated as the average prediction error accross all data points. While OOB error simplifies the modeling process and provides a robust estimate, it can be computationally intensive with a large number of trees and may have some variability due to its estimation nature.

2. In what scenario DT should be preferred over random forest?

(i) When to Prefer:

DT: When interpretability is crucial, and you need a single, understandable tree.

RF: When you seek higher predictive accuracy and robustness to outliers or noisy data.

(ii) Example use Case:

DT (Medical Diagnosis): In a scenario where medical practitioners need clear, explainable rule for diagnosing a common illness based on a small dataset of patient symptoms.

RF(churn prediction): when predicting customer churn in a telecom company with a large dataset of diverse features, aiming for improved accuracy.

(iii) Description:

DT: it provides a clear, understandable decision path, crucial in scenarios where interpretability is more important than marginal gains in accuracy.

RF: Combines multiple decision tree, reducing overfitting and improving generalization performance, making it suitable for complex, high-dimensional data.

(iv) Advantages:

DT: Easy to visualize and explain; work well with small to medium-sized datasets.

RF: Reduces overfitting through ensemble learning; captures complex relationship in data

(v) Disadvantages:

DT: Prone to overfitting on large datasets or complex data.

RF: May not provide a transparent, interpretable model.

(vi) Numerical example:

DT (Medical Diagnosis): consider a small dataset of patient symptoms for diagnosing a common illness. A single DT can provide a clear set of rules that a medical practitioner can follow for diagnosis.

RF (customer churn Prediction):
In a large telecom dataset with hundreds of features, a random forest can combine multiple DT to predict customer churn accurately, considering various factors like call duration, contract length and customer demographics.