Random Forest

Random Forest

- Algorithm used for both Classification and Regression
  - ❑ Works best for *Classification*
- Also known as an "ensemble classifier" (many models working together)
- Multiple Decision Trees are grown by taking different attributes of the same dataset; and an average is taken. For classification, voting is done (using ***mode***)
- Overcomes the problem of overfitting of a single tree and generalises well
- Trees not pruned
- Uses bootstrap method to select samples
- Easily affected by correlation; not much by outliers
- Algorithm has parameters to finetune the final model – eg: ***mtry*** (number of variables used), ***mtree*** (number of decision trees to build) etc.
- Important variables can be selected

Bootstrapping

- Bagging
- A random sampling with replacement
- Samples are also called **Out of Bag** samples
- Models are built in parallel

- **Steps**
  1. Randomly select an observation/record from the original dataset
  2. Replace the observation back in the dataset
  3. Repeat process (1 and 2) **'n'** times

     n → Total number of records in the original dataset / sample

- Final bootstrap sample with 'k' samples

Out of Bag Error

- Error estimated on these out of bag samples is known as **out of bag error**
- Study of error estimates by OOB is as accurate as a test set (of the same size as the training set)

Advantages of Random Forest

- High accuracy
- Effectively used on very large datasets
- Gives estimates of important variables
- Maintains a good accuracy in case of missing data

- **DT is better than RF when the variables are binary in nature**

Tree building process

- Let ***N*** be the total number of records and ***M*** be the full set of features / variables

- Takes a random set of data ***(n)*** with replacement. This is the training set for the tree
- Takes a random set of features ***(m)*** ***[m < M]***
  - ❑ For each iteration, ***m*** is fixed

- Study the performance of this tree

- A forest of DT's are built in this process

- Error rates of each tree is calculated

- Joint set of variables are determined that give the strongest model

Process (of building a tree)

N=1000, M=30, n=500, m=3

N → Total number of records
n → Random set of data with replacement. (This is the training set for the tree)

M → Full set of features / variables
m → Selected features (number kept constant)

First Tree

- Random sample data ***S1 (n)***
- Using ***S1***, build a "random" tree
- Take ***m*** features from ***M*** eg: ***m2, m16, m29***
- Identify the best attribute – eg: ***m16*** → root node
- Split ***S1*** on ***m16*** – gives 2 new subsets eg: ***S1_a*** and ***S1_b***

- For ***S1_a***, select ***m*** eg: ***m1,m5,m10***
  - Identify the best attribute – eg: ***m1***
  - Split ***S1_a*** into ***S1_a_1***, ***S1_a_2***
- For ***S1_b***, select ***m*** eg: ***m11,m15,m10***
  - Identify the best attribute – eg: ***m15***
  - Split ***S1_b*** into ***S1_b_1***, ***S1_b_2***
- etc…
- Tree over