Random Forest

Let's dive a bit deeper into how Random Forest works. A Random Forest is essentially nothing else but bagged decision trees, with a slightly modified splitting criteria. The algorithm works as follows:

1. Sample m data sets $D_1, \ldots, D_m$ from D with replacement.
2. For each D_i, train a full decision tree h_{D_i} (max-depth=∞) with one small modification: Before each split, randomly subsample $k \leq d$ features (without replacement so we get k unique features) and only consider these for your split. This further decreases the variance of the trees as different features will be considered on each split, thus adding some "regularization".
3. The final classifier is $\hat{h}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} h_{D_i}(\mathbf{x})$.

> ☆ **Key Points**
>
> Random Forests harness the power of bagging to reduce the variance of decision trees (which are very high-variance/low-bias classifiers).
>
> Random Forests introduce randomness in two stages: sampling the data with replacement and sampling a subset of features at each split.
>
> The final output of a Random Forest averages the predictions of all the component trees.

The Random Forest is one of the best, most popular, and simplest out-of-the-box classifiers. There are two reasons for this:

- Random Forests only have two hyperparameters: m and k. Forests are extremely *insensitive* to both of these. A good choice for k is $k = \sqrt{d}$, where d denotes the number of features. You can set m as large as you can afford.
- Decision trees do not require a lot of preprocessing. For example, the features can be of different scale, magnitude, or slope. This can be highly advantageous in scenarios with heterogeneous data; for example, the medical settings where features could be things like blood pressure, age, or gender, each of which is recorded in completely different units.

Useful Variants of Random Forest

- Split each training set into two partitions $D_l = D_l^A \cup D_l^B$, where $D_l^A \cap D_l^B = \emptyset$. Build the tree on D_l^A and estimate the leaf labels on D_l^B. You must stop splitting if a leaf has only a single point in D_l^B in it. This has the advantage that each tree and also the RF classifier become **consistent** (https://en.wikipedia.org/wiki/Consistency_(statistics)).
- Do not grow each tree to its full depth; instead, prune based on the leave-out samples. This can further improve your bias-variance tradeoff.
- Random Forests are also extremely useful to determine the importance of a particular feature. To this end, one can keep track of the accumulated reduction in loss that is achieved by splitting on a particular feature. Many variants of this approach are known and used in practice.