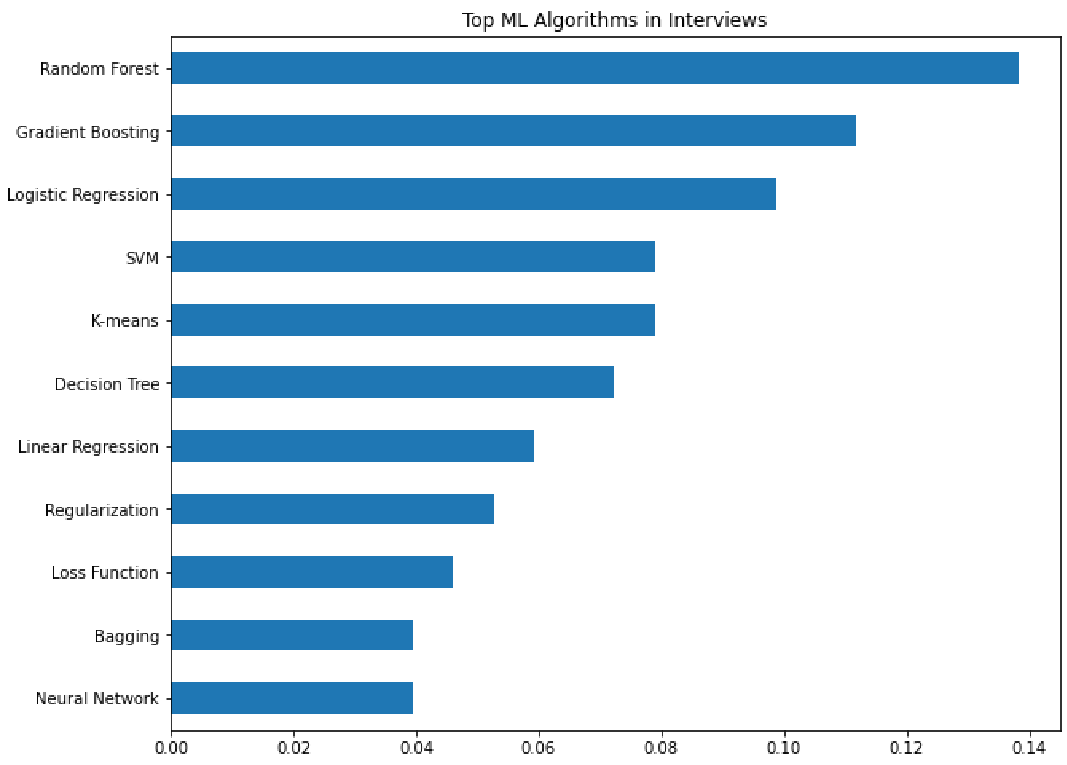
EMMA ding

emmading.com

info@datainterviewpro.com

Random Forest

📍 Random forest is the top ML algorithm asked in interviews.



Lesson Structure

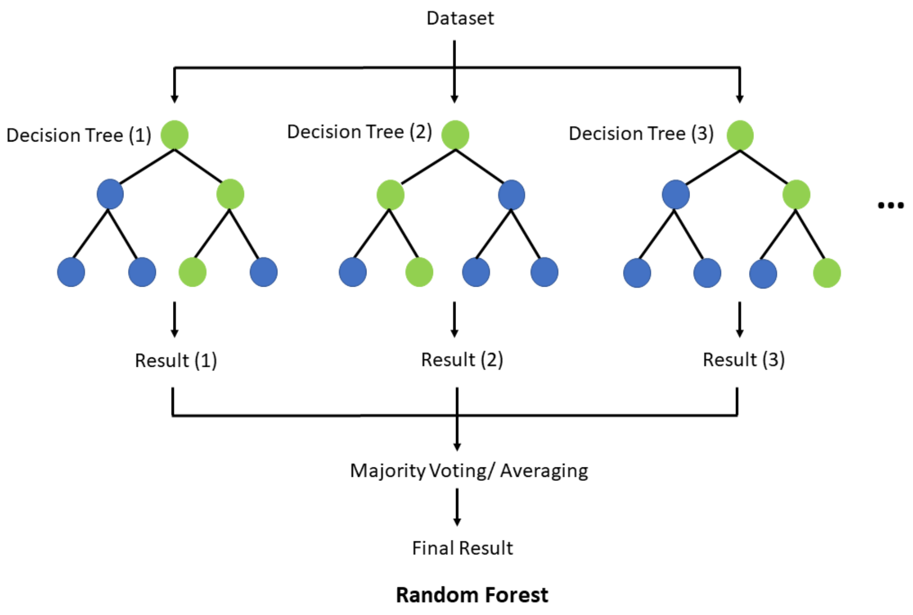
📌 Interview Questions

- How does random forest work? What are the ways in which they improve upon individual decision trees?
- Why is a random forest “random”?

- What are the hyperparameters of a random forest?
- Why can random forests help reduce variance?
- Pros and cons of a random forest.

Random Forest

Is an ensemble learning method for **classification** and **regression** tasks, that operates by constructing multiple decision trees (each trained on a **subset** of samples using a **subset** of features) at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

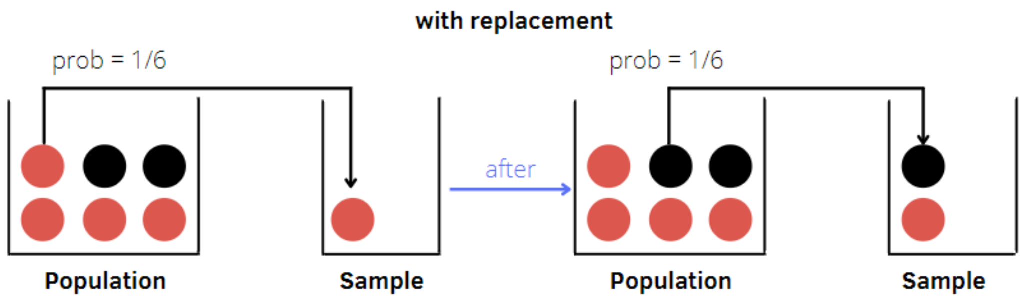


TseKiChun via Wikimedia Commons

How Random Forest works

▼ 4 steps

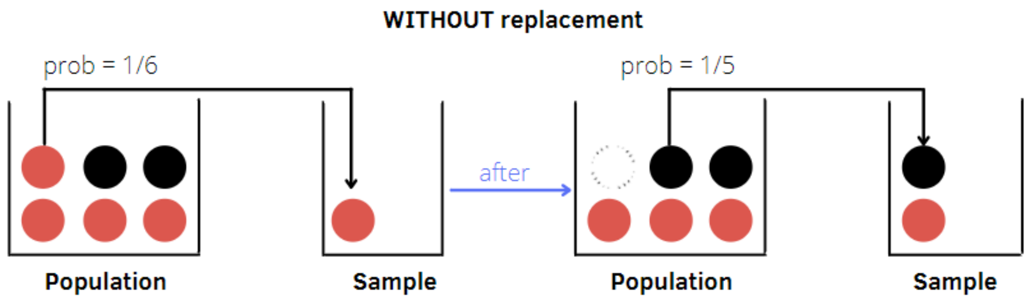
1. Draw a random bootstrap sample of size n (**randomly** choose n examples from the training dataset **with replacement**).



Credit: Eugenia Anello, Image source: https://towardsdatascience.com/an-introduction-to-probability-sampling-methods

2. Grow a decision tree from the bootstrap sample. At each node:

- **Randomly** select d features **without replacement**. e.g., if there are 20 features, choose a random five as candidates for constructing the best split.



Credit: Eugenia Anello, Image source: https://towardsdatascience.com/an-introduction-to-probability-sampling-methods

- Split the node using the feature that provides the best split according to the objective function, e.g., maximizing the information gain.

3. Repeat *steps 1-2* k times. Essentially, we will build k decision trees.

4. Aggregate the prediction by each tree to assign the class label by **majority vote** (classification) or take the **average** (regression).

📌 **Why random forest “random”?**

Random forest constructs a large number of trees with **random** bootstrap samples from the training data. As each tree is constructed, take a **random** subset of features before each node is split. Repeat this process for each node until the tree is large enough.

▼ **Random forest vs Bagging**

- Random forest is a modification of the bagging algorithm.
- Only difference: Step 2

▼ Hyperparameters

- A number of trees, k (`n_estimators`) in a random forest (step 3). The larger the number of trees, the better the performance of the random forest model at the expense of an increased computational cost.
- ▼ Less commonly used in practice
    - Size of the bootstrap sample n (`max_samples`) to consider for each tree (step 1). Typically, this is chosen to be equal to the **number of training samples** in the original training dataset.
    - A number of features d (`max_features`) to consider at each split (step 2).
        - For classification, the default is $d = \sqrt{m}$, where m is the number of features in the training dataset.
        - For regression, the default is $d = m/3$, where m is the number of features in the training dataset.

Variance reduction

▼ Average of i.d. random variables

An average of k i.i.d. random variables, each with variance σ^2, has variance $\frac{\sigma^2}{k}$.

If the variables are simply i.d. (identically distributed, but not necessarily independent) with positive pairwise correlation ρ, the variance of the average is (proof)

$$\rho\sigma^2 + \frac{1-\rho}{k}\sigma^2$$

As k increases, the second term disappears, but the first remains. The larger the correlation, the larger the variance of the average.

✏️ The size of the correlation of pairs of trees limits the benefits of averaging.

▼ How random forest reduce variance?

The idea in random forest is to reduce the variance of the average by reducing the correlation between the trees.

This is achieved in step 1 and step 2 of the algorithm:

Use a bootstrap sample to grow a tree.

Random select d features to be considered to split the node. At each split in the learning process, it inspects a random subset of the features (instead of all features) which reduce the correlation between the trees., i.e. it creates **de-correlated** trees. Reducing d will reduce the correlation between any pair of trees, and thus reduce the variance of the average.

▼ Pros and cons of Random Forest

Pros

1. Has a **better generalization** performance than an individual decision tree due to randomness (the combination of bootstrap samples and using a subset of features), which helps to decrease the model's variance (thus low overfitting). So it corrects decision trees' habit of overfitting the training data.
2. Doesn't require much **parameter tuning**. Using full-grown trees seldom costs much and results in fewer tuning parameters.
3. Less sensitive to **outliers** in the dataset.
4. It generates **feature importance** which is helpful when interpreting the results.

   https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

Con

1. **Computationally expensive**. It is fast to train but quite slow to create predictions once trained. More accurate models require more trees, which means using the model becomes slower.

Overall, for fast, simple, flexible predictive modeling, random forest is probably one of the most useful pragmatic algorithm available today.