Ensemble and Random Forest

1) What do you understand by Ensemble technique?

Ensemble technique has a similar underlying idea where we aggregate predictions from a group of predictors, which may be classifiers or regressors, and most of the times the prediction is better than the one obtained using a single predictor. Such algorithms are called Ensemble methods and such predictors are called Ensembles.

​ The goal of **ensemble methods** is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator.

Two families of ensemble methods are usually distinguished:

* In **averaging methods**, the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.

**Examples:** [Bagging methods](https://scikit-learn.org/stable/modules/ensemble.html#bagging), [Forests of randomized trees](https://scikit-learn.org/stable/modules/ensemble.html#forest), …

* By contrast, in **boosting methods**, base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble.

**Examples:** [AdaBoost](https://scikit-learn.org/stable/modules/ensemble.html" \l "adaboost), [Gradient Tree Boosting](https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting), …

2) Explain the idea behind ensemble techniques.

Ensemble methods take multiple small models and combine their predictions to obtain a more powerful predictive power.

There are few very popular Ensemble techniques which we will talk about in detail such as Bagging, Boosting, stacking etc.

Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to **decrease** **variance** (bagging), **bias** (boosting), or **improve predictions** (stacking).

The main causes of error in learning models are due to **noise, bias and variance**.

**Ensemble methods help to minimize these factors**. These methods are designed to improve the stability and the accuracy of Machine Learning algorithms.

3) What is Bootstrapping? How is sampling done in bootstrapping?

A **bootstrap sample** is a smaller sample that is “bootstrapped” from a larger sample. Bootstrapping is a type of resampling where large numbers of smaller samples of the same size are repeatedly drawn, [with replacement](https://www.statisticshowto.com/sampling-with-replacement-without/), from a single original sample.

There are two parameters that must be chosen when performing the bootstrap: the size of the sample and the number of repetitions of the procedure to perform.

**Sample Size**

In machine learning, it is common to use a sample size that is the same as the original dataset.

### Repetitions

The number of repetitions must be large enough to ensure that meaningful statistics, such as the mean, standard deviation, and standard error can be calculated on the sample.

from sklearn.utils import resample

# data sample

data = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]

# prepare bootstrap sample

boot = resample(data, replace=True, n\_samples=4, random\_state=1)

print('Bootstrap Sample: %s' % boot)

# out of bag observations

oob = [x for x in data if x not in boot]

print('OOB Sample: %s' % oob)

Bootstrapping resamples the original dataset with replacement many thousands of times to create simulated datasets. This process involves drawing random samples from the original dataset. Here’s how it works:

1. The bootstrap method has an equal probability of randomly drawing each original data point for inclusion in the resampled datasets.
2. The procedure can select a data point more than once for a resampled dataset. This property is the “with replacement” aspect of the process.
3. The procedure creates resampled datasets that are the same size as the original dataset.

4) What is bagging?

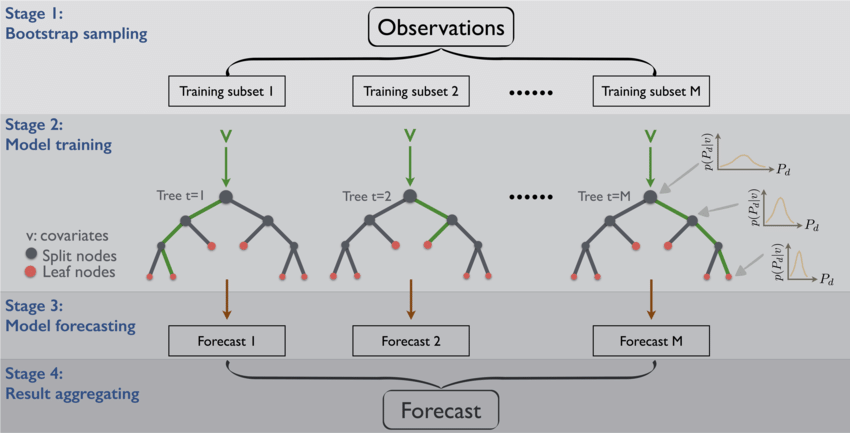
Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method.

An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.

Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithm that have high variance. An algorithm that has high variance are decision trees, like classification and regression trees (CART).

5) How prediction is made in Bagging?

Bootstrap Aggregating is an ensemble method. First, we create random samples of the training data set with replacment (sub sets of training data set). Then, we build a model (classifier or Decision tree) for each sample. Finally, results of these multiple models are combined using average or majority voting.



6) How Ensemble technique solves the high variance issue with Decision trees?

Decision trees are sensitive to the specific data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree can be quite different and in turn the predictions can be quite different.

Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.

Let’s assume we have a sample dataset of 1000 instances (x) and we are using the CART algorithm. Bagging of the CART algorithm would work as follows.

1. Create many (e.g. 100) random sub-samples of our dataset with replacement.
2. Train a CART model on each sample.
3. Given a new dataset, calculate the average prediction from each model.

When bagging with decision trees, we are less concerned about individual trees overfitting the training data. For this reason and for efficiency, the individual decision trees are grown deep (e.g. few training samples at each leaf-node of the tree) and the trees are not pruned. These trees will have both high variance and low bias. These are important characterize of sub-models when combining predictions using bagging.

7) What is pasting? How is it different from bagging?

Pasting is an ensemble technique similar to bagging with the only difference being that there is no replacement done while sampling the training dataset. This causes less diversity in the sampled datasets and data ends up being correlated. That's why bagging is more preffered than pasting in real scenarios.

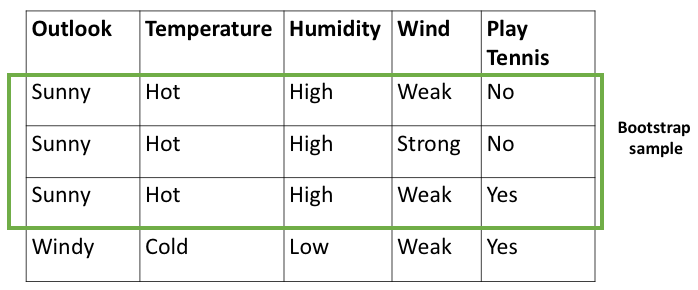
pasting\_knn = BaggingClassifier(KNeighborsClassifier(n\_neighbors=5),

n\_estimators=10, max\_samples=0.5,

bootstrap=False, random\_state=3)

8) What is Out Of Bag evaluation?

In bagging, when different samples are collected, no sample contains all the data but a fraction of the original dataset. There might be some data which are never sampled at all. The remaining data which are not sampled are called out of bag instances. Since the model never trains over these data, they can be used for evaluating the accuracy of the model by using these data for predicition. We do not need validation set or cross validation and can use out of bag instances for that purpose.



9) How does a Random Forest model works?

**random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.**

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning). Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction (see figure below).



10) What is the difference between Bagging and Random forest? Why do we use Random forest more commonly than Bagging?

The main issue with bagging is that there is not much independence among the sampled datasets i.e. there is correlation. The advantage of random forests over bagging models is that the random forests makes a tweak in the working algorithm of bagging model to decrease the correlation in trees. The idea is to introduce more randomness while creating trees which will help in reducing correlation.

ow, the difference with in random forest is how the trees are formed. In bootstraping we allow all the sample data to be used for splitting the nodes but not with random forests. When building a decision tree, each time a split is to happen, a random sample of ‘m’ predictors are chosen from the total ‘p’ predictors. Only those ‘m’ predictors are allowed to be used for the split.

Why is that?

Suppose in those ‘p’ predictors, 1 predictor is very strong. Now each sample this predictor will remain the strongest. So, whenever trees will be built for these sampled data, this predictor will be chosen by all the trees for splitting and thus will result in similar kind of tree formation for each bootstrap model. This introduces correaltion in the dataset and averaging correalted dataset results do not lead low variance. That’s why in random forest the choice for selecting node for split is limited and it introduces randomness in the formation of the trees as well.

Most of the predictors are not allowed to be considered for split.

Generally, value of ‘m’ is taken as m ≈√p , where ‘p’ is the number of predictors in the sample.

When m=p , the random forest model becomes bagging model.

11) What is feature sampling?

Most of the predictors are not allowed to be considered for split.

Generally, value of ‘m’ is taken as m ≈√p , where ‘p’ is the number of predictors in the sample.

When m=p , the random forest model becomes bagging model.

\*This method is also referred as “Feature Sampling”

The above graph represents the decrease in test classifcation error as we select different

values of ‘m’.

12) How prediction is made in Random Forest?

The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree.

13) When should we not use Random forest model?

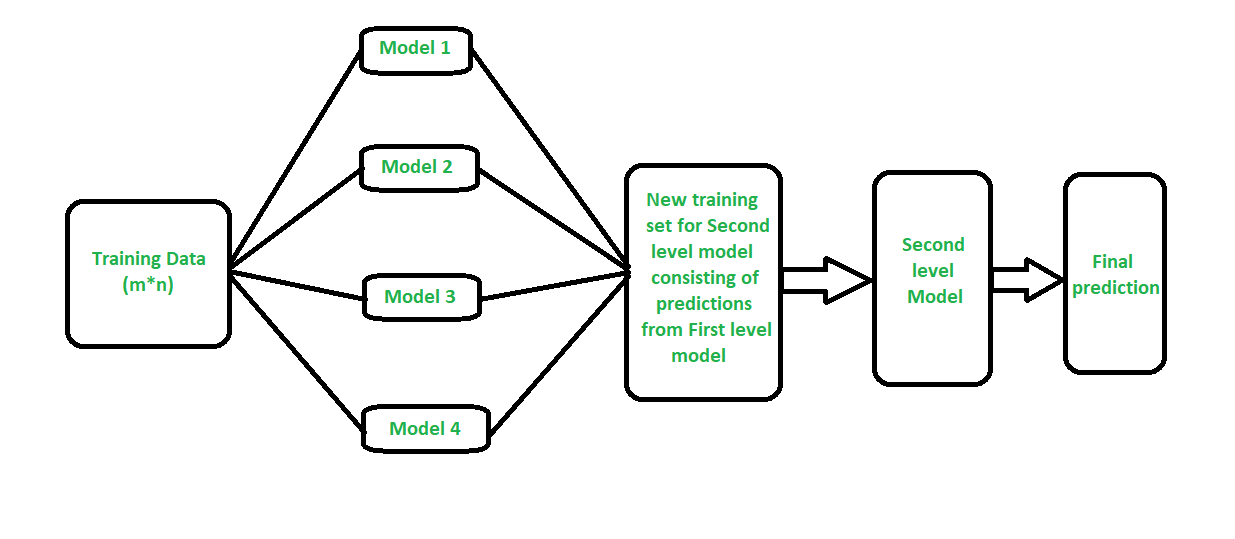
Results of a random forest are very hard to interpret in comparison with decision trees.

High computational time than other respective models.

14) What is Stacking?

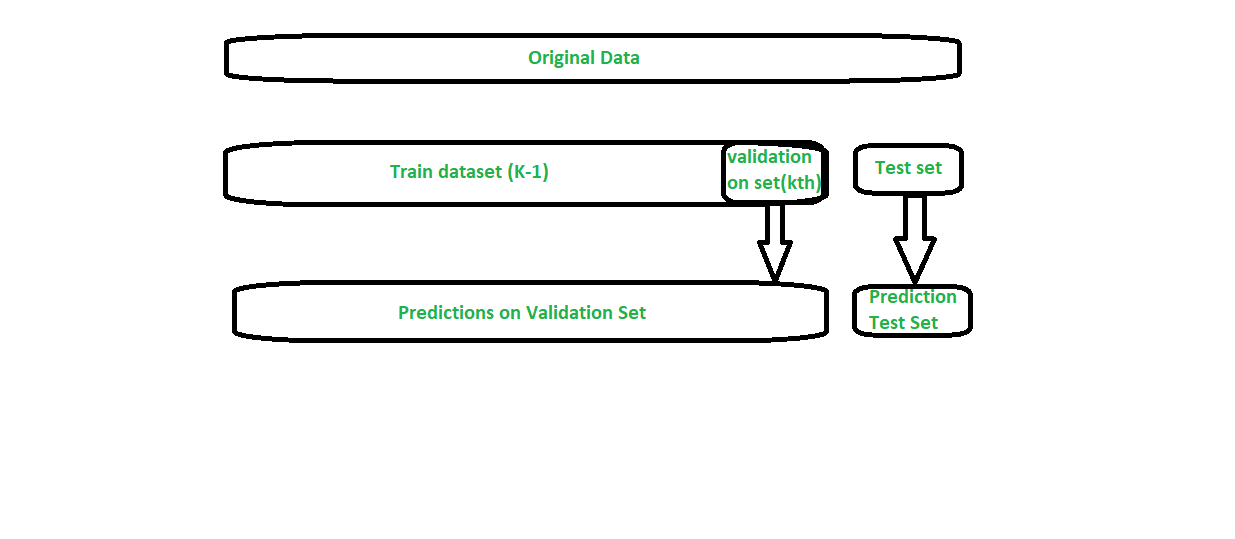
Stacking is a type of ensemble technique which combines the predictions of two or more models, also called base models, and use the combination as the input for a new model (meta-model) i.e. a new model is trained on the predictions of the base models.

15) Explain the working behind Stacking.



**How stacking works?**

1. We split the training data into K-folds just like K-fold cross-validation.
2. A base model is fitted on the K-1 parts and predictions are made for Kth part.
3. We do for each part of the training data.
4. The base model is then fitted on the whole train data set to calculate its performance on the test set.
5. We repeat the last 3 steps for other base models.
6. Predictions from the train set are used as features for the second level model.
7. Second level model is used to make a prediction on the test set.



All three are so-called "meta-algorithms": approaches to combine several machine learning techniques into one predictive model in order to decrease the variance (*bagging*), bias (*boosting*) or improving the predictive force (*stacking* alias *ensemble*).

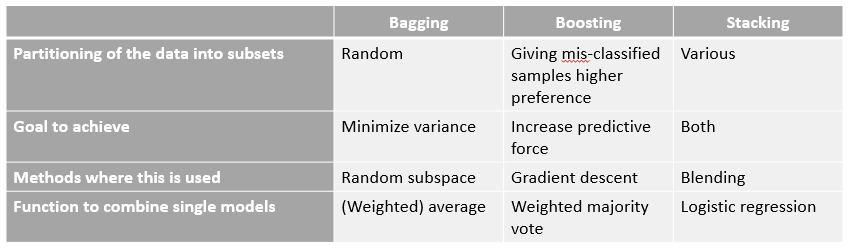
Every algorithm consists of two steps:

1. Producing a distribution of simple ML models on subsets of the original data.
2. Combining the distribution into one "aggregated" model.

Here is a short description of all three methods:

1. [**Bagging**](http://en.wikipedia.org/wiki/Bootstrap_aggregating) (stands for **B**ootstrap **Agg**regat**ing**) is a way to decrease the variance of your prediction by generating additional data for training from your original dataset using [combinations with repetitions](http://en.wikipedia.org/wiki/Combinations) to produce [multisets](http://en.wikipedia.org/wiki/Multiset) of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome.
2. [**Boosting**](http://en.wikipedia.org/wiki/Boosting_(machine_learning)) is a two-step approach, where one first uses subsets of the original data to produce a series of averagely performing models and then "boosts" their performance by combining them together using a particular cost function (=majority vote). Unlike bagging, in the [classical boosting](http://www.cs.princeton.edu/courses/archive/spr08/cos424/readings/Schapire2003.pdf) the subset creation is not random and depends upon the performance of the previous models: every new subsets contains the elements that were (likely to be) misclassified by previous models.
3. [**Stacking**](http://en.wikipedia.org/wiki/Ensemble_learning#Stacking) is a similar to boosting: you also apply several models to your original data. The difference here is, however, that you don't have just an empirical formula for your weight function, rather you introduce a meta-level and use another model/approach to estimate the input together with outputs of every model to estimate the weights or, in other words, to determine what models perform well and what badly given these input data.

Here is a comparison table:

[](https://i.stack.imgur.com/RFfqb.png)

As you see, these all are different approaches to combine several models into a better one, and there is no single winner here: everything depends upon your domain and what you're going to do. You can still treat *stacking* as a sort of more advances *boosting*, however, the difficulty of finding a good approach for your meta-level makes it difficult to apply this approach in practice.

Short examples of each:

1. *Bagging*: [Ozone data](http://en.wikipedia.org/wiki/Bootstrap_aggregating).
2. *Boosting*: is used to improve [optical character recognition](http://en.wikipedia.org/wiki/Optical_character_recognition) (OCR) accuracy.
3. *Stacking*: is used in [classification of cancer microarrays](http://link.springer.com/article/10.1007/s13721-013-0034-x) in medicine.