Ensemble Techniques

We regularly come across many game shows on television and you must have noticed an option of “Audience Poll”. Most of the times a contestant goes with the option which has the highest vote from the audience and most of the times they win. We can generalize this in real life as well where taking opinions from a majority of people is much more preferred than the opinion of a single person. 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.

Let’s suppose we have ‘n’ predictors:

Z1, Z2, Z3, ..., Zn with a standard deviation of σ

Var(z) = σ^2

If we use single predictors Z1, Z2, Z3, ..., Zn the variance associated with each will be σ2 but the expected value will be the average of all the predictors.

Let’s consider the average of the predictors:

μ = (Z1 + Z2 + Z3+.....+ Zn)/n

if we use μ as the predictor then the expected value still remains the same but see the variance now:

variance(μ) = σ^2/n

So, the expected value remained ‘μ’ but variance decreases when we use average of all the predictors.

This is why taking mean is preferred over using single predictors.

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.

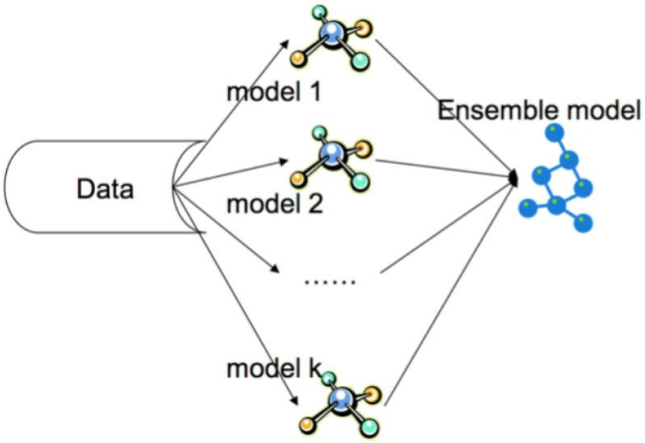
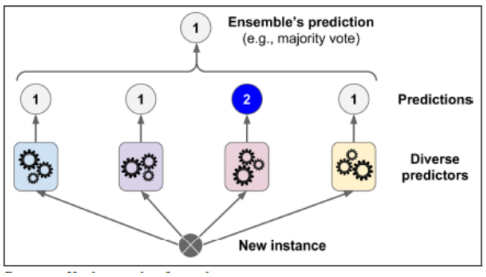


image courtsey: Google



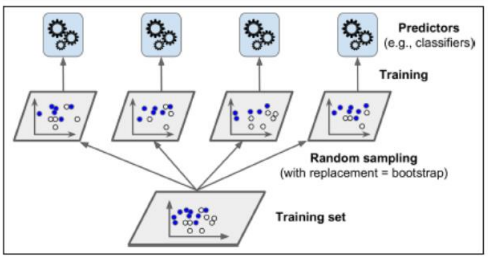
Bagging (Bootstrap Aggregation)

In real life scenarios we don’t have multiple different training sets on which we can train our model separately and at the end combine their result. Here, bootstrapping comes into picture. Bootstrapping is a technique of sampling different sets of data from a given training set by using replacement. After bootstrapping the training dataset, we train model on all the different sets and aggregate the result. This technique is known as Bootstrap Aggregation or Bagging.

Let’s see definition of bagging:

Bagging is the type of ensemble technique in which a single training algorithm is used on different subsets of the training data where the subset sampling is done with replacement (bootstrap). Once the algorithm is trained on all the subsets, then bagging makes the prediction by aggregating all the predictions made by the algorithm on different subsets. In case of regression, bagging prediction is simply the mean of all the predictions and in the case of classifier, bagging prediction is the most frequent prediction (majority vote) among all the predictions.

Bagging is also known as parallel model since we run all models parallely and combine there results at the end.



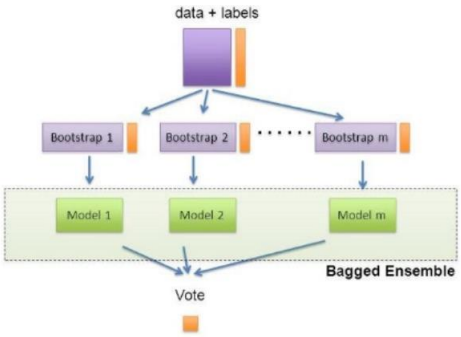


image courtsey: Google

- Advantages of a Bagging Model

1) Bagging significantly decreases the variance without increasing bias.

2) Bagging methods work so well because of diversity in the training data since the sampling is done by bootstraping.

3) Also, if the training set is very huge, it can save computional time by training model on relatively smaller data set and still can increase the accuracy of the model.

4) Works well with small datasets as well.

- **Disadvantage of a Bagging Model

The main disadvantage of Bagging is that it improves the accuracy of the model on the expense of interpretability i.e. if a single tree was being used as the base model, then it would have a more attarctive and easily interpretable diagram, but with use of bagging this interpretability gets lost.

Pasting

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.

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.

Let's see python implementation of Bagging:

let's using bagging over our KNN classifier and see if our score improves:

Great! our score sginificantly improves with use of bagging.

let's not use bootstrap and see the model accuracy! Remember this is "Pasting"

Random Forests

Decision trees are one of such models which have low bias but high variance. We have studied that decision trees tend to overfit the data. So bagging technique becomes a very good solution for decreasing the variance in a decision tree. Instead of using a bagging model with underlying model as a decision tree, we can also use Random forest which is more convenient and well optimized for decision trees. 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.

Let's understand how algorithm works for a random forest model:

1) Just like in bagging, different samples are collected from the training dataset using bootstraping.

2) On each sample we train our tree model and we allow the trees to grow with high depths.

Now, the difference with in random forest is how the trees are formed. In bootstraping we allow all the sample data to b
e used for splitting the nodes but not with random forests. When building a decision tree, each time a split is to ha
ppen, 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. S
o, 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.

*This method is also referred as "Feature Sampling"

3) Once the trees are formed, prediction is made by the random forest by aggregating the predictions of all the model. For regression model, the mean of all the predictions is the final prediction and for classification mode, the mode of all the predictions is considered the final predictions.

Problem Statement

`To build an application to classify the patients to be healthy or suffering from cardiovascular disease based on the given attributes.`

Note

`You can improve the performance by tweaking preprocessing methods`

Let's plot ROC AUC curve to choose best model

Lets find ROC AUC score

Let's check ROC AUC Curve for the fitted model