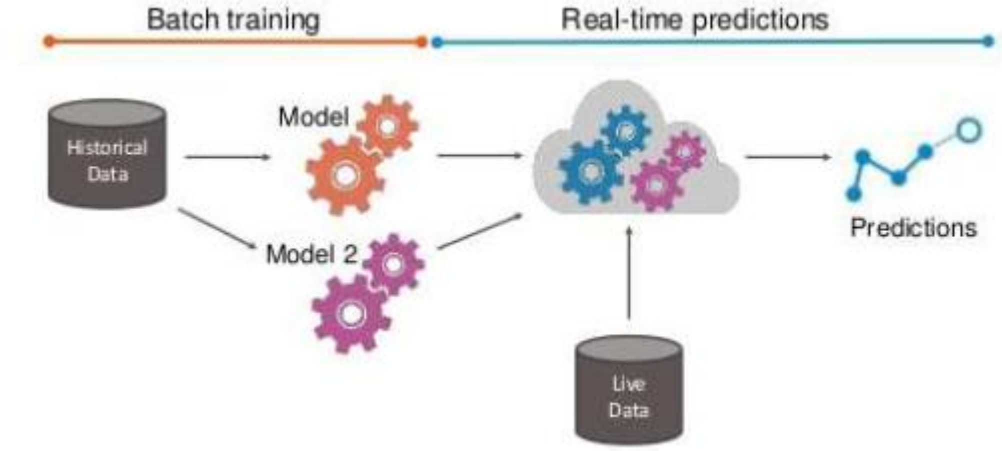
# What is Bagging in Ensemble Learning

In general, any of the machine learning problems we try to find the best possible optimal model for a given problem. That means finding the best possible model within the given model family, for example, finding the best possible decision tree or finding the best possible KNN model. And if we have more time then we can try all model families available, and come up with the best possible regression model, best possible KNN model, best possible SVM model etc. And among these again select the best possible model, which will be either KNN, SVM or any other.

However Ensemble Learning says, if we can build multiple models then why to select the best one why not top 2, again why not top 3 and why not top 10. Then if you find top 10 deploy all 10 models. And when new data comes, make a prediction from all 10 models and combine the predictions and finally make a joint prediction. This is the key idea of ensemble learning.



Now there are two questions which might come to your mind.

1. **What is meant by building/training different models?**
2. **How to combine the predictions?**

So when we say building/training different models then it means either one of the below:

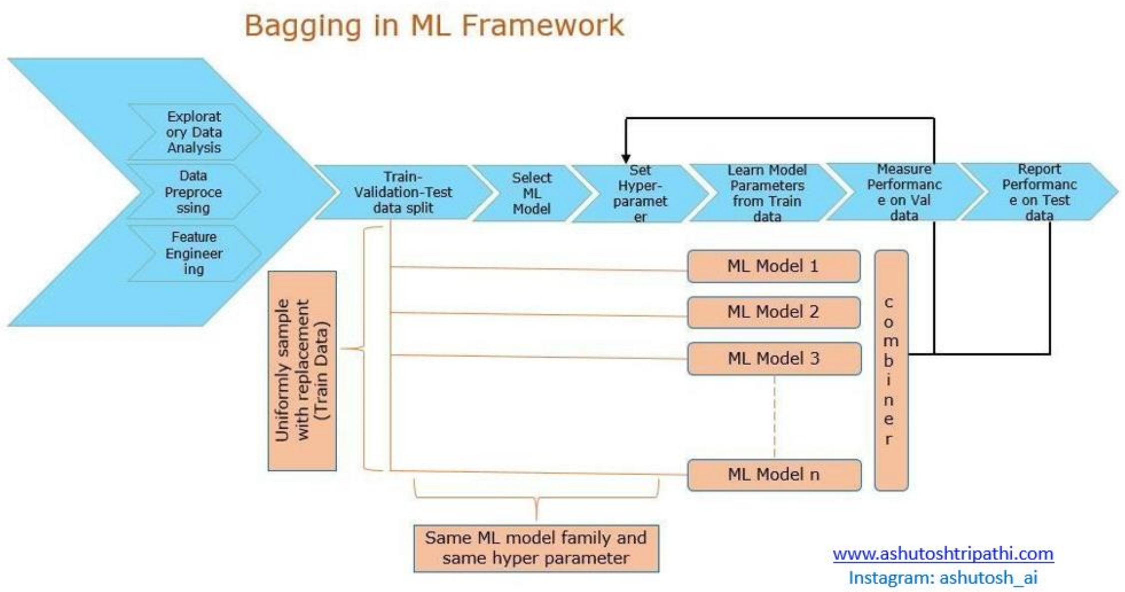
- Select different model families among KNN, Decision Trees, Linear Regression etc.
- Select one model family and Train it on different training samples resulting in different models of same ML family.
- Also can consider different feature spaces while training to result in different models.
- Will talk about the impact of selecting different features in the coming paragraphs.

And when we say combine the predictions of all models then it means:

- Regression: (Weighted) Mean, Median, Max, Min
  - If it is regression problem then you can take either mean, median or mode of all models outcomes to get the final outcome.
  - We take weighted mean when we have build models with a different set of features along with different samples then features with high importance generate the more optimal model and hence we need to assign more weight to those models while combining.
  - Also, we can assign the weight based on the model's performance on the test data. Model performing better will get more weight.
- Classification : (Weighted) Majority Voting.
  - Either normal majority counting or weighted majority counting. (the weighted reason is same as described in above point.)

I hope by now the basic idea of ensemble learning is clear. Let's now discuss one of the most important and widely used techniques of ensemble learning that is Bagging. Another technique is Boosting which will discuss in another post. Here I will focus on complete explanation of Bagging.

# Bagging



Bagging in ML Framework
Exploratory Data Analysis
Data Preprocessing
Feature Engineering
Train-Validation-Test data split
Select ML Model
Set Hyper-parameter
Learn Model Parameters from Train data
Measure Performance on Val data
Report Performance on Test data
Uniformly sample with replacement (Train Data)
ML Model 1
ML Model 2
ML Model 3
ML Model n
combiner
Same ML model family and same hyper parameter
www.ashutoshtripathi.com
Instagram: ashutosh_ai

## Bagging: Bootstrap Aggregation – Basic Idea

1. Create a sample data set from the big data set to train the model.
2. Select one model family from decision trees, KNN etc. of high variance models.
3. Train the model on the sample data. This will be called model 1.

4. Now again create a new sample with replacement strategy which means mix the original dataset with the sample created in step 1 and again create a new fresh sample.
5. Now again train the same model family with new sample data. This will be called model 2.
6. This way we create multiple models by training on different samples data.

One of the most important reasons to use Bagging is that it reduces the model variance. As we have already studied, the best model should have low bias and low variance. But when we move towards the more complex machine learning algorithms bias reduces but variance increases. Hence to get the optimal performance we need to reduce variance also and here Bagging technique becomes very helpful.

Let's understand how Bagging reduces the variance of the model.

Suppose you have build 5 models with the outcome as y1, y2, y3, y4 and y5 respectively.
Then their mean (y1+y2+y3+y4+y5)/5 will be the outcome of the final prediction. Note I am taking mean to explain the concept, you can take median, mode etc based on the problem set.

You remember these outcomes y1, y2, y3, y4, y5 are predicted from the models trained on different samples from the big dataset. Now recall Central Limit Theorem, which says if we have large enough population size and a good number (greater than 30 is optimal) of samples then means of all samples mean will follow the normal distribution and will be same as the population mean.

***And the important one here is the standard deviation of the samples will get reduced by the square root of n where n is the sample size. And the standard deviation is nothing but the square root of the variance hance we can say variance will also get reduced by a factor of 1/n.***

That is how a combiner in Bagging reduces the model variance. And hance Bagging is used with high variance machine learning algorithms like decision trees, KNN and neural networks.

**Some Important points regarding Bagging**

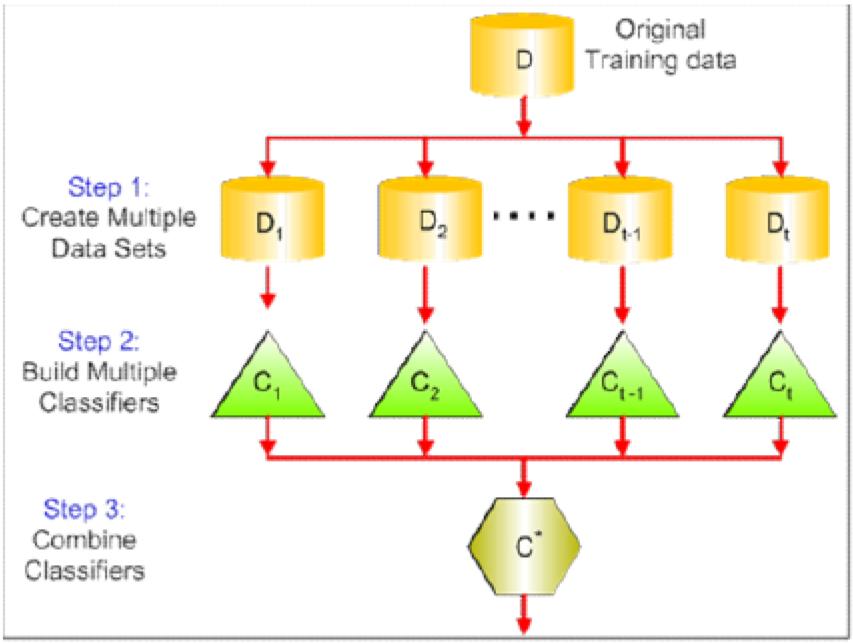
- Algorithm independent: general-purpose technique, can work with any machine learning algorithms. (However preferable is to use with high variance algorithms)
- Well suited for high variance algorithms.
- Variance Reduction: Averaging a set of observations reduces variance – Central Limit Theorem. (explained above)
- Choose # of models to build. (This is hyperparameter, as there is no mathematical formula to determine this)

***One Important thought, how it is different from the K-fold cross-validation technique.***

***In k-fold cross-validation, we use the same training data and do different iteration by choosing different hyperparameter values. It is basically used to find the best possible hyperparameter. However in bagging we use different training data set.***

- Easy to parallelize.
- **Limitation**: Loss of Interpretability, for example, if we take the decision tree as a single model, then it is very easy to interpret however if we build multiple decision trees with the different sample then it becomes a forest and we decide the outcome by combining the outputs of various models here we lose interpretability.
- **Limitation**: **What if one of the features dominates?** If one feature dominates more then if you build multiple decision trees then they all will be exactly same (think about the if-else rule in decision trees) and bagging won't work properly. That is where we do different feature selection also along with different sampling and then assign different weight to the models depending upon the dominating features. And when we involve different feature subspace with different sampling then this method is called Random forest.

Bagging is described using the below infographics on bagging technique.


D
Original Training data
Step 1: Create Multiple Data Sets
$D_1$
$D_2$
$D_{t-1}$
$D_t$
Step 2: Build Multiple Classifiers
$C_1$
$C_2$
$C_{t-1}$
$C_t$
Step 3: Combine Classifiers
$C^*$

So let’s understand Random Feature Subspace in detail.

## Random Feature Subspaces

**Build Different Models using:**

- A different subset of training data (Create samples with replacement)
- A random subset of the features! (In Bagging every time we take all features)
- The same Ml algorithm on each training only above two things gets change.

**Why we need Random Feature Subspaces?**

- Think “regularization”
- If there are one very strong predictor & other moderately strong predictors.
- All models will give high importance to the strong predictor which means all models in the ensemble will be similar.
- This is the reason we do feature subspace along with different samples.
- Choose # of models to build and # of features (m) for each sample from p available features.
- Recommended heuristics to select m out of p is **m = sqrt(p)**
- If m = p: Approach reduces to Bagged Trees. (means we include all the features in every sample training data set)

**Advantages**

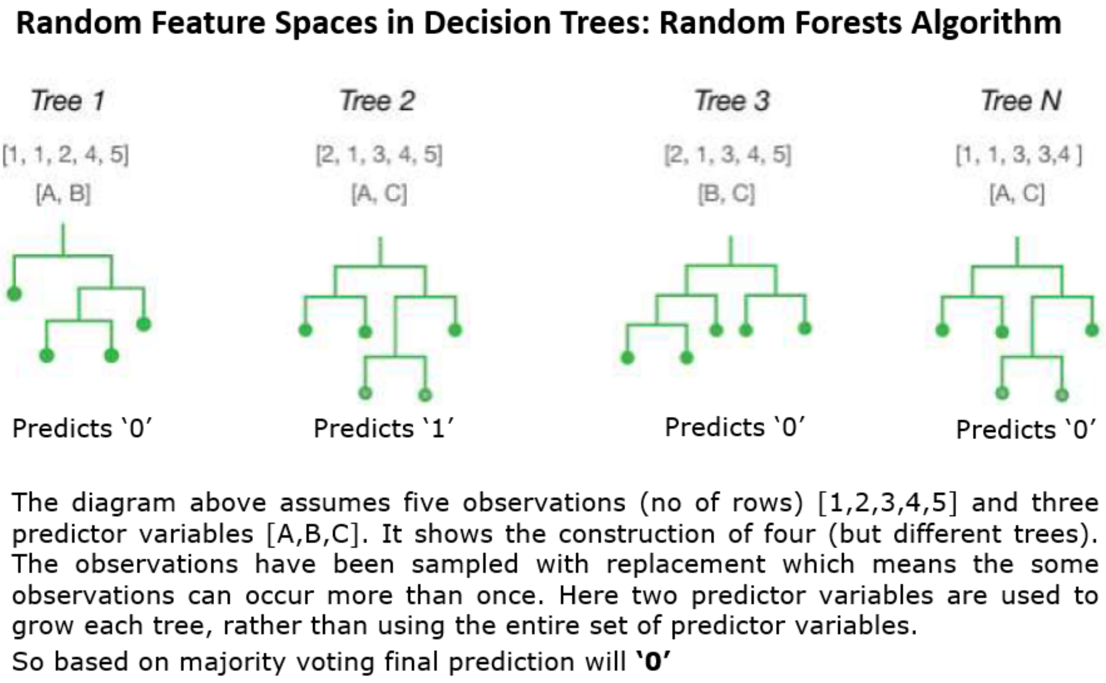
- De-correlates the models in the ensemble
- Improve the accuracy of prediction
- Of-course reduces model variance. This comes from bagged trees.

**Random Feature Spaces in Decision Trees: Which is known as Random Forests Algorithm**

- Decision Trees have high variance.
- The resulting tree (model) depends on the underlying training data.
- Bagged Trees can help reduce variance; Random Forests even more so…

**Random Forests**

- Sample with replacement (shift from 1 training set to Multiple training sets)
- Train model on each training set
- Each tree uses a random subset of the feature: A random forest
- Each DT predicts
- Take Mean / Majority vote prediction for the final prediction
- Faster than bagging (fewer splits to evaluate per tree)



**Random Feature Spaces in Decision Trees: Random Forests Algorithm**

The diagram above assumes five observations (no of rows) [1,2,3,4,5] and three predictor variables [A,B,C]. It shows the construction of four (but different trees). The observations have been sampled with replacement which means the some observations can occur more than once. Here two predictor variables are used to grow each tree, rather than using the entire set of predictor variables.

So based on majority voting final prediction will **'0'**

# Thank You

If you like my Posts on Machine Learning, Please connect with me on

Follow my blog: https://ashutoshtripathi.com/

LinkedIn: https://www.linkedin.com/in/ashutoshtripathi1/

Instagram: https://www.instagram.com/ashutosh_ai/

Medium Articles: https://medium.com/@ashutosh.optimistic