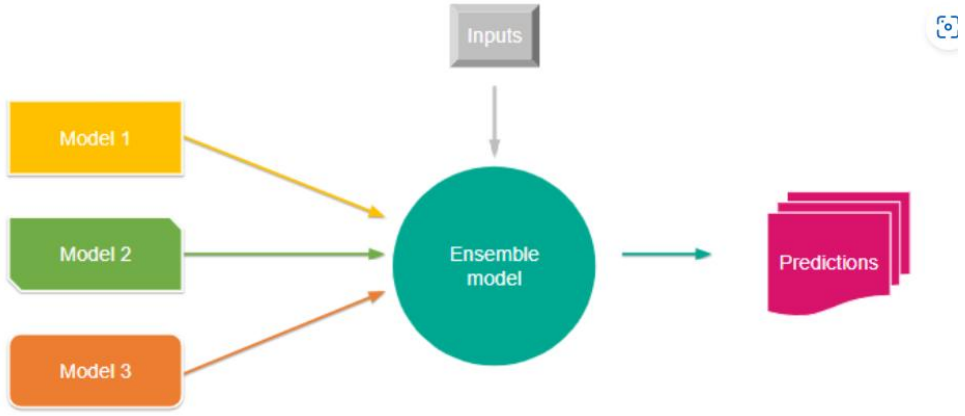
Ensemble Learning

- **The wisdom of the crowd** : Any opinion coming from an aggregation over a diversity of estimates will have more accuracy than an opinion coming from a single expert estimate.
- This technique of combining multiple weak classifiers instead of a strong classifier is called as ensemble learning.

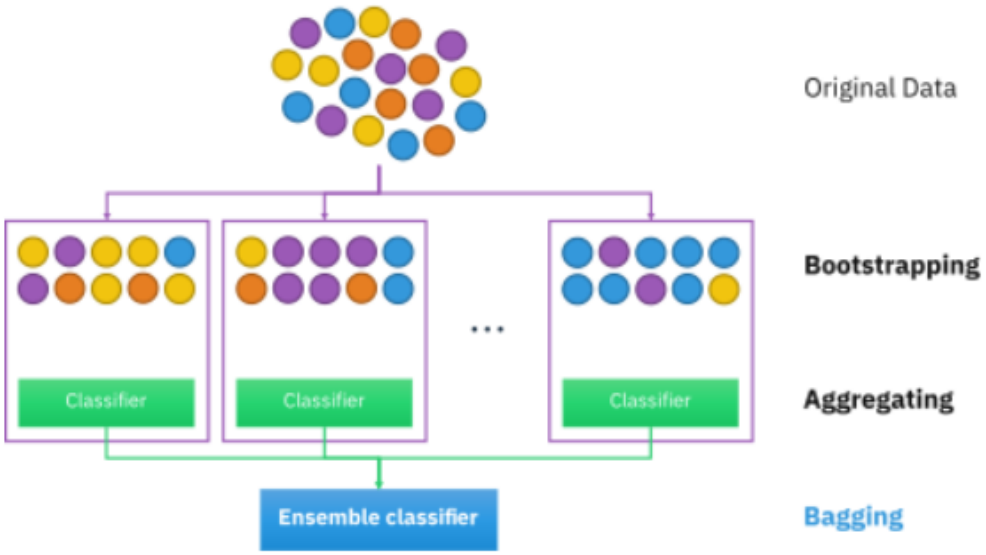
1. **Voting Classifier**

- On a training data train **N diverse classifiers**.
- Once we have the classifiers trained, then on test data predict using the N classifiers.
- Finally take the majority as the final prediction.



2. **Bagging Classifier**

- In bagging classifier unlike voting classifier **we use a single class of algorithm** (base classifier) and train it on different samples of training data (bootstrapped samples).
- Finally once trained we take the aggregate of all the classifiers on the test data.

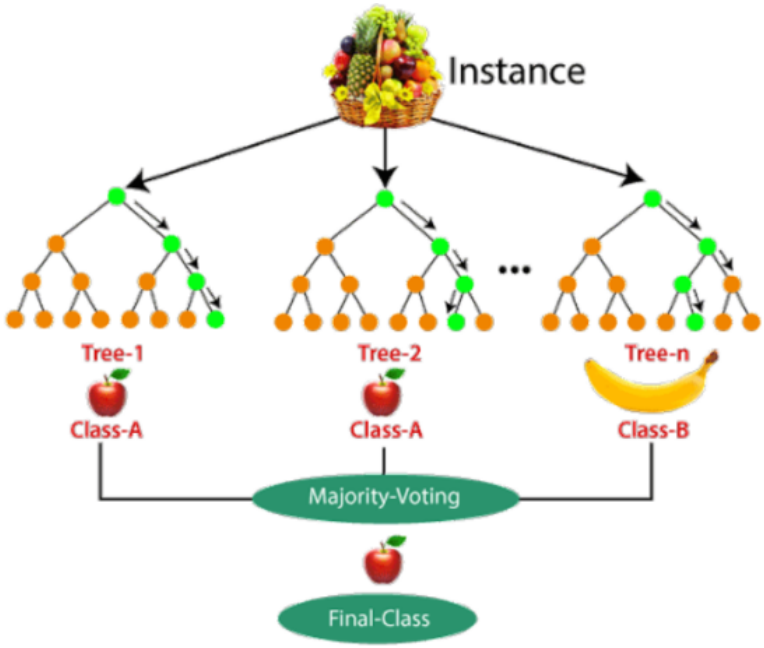


Random Forest Classifier

- The Random Forest classifier is a subset of bagging classifier but it has two differences.

  1. It only uses Decision Tree as base classifier.

  2. While splitting a DT it randomly susbsets the features of the sampled data.



- All of the above algorithms are robust to overfitting.

- You know that overfitting in DT happened due to deep DT's. This created a strong learner but highly overfitting.
- The **Random Classifier ensures that no deep tree is built** but we built various weak learners by randomly choosing columns and that result in (partially grown tree) with high bias and low variance.
- Each weak learner in RF is nothing but a simple mean estimation (high bias).
- By combining several of these high bias weak learners we follow the principle of ensemble learning and we achieve better results.

Important Points

- **Diversity:** Not all attributes are considered while making an individual tree; each tree is different.
- **Immune to the curse of dimensionality:** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization:** Each tree is created independently out of different data and attributes. This means we can fully use the CPU to build random forests.

Hyperparameters :

- **n_estimators:** Number of trees the algorithm builds before averaging the predictions.
- **max_features:** Maximum number of features random forest considers splitting a node.
- **criterion:** How to split the node in each tree? (Entropy/Gini impurity)

Boosting:

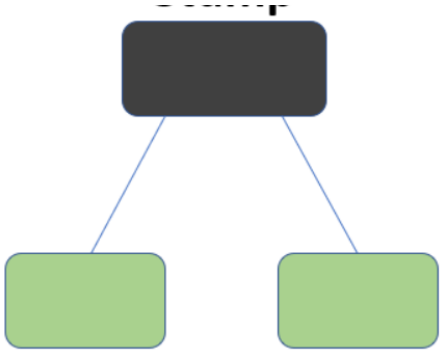
- In boosting classifiers, we add weak learners sequentially such that the successor tries to classify the mistakes of its predecessor.
- Hence you will observe unlike the bagging classifiers here we add model in a sequential manner.
- The boosting technique follows a *sequential* **order**.

Boosting Algorithms:

- Adaboost
- Gradient Boosting
- Xgboost

Adaboost:

- It is also called Adaptive Boosting
- It uses decision tree with depth=1



- It builds a model and assign equal weights to all the data points.
- It then assigns higher weights to points that are wrongly classified.
- Now all the points which have higher weights are given more importance in the next model.
- It will keep training models until and unless a low error is received.

Step1:

Every record is assigned with some weights (1/n), at start, all the weights will be equal.

Step 2:

Create a stump for each of the features (tree with height =1) and then calculate the ***Entropy*** of each stump.

Step 3:

Calculate the total error for this stump.

$$\frac{1}{2}\log\frac{1 - Total\ Error}{Total\ Error}$$

The total error is nothing, but the summation of all the sample weights of misclassified data points. Let's assume there is 1 wrong output, so our total error will be 1/7.

Step 4:

Weight updating

$$New\ sample\ weight\ =\ old\ weight\ *\ e^{\pm Amount\ of\ say\ (\alpha)}$$

The amount of say (alpha) is -ve when the sample is **correctly classified**.

The amount of say (alpha) is +ve when the sample is **wrongly-classified.**

Gradient Boosting Algorithm:

- It builds models sequentially and these subsequent models try to reduce the errors of the previous model.
- This is done by building a new model on the errors or residuals of the previous model.

Steps:

- Fit a simple model.
- Calculate the error residuals (Act Val – Pred Val).
- Fit the new model on error residuals as target variable.
- Fit another model on residuals and repeat until residuals becomes less.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Exp	**Qual**	**Sal**	**B1**	**Res1**	**B2**	**Res2**	**B3**	**Res3**	**B4**	**Res4**	**B5**	**Res5**	
-	-	50	87.5	-38	-30	-7.5	-5	-2.5	-1.9	-0.6	-0.59999	-0	
-	-	75	87.5	-13	-5	-7.5	-4	-3.5	-1.8	-1.7	-1.689	-0	
-	-	100	87.5	12.5	7	5.5	4	1.6	0.4	1.2	1.18	0.02	
-	-	125	87.5	37.5	32	5.5	3	2.3	0.9	1.4	1.28	0.12	
-	-	96	x		y		z		a		b		

XGBoost: (Extreme Gradient Boosting)

- XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
- XGBoost is an implementation of gradient-boosting.
- Tianqi Chen and Carlos authored XGBoost :
- It can handle missing values by itself.
- XGBoost is popular because it's speed, and that speed comes at no cost to accuracy.
- It has cache-aware access.
- XGBoost is used for these two reasons: execution speed and model performance.
- It Runs smoothly on Windows, Linux.
- Languages: Supports all major programming languages including C++, Python, R, Java, Scala, and Julia.
- XGBoost is designed for speed, ease of use, and performance on large datasets.
- When you use XGBoost, **there are no restrictions on the size of your dataset**, so you can work with datasets that are larger than what would be possible with other algorithms.

Difference Between Bagging & Boosting

- The models are created independently.
- The model creation is dependent on the previous ones.

- In Bagging, training data subsets are drawn randomly with a replacement for the training dataset.
- In Boosting, every new subset comprises the elements that were misclassified by previous models.

- Base classifiers are trained parallelly.
- Base classifiers are trained sequentially.