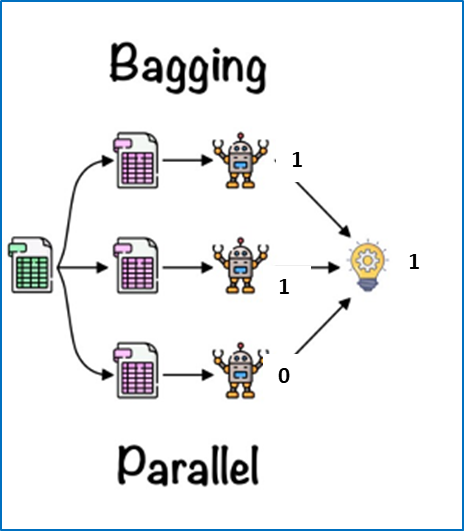
**Random Forest:**

Random Forest Classifier Regressor: Random forest is used for both Classification as well as Regression problems.

**Bagging:**

Bagging fits multiple models on different subsets of a training dataset, models are trained parallelly, then combines the predictions from all models.

1. **Bootstrapping:** creating a parallel model and passing the data in each model by row Sampling and feature sampling with replacement.
2. **Aggregation:** Aggregation refers to the process of combining predictions or outputs from multiple models or algorithms to make a final prediction or decision. Aggregation can be done in various ways depending on the problem and the type of models used. For classification, we take the majority and in terms of regression we take the mean of the results obtained from different models.



The bagging method is prone to Overfitting, (characterized by bias and high variance), which poses a significant challenge in this model. To mitigate high variance, the implementation of techniques such as "preprunning" and "postprunning" can be employed. However, executing these methods can be time-consuming and complex. Alternatively, reducing high variance can be achieved through row and feature sampling(as mentioned above) and constructing the model at an optimal depth. These strategies help maintain a balance between model complexity and generalizability, ensuring more reliable and accurate predictions.

Further to the above, there is one more problem in bagging. In bagging there is a chance that some of the rows or features will never get a chance to be part of the evaluation process, hence to make them part of the evaluation process use the “**Out-of-Bag Evaluation”** process.

**Out-of-Bag Evaluation:**

In a random forest, during the construction of each individual decision tree, some data points are not used for training and are left out. These "out-of-bag" data points are not part of the training set for that particular tree and when you set the `oob\_score=True` parameter in the random forest, it means that the out-of-bag data points are used for validation. These data points are fed into each corresponding decision tree that did not use them during training, and their predictions are collected.

The out-of-bag predictions are then compared with the actual outcomes of the corresponding data points to calculate the out-of-bag score. This score provides an estimate of how well the random forest is likely to perform on new, unseen data.

The out-of-bag error is simply the error rate derived from the out-of-bag predictions. It represents the prediction error of the random forest on unseen data and gives an idea of how well the model will generalize to new examples.

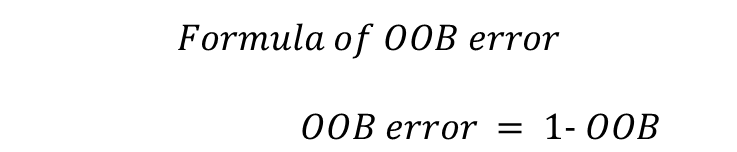
The beauty of this approach is that it allows us to evaluate the model's performance without the need for a separate validation dataset. By leveraging the out-of-bag data points, we can estimate the model's accuracy on unseen data, making the out-of-bag evaluation a very useful technique in the context of random forests.

**The OOB score:**

The OOB score: is the number of correctly predicted data on OOB samples taken for validation. It means that the more the error bottom model does, the Less the OOB score for the bottom model.

**The OOB error:**

The OOB error: can be calculated using the below formula score shown below if the OOB score is 0.83.



What is the “OOB error” **if “OOB score is 0.83”**OOB error = 1- 0.83  
OOB error = 0.17

**Pre-pruning:**

"Pre-pruning" refers to a technique used in decision tree algorithms, where the tree is not allowed to grow beyond a certain threshold **during the training phase**. This means that the tree stops expanding as soon as certain criteria are met, without fully developing all possible branches. Pre-pruning is the opposite of post-pruning, where the tree is first grown to its maximum size and then pruned back.

There are several common pre-pruning techniques:

**Maximum Depth**: Limiting the maximum depth of the tree. Once a node reaches this depth, it stops growing, even if it still contains impure subsets.

**Minimum Samples Split**: Specifying a minimum number of samples required to split an internal node. If the number of samples is below this threshold, the node is not split further.

**Minimum Samples Leaf:** Specifying a minimum number of samples required to be at a leaf node. If a split results in a leaf node with fewer samples than this threshold, the split is not performed.

**Maximum Features:** Limiting the number of features considered when looking for the best split. This can help prevent overfitting by reducing the complexity of the tree.

Pre-pruning helps prevent overfitting by stopping the tree from becoming too complex and capturing noise in the training data. It can also reduce computational resources and improve the interpretability of the resulting tree.

However, prepruning may lead to underfitting as well if the tree is prematurely pruned, resulting in a model that fails to capture important patterns in the data. Therefore, it's important to carefully tune the prepruning parameters to find the optimal balance between underfitting and overfitting.

**PostPrunning:**

"Post-pruning," also known as "pruning," is a technique used in decision tree algorithms to prevent overfitting after the tree has been fully grown. The process involves removing or collapsing parts of the tree that are not providing significant improvements in predictive accuracy or that are likely to be capturing noise in the training data.

Here's how post-pruning typically works:

1. \*\*Grow the Tree\*\*: Initially, the decision tree is grown to its full size using a recursive binary splitting process. This process continues until each leaf node contains either pure samples (all belonging to the same class for classification or having similar target values for regression) or until a stopping criterion is met (such as reaching a maximum depth or having too few samples in a node).

2. \*\*Pruning Process\*\*: Once the tree is fully grown, pruning begins. The goal is to remove unnecessary branches or nodes to improve the tree's generalization performance on unseen data.

3. \*\*Evaluation of Nodes\*\*: Nodes in the tree are evaluated based on their importance or predictive ability. Common measures used for this evaluation include:

- \*\*Impurity Measures\*\*: Gini impurity for classification trees or mean squared error for regression trees.

- \*\*Statistical Tests\*\*: Chi-square test, Fisher's exact test, or others to determine if splitting a node provides a significant improvement in predictive accuracy.

4. \*\*Pruning Decision\*\*: Nodes that do not provide sufficient improvement in predictive accuracy or meet certain criteria (such as not meeting a minimum impurity reduction threshold) are candidates for pruning.

5. \*\*Pruning Actions\*\*: Pruning actions can involve collapsing a subtree into a single leaf node, removing the subtree altogether, or merging adjacent leaf nodes with similar predicted outcomes.

6. \*\*Validation Set\*\*: The decision to prune nodes is typically based on performance metrics evaluated on a validation set or through cross-validation.

7. \*\*Stopping Criterion\*\*: Pruning continues until no further improvements in predictive accuracy are observed, or until a predetermined stopping criterion is met.

Post-pruning helps prevent overfitting by simplifying the tree structure and removing branches that are likely to be capturing noise or irrelevant features in the training data. This results in a more generalized and interpretable decision tree model. However, it's essential to strike a balance between removing unnecessary complexity and preserving important predictive information in the tree. Proper tuning of post-pruning parameters is crucial to achieve optimal model performance.