# Ensemble

**Ensemble Methods in Machine Learning:**

Ensemble methods involve combining the predictions of multiple models to improve overall performance and robustness. The core idea is to leverage the diversity of multiple models to achieve better generalization and accuracy.

1. **Voting:**
   * **Classification:** Involves combining the predictions of multiple classifiers and selecting the class label that receives the majority of votes (hard voting) or a weighted combination of class probabilities (soft voting).
   * **Regression:** Takes the average (mean) of predictions from multiple regressors.
2. **Stacking:**
   * Stacking combines the predictions of multiple models, often referred to as base models, by passing them as input to a final model (meta-model).
   * The final model learns how to best combine the predictions of the base models, usually through training on a hold-out dataset.
3. **Bagging (Bootstrap Aggregating):**
   * Involves training multiple instances of the same base model on different bootstrap samples (randomly sampled with replacement) of the training dataset.
   * Aggregates the predictions of individual models, often by averaging (regression) or voting (classification).
   * **Random Forest:** A specific bagging algorithm that uses decision trees as base models.
4. **Boosting:**
   * Boosting focuses on improving the accuracy of a single base model by iteratively giving more weight to misclassified instances.
   * Base models are trained sequentially, with each subsequent model correcting errors made by the previous ones.
   * Popular algorithms include AdaBoost, Gradient Boosting, and XGBoost.

**Additional Notes:**

* **Random Forest:** It's worth mentioning that Random Forest, although a form of bagging, introduces additional randomness by considering only a subset of features at each split in decision trees. This further enhances diversity among base models.
* **Weighted Voting in Stacking:** In stacking, the final model assigns different weights to the predictions of each base model. This allows the stacking model to emphasize the expertise of certain base models over others.

Benefits of Ensemble

* Improvement in performance
* Low bias and Low variance
* Robustness

## Voting Ensemble

In a voting ensemble, multiple models (which can be different algorithms or variations of the same algorithm) are trained independently on the same dataset. When making predictions on new, unseen data, the predictions from each model are combined, often through majority voting for classification problems or by taking the mean for regression problems.

Regarding the assumptions:

1. **All Models Should Be Different:**
   * While it's a common practice to use diverse models in ensemble methods to capture different aspects of the data, it's not a strict requirement. Ensembles can still be effective with similar models if they exhibit complementary strengths and weaknesses.
2. **Accuracy of All Models Should Be Greater Than 50%:**
   * In general, it is desirable for individual models to perform better than random chance (50% accuracy). However, the success of an ensemble doesn't solely depend on the individual models having accuracy greater than 50%. The key is that the models should contribute unique information and should not be overly correlated.

Now, if we have three models in a voting ensemble, and each model has an accuracy of 75%, this is a positive scenario. It indicates that each individual model is performing well above random chance. In a voting ensemble, the combination of these models may lead to even better overall performance, especially if the models make errors on different instances, providing complementary information.

When combining the outputs of these models, the ensemble's accuracy might be higher than that of any individual model. The ensemble approach tends to be more robust and less prone to overfitting compared to individual models.

### Core Idea intuition

### Hard Voting and Soft Voting

Hard voting and soft voting are two different strategies for combining the predictions of multiple models in ensemble methods, especially in the context of voting classifiers.

1. **Hard Voting:**
   * In hard voting, each model in the ensemble (voting classifier) predicts a class label, and the final prediction is determined by a simple majority vote.
   * For a binary classification problem, the class with the majority of votes becomes the ensemble's prediction. In the case of multi-class classification, the class with the highest number of votes is selected.
   * Hard voting is most effective when the individual models in the ensemble are diverse and may make errors on different instances.

Example:

* + Model 1 predicts class A.
  + Model 2 predicts class B.
  + Model 3 predicts class A.
  + Hard voting result: Class A (since it has more votes).

1. **Soft Voting:**
   * In soft voting, each model in the ensemble provides a probability estimate for each class, and the final prediction is based on the average probability across all models.
   * For binary classification, if the average probability for class 1 is greater than 0.5, the ensemble predicts class 1; otherwise, it predicts class 0.
   * For multi-class classification, the class with the highest average probability is selected.
   * Soft voting can be advantageous when models output probability estimates and those estimates are well-calibrated.

Example:

* + Model 1 predicts [0.3, 0.7].
  + Model 2 predicts [0.4, 0.6].
  + Model 3 predicts [0.2, 0.8].
  + Soft voting result: [0.3, 0.7] (average probabilities for each class).

**Considerations:**

* Hard voting is generally suitable for classifiers that make discrete class predictions.
* Soft voting is suitable when classifiers provide probability estimates or confidence scores.
* Soft voting can be more informative and might be preferred when individual models provide well-calibrated probability estimates.
* The choice between hard and soft voting depends on the nature of the problem, the characteristics of the models, and the type of output they produce.

## Bagging

1. **Bootstrapping:**
   * Bagging involves creating multiple subsets of the original dataset through a process called bootstrapping.
   * Bootstrapping is a random sampling technique where multiple samples (with replacement) are drawn from the original dataset to create diverse subsets.
   * Each subset is used as a training set for an individual model.
2. **Aggregating:**
   * Multiple models are trained independently on these bootstrapped subsets of the data.
   * The predictions from individual models are then aggregated to form a combined or ensemble prediction.
3. **Reducing Variance:**
   * One of the main advantages of bagging is its ability to reduce variance.
   * By training models on different subsets of the data, bagging helps to decrease the impact of noise and outliers in the dataset, leading to a more robust and stable ensemble.
4. **Prediction Aggregation:**
   * In classification problems, the final prediction is often determined by majority voting (mode), where each model "votes" for a class, and the class with the most votes is chosen.
   * In regression problems, the predictions from individual models are averaged to produce the final output.
5. **Parallel Training:**
   * Bagging allows for parallel training of models, making it computationally efficient.
   * Each model is trained independently, making bagging suitable for parallel and distributed computing.

### Bagging for Classification:

1. **Data Sampling (Bootstrapping):**
   * Randomly sample subsets of the training data with replacement (bootstrapping).
   * Each subset is used to train an individual classifier.
2. **Individual Classifier Training:**
   * Train multiple classifiers (e.g., decision trees) independently on each bootstrapped subset.
3. **Prediction Aggregation:**
   * For a new input, each classifier predicts a class.
   * The final prediction is determined by a majority vote (hard voting). The class with the most votes across all classifiers is chosen.
4. **Example:**
   * If you have three classifiers predicting classes A, B, and A, the aggregated prediction would be class A.

### Bagging for Regression:

1. **Data Sampling (Bootstrapping):**
   * Randomly sample subsets of the training data with replacement (bootstrapping).
   * Each subset is used to train an individual regressor.
2. **Individual Regressor Training:**
   * Train multiple regressors (e.g., decision trees) independently on each bootstrapped subset.
3. **Prediction Aggregation:**
   * For a new input, each regressor predicts a continuous value.
   * The final prediction is determined by averaging the predictions of all regressors.
4. **Example:**
   * If you have three regressors predicting values 10, 12, and 11, the aggregated prediction would be the average: (10 + 12 + 11) / 3 = 11.

### Key Points:

* **Diversity:**
  + Bagging works well when individual models exhibit diversity (e.g., due to different training subsets or subsets of features).
* **Reduction of Variance:**
  + Bagging is particularly effective in reducing variance, making it beneficial when individual models are sensitive to noise or outliers.
* **Parallelization:**
  + Bagging allows for parallel training of models, as each model is trained independently.
* **Popular Algorithm:**
  + Random Forest is a popular bagging algorithm for both classification and regression. It extends basic bagging by introducing randomness in feature selection during each split in decision trees.

### Random Forest Overview:

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training. The "forest" is formed by combining the predictions of these individual trees to make a more robust and accurate prediction.

Each tree in the Random Forest is trained on a different subset of the original dataset, sampled with replacement. This process is known as bootstrapping. Bootstrapping introduces diversity among the trees, making the ensemble less prone to overfitting.

At each split in a decision tree, a random subset of features is considered for determining the best split. This further enhances the diversity of individual trees and prevents a single dominant feature from overpowering the entire model.

#### Bias-Variance Trade-Off:

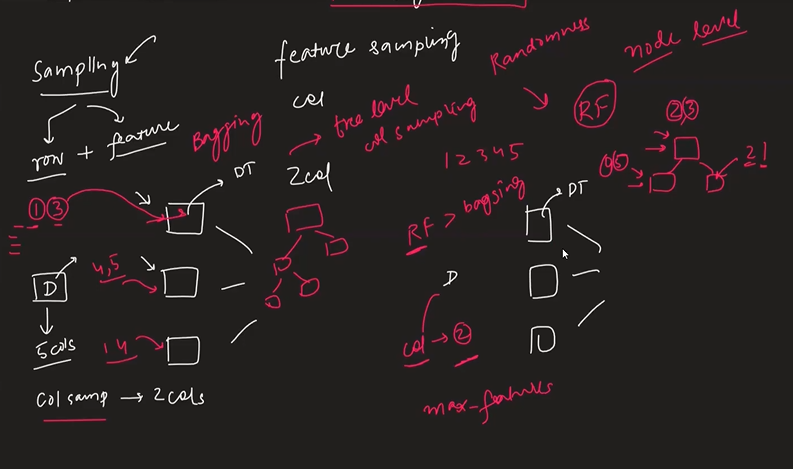
1. **Bias and Variance:**
   * **Bias:** Bias refers to the error introduced by approximating a real-world problem, which may be extremely complex, by a simplified model. High bias models may oversimplify the problem.
   * **Variance:** Variance refers to the model's sensitivity to fluctuations in the training data. High variance models may capture noise in the training data, leading to poor generalization to new data.
2. **Fully Grown Models:**
   * Fully grown decision trees, SVM, and KNN can often face low bias but high variance. These models may capture complex patterns in the training data but may not generalize well to new data.
3. **Low Bias, Low Variance Goal:**
   * The ideal model strikes a balance between low bias and low variance, achieving good performance on both the training and test data.
4. **Role of Random Forest:**
   * Random Forest overcomes the bias-variance trade-off by building an ensemble of weak learners (decision trees) and introducing randomness in the learning process.
   * The randomness comes from both bootstrapped data samples and random feature selection at each split.
   * The combination of diverse trees reduces the overall variance of the model.

#### Advantages of Random Forest:

1. **Less Prone to Overfitting:**
   * Random Forest is less prone to overfitting compared to individual decision trees.
2. **Improved Generalization:**
   * The ensemble nature of Random Forest improves generalization to new, unseen data.
3. **Less Sensitive to Hyperparameters:**
   * Random Forest often provides good results even without extensive hyperparameter tuning, making it more user-friendly.

#### Difference between Bagging and Random Forest:

1. Any base model can be used in Bagging whereas in Random Forest only Decision tree is the base model and it cannot be changed.
2. Feature sampling, is another major difference between bagging and Random Forest. At every node the feature sampling is performed in Random Forest whereas in Bagging the feature sampling is performed at tree level.



#### Out of bag:

In Random Forest, each tree is trained on a bootstrapped subset of the original dataset. As a result some observations are left out or not included in the training set for each tree. The observations that are not included in the training set of a particular tree are referred to as a ‘out of bag’ sampled for that tree. The OOB samples serve as a kind of unseen dataset for each individual tree. Since, these OOB sample are not used in the training of that specific tree, it can be used to evaluate its performance. With the OOB error we can estimate that how well the Random Forest is likely to generalize to unseen data.