Machine Learning
SUBJECT CODE: 203105515

Dr. Ashwini Kumar Jha, Associate Professor
Computer Science & Engineering

CHAPTER - 6

Ensemble Learning

Parul® University

Contents

- ❑ Ensemble Learning
- ❑ Bagging and Boosting
- ❑ Random forests
- ❑ Adaboost
- ❑ G boost inclusive

Introduction to Ensemble Learning

- **Ensemble:** a group of items viewed as a whole rather than individually
- Ensemble learning combines multiple models to obtain better model performance.
- It helps you improve robustness and provide a generalized model.
- In short, it combines different decisions from the model to improve performance.
- In the other words, Ensemble learning is one of the most powerful machine learning techniques that use the combined output of two or more models/weak learners and solve a particular computational intelligence problem. E.g., a Random Forest algorithm is an ensemble of various decision trees combined

Introduction to Ensemble Learning

- ***An ensembled model is a machine learning model that combines the predictions from two or more models."***
- The idea behind ensemble learning is that by aggregating the output of multiple models, the overall performance can be improved compared to using any single model alone.
- Ensemble methods are particularly effective when individual models may have different strengths and weaknesses or when there is noise or uncertainty in the data.
- Ensemble learning can be applied to various types of machine learning tasks, including classification, regression, and even anomaly detection.

Why Ensemble Learning is important

Ensemble learning is important in machine learning and data science for several compelling reasons:

1. Improved Predictive Performance: One of the primary motivations for using ensemble learning is that it often leads to better predictive performance compared to single models. By combining the predictions of multiple models, ensemble methods can reduce bias and variance, leading to more accurate and robust predictions. This can result in higher accuracy, lower error rates, and improved generalization to new, unseen data.

2. Reduction of Overfitting: Ensemble methods can mitigate the risk of overfitting, a common problem in machine learning where a model performs well on the training data but poorly on new data. By combining multiple models that may overfit differently, ensemble learning can reduce the overall overfitting effect.

Why Ensemble Learning is important

3. Robustness: Ensembles are more robust to noisy or erroneous data. Outliers or mislabeled data points that can significantly affect a single model's performance are less likely to have a substantial impact on ensemble predictions because they are smoothed out when combining multiple models.

4. Handling Model Diversity: Ensemble methods allow you to leverage the diversity of different base models. Individual models may have different strengths and weaknesses, and ensembles can exploit this diversity to capture a broader range of patterns and insights in the data.

5. Versatility: Ensemble learning can be applied to various machine learning algorithms and tasks, including classification, regression, ranking, and even anomaly detection. This versatility makes it a valuable tool in a wide range of applications.

Why Ensemble Learning is important

6. Stability and Consistency: Ensemble methods tend to produce more stable and consistent results across different datasets and under various conditions. This makes them a preferred choice in situations where model performance needs to be reliable and consistent.

7. State-of-the-Art Performance: In many machine learning competitions and real-world applications, ensemble methods have been shown to achieve state-of-the-art performance. They are commonly used by data scientists and machine learning practitioners to push the boundaries of what is achievable with machine learning models.

8. Risk Mitigation: By combining multiple models, ensemble learning can help mitigate the risk associated with relying on a single model's predictions, especially in critical applications like finance, healthcare, and autonomous systems.

Simple Ensemble Techniques

1. Max Voting

The maximum vote method is commonly used for classification problems. This technique uses multiple models to make predictions for each data point. Predictions from each model are considered “votes.” The predictions from most of the models are used as final predictions.Let’s take an example where we have three classifiers with the following predictions:

Classifier 1 – Class B

Classifier 2 – Class B

Classifier 3 – Class A

The final prediction here will be class B with the most votes.

Simple Ensemble Techniques

2. Averaging

With averaging, the final output will be the average of all predictions. This applies to regression problems. For example, in random forest regression, the final result is the average of predictions from individual decision trees.

Regressor 1 – 500
Regressor 2 – 200
Regressor 3 – 100
The final prediction will be the average of 500, 200, and 100.

Simple Ensemble Techniques

3. Weighted Averaging

A weighted average emphasizes the underlying model with high predictive power. each regressor is assigned a weight. Because the model weights are only small positive values and the sum of all weights equals 1, the weights can indicate each model’s confidence or expected performance percentage.

Suppose the regressors are given weights of 0.35, 0.2, and 0.1, respectively. The final model prediction can be computed as

$P=W1*p1+ W2*p2+ W3*p3$

Where W1, W2, W3 are the weights. And p1,p2, and p3 are predictions by different models.

P=Final prediction

$0.35 * 100 + 0.2 * 200 + 0.1 * 500 = 285.$

Common Ensemble Learning Methods

There are 3 most common ensemble learning methods in machine learning. These are as follows:

- ❑ Bagging
- ❑ Boosting
- ❑ Stacking

Bagging (Bootstrap Aggregating)

- Bagging, the short form for bootstrap aggregating, is mainly applied in classification and regression.
- Bagging is a technique where multiple instances of the same base model are trained on different random subsets of the training data with replacement (bootstrap samples).
- The predictions of these individual models are then aggregated to make the final prediction.

Bagging (Bootstrap Aggregating)

It is generally completed in two steps as follows:

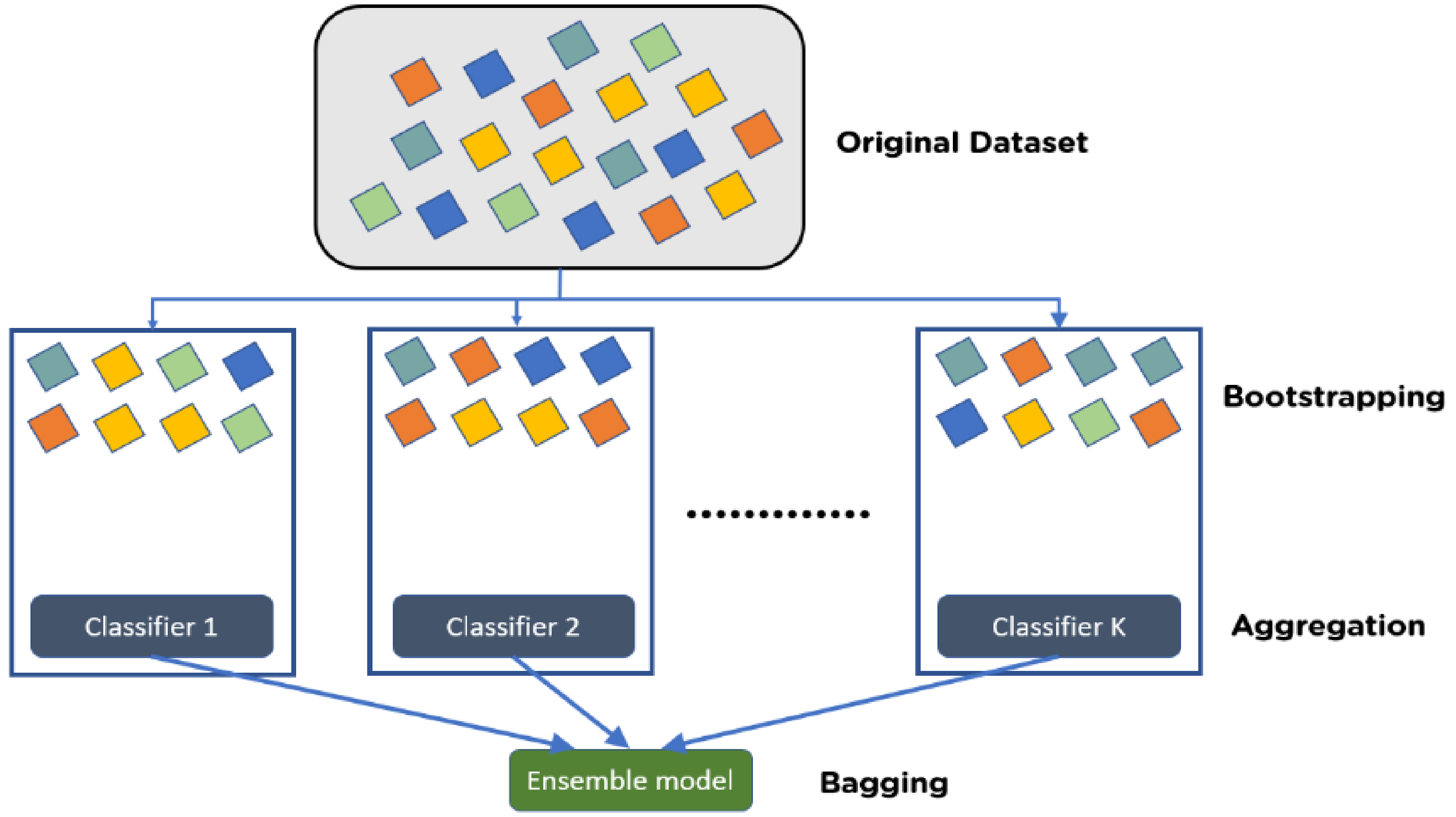
•**Bootstrapping:** It is a random sampling method that is used to derive samples from the data using the replacement procedure. In this method, first, random data samples are fed to the primary model, and then a base learning algorithm is run on the samples to complete the learning process.

•**Aggregation:** This is a step that involves the process of combining the output of all base models and, based on their output, predicting an aggregate result with greater accuracy and reduced variance.

Implementation steps of Bagging –

1. Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
2. A base model is created on each of these subsets.
3. Each model is learned in parallel from each training set and independent of each other.
4. The final predictions are determined by combining the predictions from all the models

Bagging (Bootstrap Aggregating)



Bagging Example

Example: In the Random Forest method, predictions from multiple decision trees are ensembled parallelly. Further, in regression problems, we use an average of these predictions to get the final output, whereas, in classification problems, the model is selected as the predicted class

Application of the Bagging:

There are various applications of Bagging, which are given below

1. IT: Bagging can also improve the precision and accuracy of IT structures, together with network intrusion detection structures. In the meantime, this study seems at how Bagging can enhance the accuracy of network intrusion detection and reduce the rates of fake positives.

2. Environment:

Ensemble techniques, together with Bagging, were carried out inside the area of far-flung sensing. This study indicates how it has been used to map the styles of wetlands inside a coastal landscape.

Application of the Bagging:

3. Finance:

Bagging has also been leveraged with deep gaining knowledge of models within the finance enterprise, automating essential tasks, along with fraud detection, credit risk reviews, and option pricing issues. This research demonstrates how Bagging amongst different device studying techniques was leveraged to assess mortgage default hazard. This highlights how Bagging limits threats by saving you from credit score card fraud within the banking and economic institutions.

4. Healthcare:

The Bagging has been used to shape scientific data predictions. These studies (PDF, 2.8 MB) show that ensemble techniques had been used for various bioinformatics issues, including gene and protein selection, to perceive a selected trait of interest. More significantly, this study mainly delves into its use to expect the onset of diabetes based on various threat predictors.

Advantages of Bagging are -

There are many advantages of Bagging. The benefit of Bagging is given below –

1. **Easier for implementation:** Python libraries, including scikit-examine (sklearn), make it easy to mix the predictions of base beginners or estimators to enhance model performance. Their documentation outlines the available modules you can leverage for your model optimization.
2. **Variance reduction:** The Bagging can reduce the variance inside a getting to know set of rules which is especially helpful with excessive-dimensional facts, where missing values can result in better conflict, making it more liable to overfitting and stopping correct generalization to new datasets.

Disadvantages of Bagging are -:

There are many advantages of Bagging. The benefit of Bagging is given below -

1.Increased Model Complexity: Bagging often involves the creation of a large number of base models (e.g., decision trees), which can lead to a more complex ensemble. This can make it challenging to interpret and understand the combined model.

2.Limited Improvement for Stable Models: Bagging is particularly effective when base models are unstable and have high variance. If the base model is already stable and low in variance, bagging may not provide significant improvements.

3.No Explicit Feature Selection: Bagging typically doesn't perform feature selection, meaning all features are used in the training of base models. This can be inefficient and less effective when dealing with high-dimensional datasets where many features are irrelevant.

Disadvantages of Bagging are -:

4. Slower Training: Creating multiple base models and aggregating their predictions can be computationally expensive and time-consuming, especially when working with large datasets or complex base models.

5. Sensitivity to Noise: Bagging can still be sensitive to noisy or mislabeled data points if they are included in multiple bootstrap samples. While it reduces the impact of individual outliers, it doesn't eliminate their influence entirely.

6. Parallelization Challenges: While base models in bagging can be trained in parallel, coordinating and aggregating their results can introduce communication overhead in distributed computing environments.

Disadvantages of Bagging are -:

7. Equal Weighting: Bagging assigns equal weight to all base models when aggregating predictions. If some base models consistently perform better than others, their valuable insights may be diluted by the weaker models.

8. Not Suitable for All Data: Bagging may not always be the best choice for every dataset or problem. Other ensemble methods like boosting might perform better in certain situations, especially when dealing with highly imbalanced datasets.

9. Lack of Diversity: In some cases, bagging may not introduce enough diversity among base models, especially if the base model selection is limited. This can limit the overall improvement achieved through ensemble learning.

Boosting

- Boosting is a technique where multiple weak learners (models that perform slightly better than random guessing) are trained sequentially, with each new model focusing on the examples that the previous models found challenging. It assigns weights to the training instances, giving more importance to the ones that are misclassified.
- Boosting is an ensemble method that enables each member to learn from the preceding member's mistakes and make better predictions for the future. Unlike the bagging method, in boosting, all base learners (weak) are arranged in a sequential format so that they can learn from the mistakes of their preceding learner. Hence, in this way, all weak learners get turned into strong learners and make a better predictive model with significantly improved performance.

How it works:

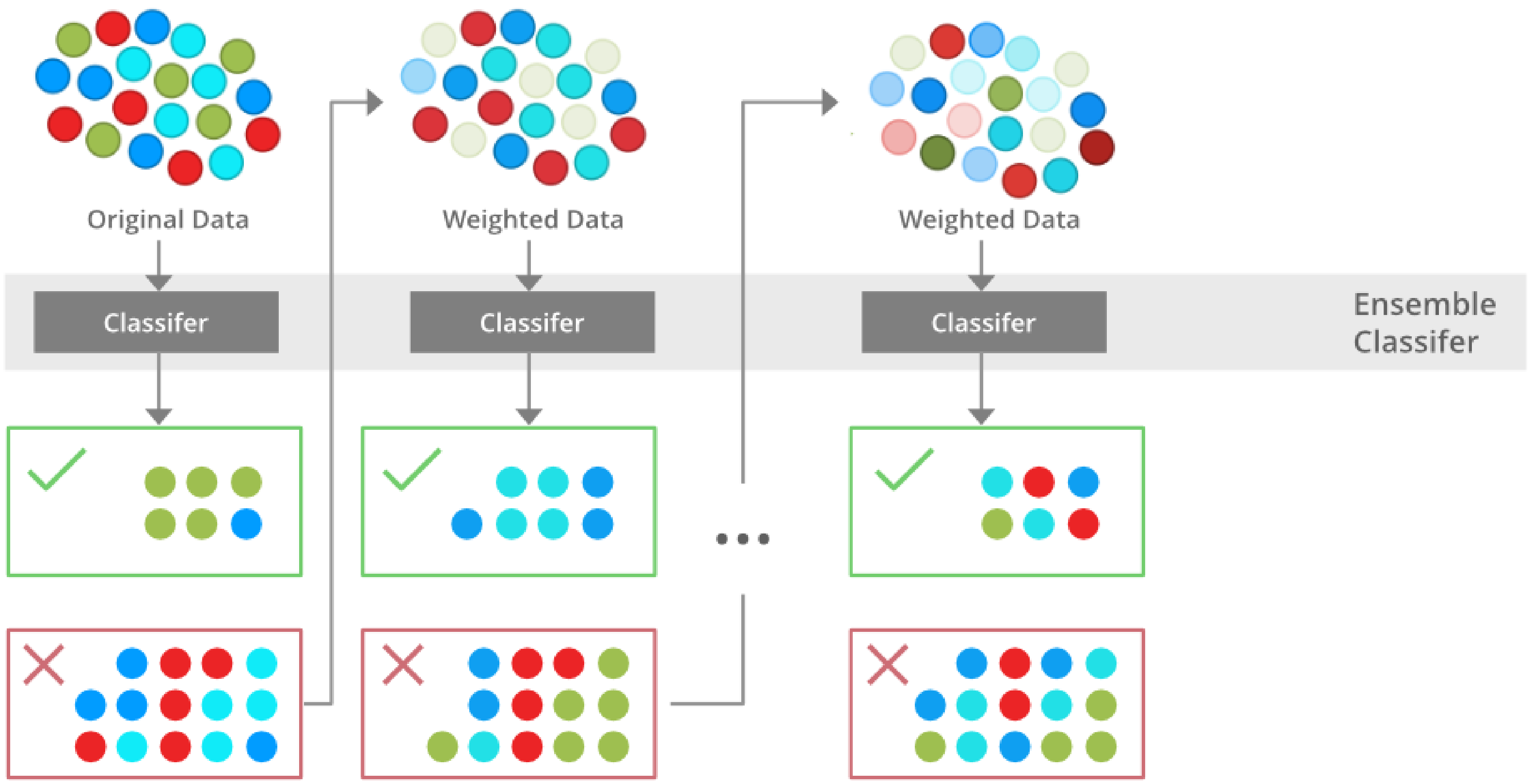
- ❑ Weak base models (e.g., decision stumps or shallow trees) are trained iteratively.
- ❑ At each iteration, more weight is given to misclassified instances from the previous iteration.
- ❑ The final prediction is a weighted sum of the individual model predictions.

Parul® University

Algorithm Boosting

```
1.Initialise the dataset and assign equal weight to each of the data point.
2.Provide this as input to the model and identify the wrongly classified data points.
3.Increase the weight of the wrongly classified data points.
4.if (got required results)
  Goto step 5
else
  Goto step 2
5.End
```

Boosting


Original Data
Weighted Data
Weighted Data
Classifer
Classifer
Classifer
Ensemble Classifer
...

Examples of Boosting Algorithms:

- **AdaBoost (Adaptive Boosting):** Uses weighted data points and focuses on misclassified samples.
- **Gradient Boosting (e.g., XGBoost, LightGBM):** Fits each new model to the residual errors of the previous model.
- **Gradient Boosted Decision Trees:** Sequentially adds decision trees to improve predictive performance.

Applications of Boosting

1. **Healthcare:** Boosting is used to lower errors in medical data predictions, such as predicting cardiovascular risk factors and cancer patient survival rates. For example, research shows that ensemble methods significantly improve the accuracy in identifying patients who could benefit from preventive treatment of cardiovascular disease while avoiding unnecessary treatment of others. Likewise, another study found that applying boosting to multiple genomics platforms can improve the prediction of cancer survival time.
2. **IT:** Gradient boosted regression trees are used in search engines for page rankings, while the Viola-Jones boosting algorithm is used for image retrieval. As noted by Cornell, boosted classifiers allow the computations to be stopped sooner when it's clear which direction a prediction is headed. A search engine can stop evaluating lower-ranked pages, while image scanners will only consider images containing the desired object.

Applications of Boosting

3. Finance: Boosting is used with deep learning models to automate critical tasks, including fraud detection, pricing analysis, and more. For example, boosting methods in credit card fraud detection and financial product pricing analysis improves the accuracy of analyzing massive data sets to minimize financial losses.

.

Advantages of Boosting

1. **Improved Accuracy –** Boosting can improve the accuracy of the model by combining several weak models' accuracies and averaging them for regression or voting over them for classification to increase the accuracy of the final model.
2. **Robustness to Overfitting –** Boosting can reduce the risk of overfitting by reweighting the inputs that are classified wrongly.
3. **Better handling of imbalanced data –** Boosting can handle the imbalance data by focusing more on the data points that are misclassified
4. **Better Interpretability –** Boosting can increase the interpretability of the model by breaking the model decision process into multiple processes.

Parul® University

Disadvantages of Boosting

5. Complex: It is complex to handle all models’ working and increase the data’s weight from every error. Algorithms are complicated to run in real time.

6. Dependency: Each successor model is dependent on the last model which may result in an error.

Similarities Between Bagging and Boosting

Bagging and Boosting, both being the commonly used methods, have a universal similarity of being classified as ensemble methods. Here we will explain the similarities between them.

1. Both are ensemble methods to get N learners from 1 learner.
2. Both generate several training data sets by random sampling.
3. Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
4. Both are good at reducing variance and provide higher stability.

DIGITAL LEARNING CONTENT

Differences Between Bagging and Boosting

NO	Bagging	Boosting
1.	The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
2.	Aim to decrease variance, not bias.	Aim to decrease bias, not variance.
3.	Each model receives equal weight.	Models are weighted according to their performance.
4.	Each model is built independently.	New models are influenced by the performance of previously built models.
5.	Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
6.	Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.
7.	If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
8.	In this base classifiers are trained parallelly.	In this base classifiers are trained sequentially.
9	Example: The Random forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques

Random Forest

- Random forest, merges the outputs of numerous decision trees to produce a single outcome making it suitable for both classification and regression tasks, its widespread popularity stems from its user-friendly nature and adaptability.
- The algorithm’s strength lies in its ability to handle complex datasets and mitigate overfitting, making it a valuable tool for various predictive tasks in machine learning, it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification

Random Forest Applications

- ❑ Customer churn prediction: Businesses can use random forests to predict which customers are likely to churn (cancel their service) so that they can take steps to retain them. For example, a telecom company might use a random forest model to identify customers who are using their phone less frequently or who have a history of late payments.
- ❑ Fraud detection: Random forests can be used to identify fraudulent transactions in real time. For example, a bank might use a random forest model to identify transactions that are made from unusual locations or that involve unusually large amounts of money.

Random Forest Applications

- ❑ Stock price prediction: Random forests can be used to predict future stock prices. However, it is important to note that stock price prediction is a very difficult task, and no model is ever going to be perfectly accurate.
- ❑ Medical diagnosis: These can be used to help doctors diagnose diseases. For example, a or might use a random forest model to help them diagnose a patient with cancer.
- ❑ Image recognition: It can be used to recognize objects in images. For example, a self-driving car might use a random forest model to identify pedestrians and other vehicles on the road.

Steps Involved in Random Forest Algorithm

Step 1: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.

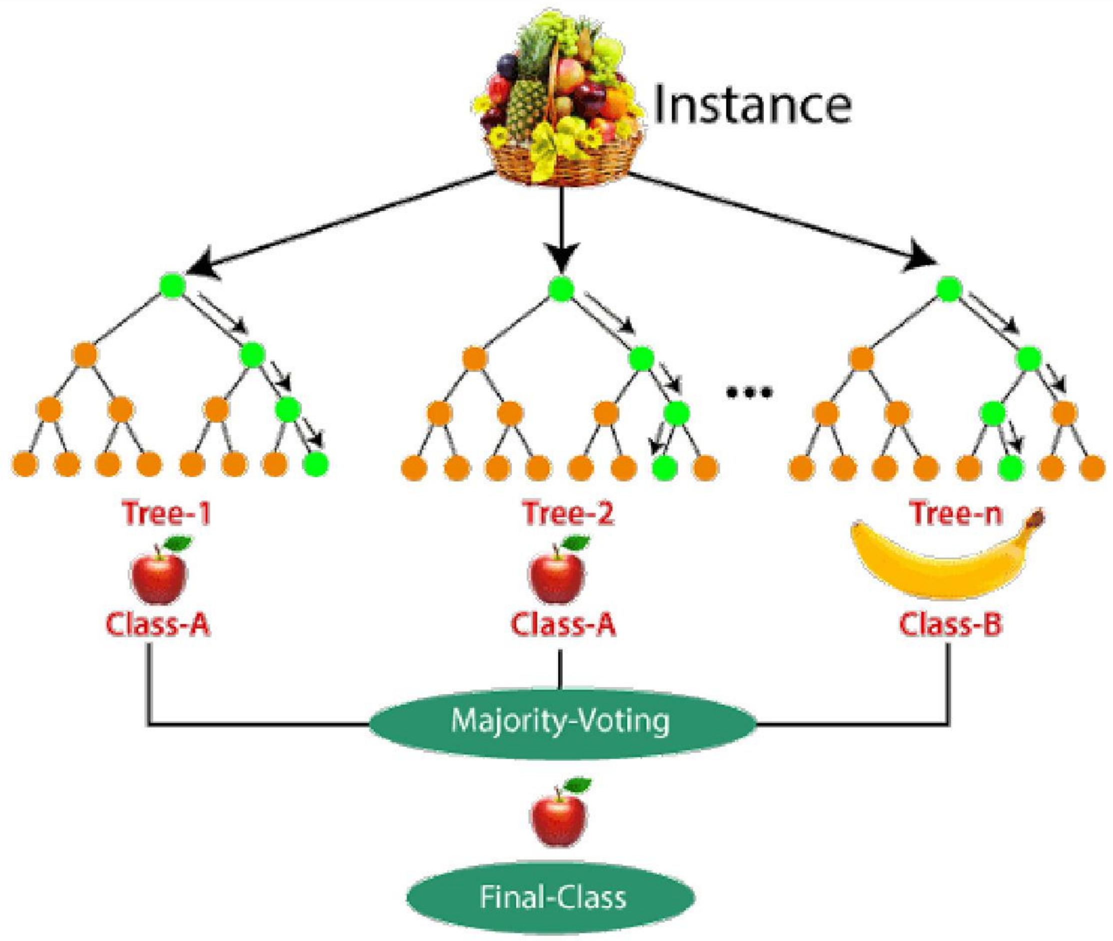
Random Forest Example

- ❑ Consider the fruit basket as the data as shown in the figure next slide. Now n number of samples are taken from the fruit basket, and an individual decision tree is constructed for each sample.
- ❑ Each decision tree will generate an output, as shown in the figure. The final output is considered based on majority voting.
- ❑ You can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

Parul® University

DIGITAL LEARNING CONTENT

Random Forest Example



Important Features of Random Forest

1. Diversity: Not all attributes/variables/features are considered while making an individual tree; each tree is different.
2. Immune to the curse of dimensionality: Since each tree does not consider all the features, the feature space is reduced.
3. Parallelization: Each tree is created independently out of different data and attributes.
4. Train-Test split: we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
5. Stability: Stability arises because the result is based on majority voting/ averaging.

Difference Between Decision Tree and Random Forest

Random forest is a collection of decision trees; still, there are a lot of differences in their behavior.

Decision trees	**Random Forest**
1. Decision trees normally suffer from the problem of overfitting if it’s allowed to grow without any control.	1. Random forests are created from subsets of data, and the final output is based on average or majority ranking; hence the problem of overfitting is taken care of.
2. A single decision tree is faster in computation.	2. It is comparatively slower.
3. When a data set with features is taken as input by a decision tree, it will formulate some rules to make predictions.	3. Random forest randomly selects observations, builds a decision tree, and takes the average result. It doesn’t use any set of formulas.

Important Hyperparameters in Random Forest

Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster. Hyperparameters to Increase the Predictive Power

1. n_estimators: Number of trees the algorithm builds before averaging the predictions.
2. max_features: Maximum number of features random forest considers splitting a node.
3. mini_sample_leaf: Determines the minimum number of leaves required to split an internal node.
4. criterion: How to split the node in each tree? (Entropy/Gini impurity/Log Loss)
5. max_leaf_nodes: Maximum leaf nodes in each tree

Hyperparameters to Increase the Speed

1. n_jobs: it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor, but if the value is -1, there is no limit.
2. random_state: controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and has been given the same hyperparameters and training data.
3. oob_score: OOB means out of the bag. It is a random forest cross-validation method. In this, one-third of the sample is not used to train the data; instead used to evaluate its performance. These samples are called out-of-bag samples.

Random Forest :Use Cases/Advantage/Disadvantage

Use Cases

This algorithm is widely used in E-commerce, banking, medicine, the stock market, etc. For example:

In the Banking industry, it can be used to find which customer will default on a loan.

Advantages

It can be used in classification and regression problems.

It solves the problem of overfitting as output is based on majority voting or averaging.

It performs well even if the data contains null/missing values.

Each decision tree created is independent of the other; thus, it shows the property of parallelization.

It is highly stable as the average answers given by a large number of trees are taken.

Random Forest :Use Cases/Advantage/Disadvantage

Disadvantages

Random forest is highly complex compared to decision trees, where decisions can be made by following the path of the tree.

Training time is more than other models due to its complexity. Whenever it has to make a prediction, each decision tree has to generate output for the given input data.

Ada Boost

- AdaBoost, which stands for Adaptive Boosting, is a machine learning ensemble method used for classification and regression tasks.
- It was introduced by Yoav Freund and Robert Schapire in 1996.
- AdaBoost combines the predictions from multiple weak learners (typically decision trees or other simple models) to create a strong ensemble model.

How AdaBoost works

1. **Initialization**: Each data point is assigned an equal weight at the beginning.

2. **Iterative Training**: AdaBoost fits a sequence of weak learners to the data. A weak learner is a simple model that performs slightly better than random guessing, such as a decision stump (a one-level decision tree).

3. **Weighted Learning**: After each weak learner is trained, AdaBoost assigns higher weights to data points that were misclassified by the previous learner. This allows subsequent learners to focus more on the samples that are difficult to classify correctly.

4. **Combine Weak Learners**: AdaBoost combines the predictions of all the weak learners by giving each learner a weight based on its accuracy. More accurate learners are given higher weights, and their predictions carry more influence in the final ensemble.

5. **Final Prediction**: To make a prediction for a new data point, AdaBoost takes a weighted vote of the predictions from all the weak learners. The final prediction is determined by majority voting (for classification tasks) or a weighted sum (for regression tasks).

How AdaBoost works

The key idea behind AdaBoost is that by giving more weight to the misclassified samples in each iteration, it focuses the subsequent learners on the examples that are harder to classify. This adaptability allows AdaBoost to create a strong ensemble model that often outperforms individual weak learners.

AdaBoost is a powerful technique and is used in various applications, including face detection, object recognition, and text classification. However, it can be sensitive to noisy data and outliers, so preprocessing and careful selection of weak learners are important considerations when using AdaBoost. Additionally, it's essential to choose an appropriate number of iterations (weak learners) to avoid overfitting.

XGBoost

It is a popular and powerful machine learning algorithm that falls under the category of gradient boosting methods. It is designed for both regression and classification tasks and is known for its efficiency and performance.

Parul® University

DIGITAL LEARNING CONTENT

key features and characteristics of XGBoost

1. **Gradient Boosting**: Like AdaBoost, XGBoost is an ensemble method. It builds a strong predictive model by combining the predictions of multiple weaker models (typically decision trees). However, XGBoost uses a gradient boosting framework, which means it optimizes a differentiable loss function.

2. **Regularization**: XGBoost includes L1 (Lasso) and L2 (Ridge) regularization terms in its objective function to prevent overfitting. Regularization helps control the complexity of the model and improve generalization.

3. **Parallel Processing**: XGBoost is designed to be highly scalable and efficient. It can take advantage of parallel processing and is faster than many other gradient boosting implementations.

4. **Handling Missing Values**: XGBoost has built-in support for handling missing values, which means you don't need to preprocess your data to fill in missing values before using the algorithm.

5. **Cross-Validation**: It supports built-in cross-validation to help you tune hyperparameters and assess model performance effectively.

key features and characteristics of XGBoost

6. **Tree Pruning**: XGBoost incorporates tree pruning techniques to reduce tree complexity and combat overfitting.

7. **Feature Importance**: XGBoost provides feature importance scores, allowing you to understand which features are the most influential in making predictions.

8. **Customizable Objective Functions**: You can define custom loss functions for specific tasks, which makes it flexible for a wide range of machine learning problems.

9. **Availability in Multiple Languages**: XGBoost is available in multiple programming languages, including Python, R, Java, and others.

10. **Wide Adoption**: XGBoost has been widely adopted in machine learning competitions (such as Kaggle) and real-world applications due to its effectiveness and versatility.

Parul® University

key features and characteristics of XGBoost

To use XGBoost, you typically install the XGBoost library for your programming language of choice (e.g., xgboost in Python) and use its functions and classes to train and evaluate models.

Hyperparameter tuning is often necessary to get the best performance, and cross-validation can help with this process.