### 

**Assignment-02**

**on**

**“Random Forest”**

**Course Title: Machine Learning.**

**Course Code: CSE475**

**Section: 02**

**Submitted by:**

**Md Tariqul Islam**

**ID:2021-1-60-054**

**Submitted to:**

**Dr.Raihan ul Islam(DRUI)**

**Associate Professor**

**Department of Computer Science and Engineering**

**East West University**

**Submission Date: 15/9/2024**

### **Answer to the Question no:1**

Bootstrapping and Bagging are two key techniques used in the construction of Random Forests, a learning method for classification and regression. These techniques contribute to the robustness and performance of the model.

#### **1. Bootstrapping**

Bootstrapping is a statistical technique that involves sampling with replacement. In Random Forests, bootstrapping is used to create multiple datasets from the original training dataset.

* From the original dataset containing NNN samples, a new dataset is created by randomly selecting NNN samples with replacement.
* This means some samples from the original dataset may appear multiple times in the bootstrap sample, while others may not appear at all.
* Each bootstrap sample is then used to train a single decision tree in the Random Forest.

Bootstrapping is used to create diversity among the individual decision trees in the forest. Since each tree is trained on a slightly different dataset, they will learn different patterns and make different errors.

#### **2. Bagging (Bootstrap Aggregating)**

Bagging is a technique that combines the predictions of multiple models to produce a single, aggregated prediction. In Random Forests, bagging is applied as follows:

* After training several decision trees on different bootstrap samples, each tree makes a prediction for a given input.
* For classification, the final prediction of the Random Forest is determined by a majority vote.
* For regression, the final prediction is the average of the predictions made by all the trees.

#### **Contribution to Robustness and Performance**

* **Reducing Variance:**

The benefit of bootstrapping and bagging in Random Forests is the reduction of variance in the model’s predictions. Since each tree is trained on a different subset of data, it is likely to make different errors. By averaging these predictions or taking a majority vote, Random Forests reduce the impact of any one tree’s errors, leading to a more stable and reliable model.

* **Improving Generalization:**

Bootstrapping introduces randomness in the training process, which helps in creating a diverse set of decision trees. This diversity reduces the risk of overfitting to the training data, improving the model's ability to generalize to new, unseen data.

* **Handling Overfitting:**

Individual decision trees are prone to overfitting, especially when they are deep and complex. By combining many such trees through bagging, the Random Forest reduces the likelihood of overfitting to any specific noise or peculiarities in the training data.

* **Robustness to Noise:**

Random Forests aggregate predictions from multiple trees, they are less sensitive to outliers or noise in the dataset. Outliers or noisy data points may significantly affect some individual trees, but their impact is diluted when the predictions are averaged or voted upon by all trees in the ensemble.

### **Answer to the Question no:2**

Random Forests offer several advantages over single decision trees, especially in reducing overfitting and improving generalization to new data.

**Random Forests achieve these benefits:**

#### **1. Reduced Overfitting**

* **Decision Trees Tend to Overfit:**

A single decision tree can easily overfit the training data, especially when it is deep or has many levels. It tends to learn not just the underlying patterns but also the noise and outliers in the data, leading to poor performance on unseen data.

* **Random Forests Minimize Overfitting:**

Random Forests reduce the risk of overfitting by averaging the predictions of multiple trees. Each tree is trained on a different subset of the data using bootstrapping and on different subsets of features by randomly selecting a subset of features at each split. This randomization makes each tree in the forest slightly different from the others, reducing the chance that they will all overfit in the same way.

By combining the outputs of multiple trees, Random Forests smooth out the predictions, which results in a model that is less sensitive to noise and less likely to overfit.

#### **2. Improved Generalization**

* **Decision Trees and Generalization:**

A single decision tree might generalize poorly because it can become overly complex and specific to the training data. As a result, it may fail to perform well on new data that it has never seen before.

* **Random Forests Enhance Generalization:**

Random Forests improve generalization by leveraging the collective power of multiple decision trees. Since each tree is built on a different sample and considers a different subset of features, the model captures a broader range of patterns in the data.

The use of a majority vote for classification or averaging for regression helps the model make more balanced decisions that are less likely to be skewed by anomalies in the training data.

#### **3. Increased Stability and Robustness**

* **Robust to Noisy Data:**

Random Forests are less affected by noise in the training data. While a single decision tree might be significantly influenced by outliers or noisy instances, the effect of such anomalies is diminished in a Random Forest because they are less likely to be selected in multiple bootstrap samples.

* **Stable Predictions:**

Random Forests aggregate the results from multiple trees, their predictions tend to be more stable. Small changes in the training data are unlikely to lead to drastic changes in the final output, unlike in a single decision tree, where a small change can significantly alter the structure and predictions of the tree.

#### **4. Better Handling of High-Dimensional Data**

* **Feature Selection at Each Split:**

Random Forests randomly select a subset of features at each node to determine the best split, which makes them effective in high-dimensional datasets. This random selection process reduces the correlation between the trees and ensures that the model does not rely too heavily on any particular set of features, which helps improve its generalization capability.

### **Answer to the Question no:3**

"Curse of dimensionality" is the problems that arise while working with high-dimensional data, such as datasets with a large number of features. As the number of dimensions increases, the amount of data needed to reliably model the relationships between these features grows exponentially, which lead the issues: overfitting and poor generalization.

Random Forests are immune to these problems.

1. **Random Selection of Features at Each Split:**

Random Forests effective in high-dimensional spaces is the random selection of features at each decision tree split. Instead of considering all available features when finding the best split, Random Forests randomly select a subset of features to evaluate.

This randomization reduces the likelihood of overfitting because it prevents the model from relying too heavily on any one feature or a small group of features. In high-dimensional datasets, many features may be irrelevant or redundant, and by selecting only a subset, Random Forests focus on the most important and diverse features, leading to better generalization.

1. **Reduced Complexity and Overfitting:**

In high-dimensional data, a single decision tree might easily overfit by creating overly complex rules based on all the features available. Random Forests build a more generalized model, by randomly selecting features for each tree. Each tree gets a unique view of the data, which means the Random Forest is less likely to overfit compared to a single decision tree.

1. **Increased Diversity Among Trees:**

Because each tree is trained on a different subset of features, the trees in a Random Forest are more diverse. This diversity is crucial because it means that each tree will capture different patterns in the data. When the predictions from these diverse trees are combined, they balance each other out, leading to a more robust model that performs well across different datasets.

1. **Handles High-Dimensional Data Efficiently:**

Random Forests are good at handling datasets with many features. In high-dimensional spaces, many machine learning models struggle because they try to learn from every feature. Random Forests, on the other hand, simplify the problem by only looking at a subset of features at each split. This reduces the complexity of each tree, speeds up the training process, and captures the important patterns in the data.