Practical no. 4

Name : Saloni Vishwakarma
Roll no. - C1-13
Subject - Artificial Intelligence and Cyber Security Lab

Aim : To implement a machine learning classification model using Decision Tree and Random Forest Algorithms

Decision Tree Algorithm :

A Decision Tree is a popular supervised machine learning algorithm used for both classification and regression tasks. It's a tree-like model that makes decisions by splitting the dataset into subsets based on the values of input features. These splits are determined by a set of rules that are learned from the training data.

1. Initialization: Start with the entire dataset as the root node of the tree.

2. Feature Selection: Choose the best feature to split the data, typically based on information gain, impurity reduction, or other criteria.

3. Data Splitting: Divide the dataset into subsets based on the chosen feature's values, creating child nodes.

4. Recursive Process: Recursively apply the above steps to each child node until a stopping condition is met (e.g., maximum depth or minimum samples per leaf).

5. Prediction: To make predictions, follow the decision rules in the tree from the root node down to a leaf node.

7. Advantages: Easy to interpret, handles both categorical and numerical data, and no need for feature scaling.

8. Disadvantages:Prone to overfitting, sensitive to data variations, and may not work well with complex relationships.

9. Variants: Popular variants include Random Forests, Gradient Boosted Trees, and others, which improve upon Decision Trees' weaknesses while preserving their strengths.

Gini Index : Gini Index measures the impurity of a set of examples. It ranges from 0 to 1, where 0 represents the purest node (all elements belong to the same class), and 1 represents the most impure node (elements are evenly distributed among various classes).

Formula:
Gini(D) = 1 - sum{i=1}to{c} (pi)^2

where (pi) is the probability of choosing a class (i) from the dataset (D).

Example:Consider a dataset with two classes, A and B. If the dataset is pure (all instances are of one class), the Gini Index is 0. If it is evenly split between A and B, the Gini Index is 0.5.

Entropy: Entropy measures the impurity or disorder of a set of examples. It is commonly used in the context of decision trees.

Formula :

$$E = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Example:
Similar to Gini Index, if a dataset is pure, the entropy is 0. If it is evenly split between classes, the entropy is at its maximum.

Information Gain:
Information Gain measures the effectiveness of an attribute in classifying the dataset. It is the difference between the entropy or Gini Index before and after the split.

Formula:
Information Gain = Entropy before splitting - Entropy after splitting

Example:
Let's say we split a dataset based on an attribute, and the resulting subsets have lower entropy compared to the original dataset. The Information Gain is high.

Random Forest Algorithm:
Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

How Random Forest Works:
1. Bootstrapped Sampling:Randomly select subsets of the training data (with replacement) to train each tree.

2. Feature Randomness: At each node of the tree, consider only a random subset of features for splitting.

3. Voting or Averaging:For classification, each tree "votes" for a class, and the majority class is the predicted class. For regression, the predictions of all trees are averaged.

Advantages:
- Reduces overfitting.
- Handles missing values well.
- Provides feature importance.

Example:
Imagine you're in a forest, and you want to decide whether to go left or right based on various environmental factors. You ask different trees (each representing a decision based on different factors), and the majority advice is your decision.
So, that's a nutshell on decision trees, impurity measures, information gain, and the powerful Random Forest algorithm. If you have specific questions or want more depth on any aspect, feel free to ask!