UNIT-1 (INTRODUCTION)

Subject: Machine Learning Techniques
Subject code: BCS-055
Session: 2024-25 (ODD)
Faculty Name: Utkarsh Yashvardhan
Section: D and H2

Inductive Bias:

- **Definition:** Inductive bias refers to the assumptions a learning algorithm makes to generalize beyond the training data. In other words, it's the "starting preference" that guides the algorithm on which patterns to prioritize.

- **Example:** If you're predicting whether a day is sunny based on past data, you might assume that sunny days often follow other sunny days. This assumption is the inductive bias, helping the algorithm make guesses for new, unseen data.

Inductive Inference with Decision Trees:

- **Definition:** Inductive inference is the process of making general conclusions from specific examples, aiming to predict outcomes for new situations. With decision trees, this means creating a tree structure based on past data that lets you make predictions for future data.

- **Example:** If you're creating a decision tree to classify animals, you use characteristics (like whether it has feathers or fur) from known examples. The tree then helps you infer (or predict) classifications for new animals.

Entropy and Information Theory:

- **Definition:** In information theory, entropy measures the "uncertainty" or "disorder" in data. High entropy means the data is highly unpredictable, while low entropy means it's more predictable.
- **Example:** If you have a set of weather days where half are sunny and half are rainy, entropy is high because the weather is unpredictable. But if 90% are sunny and only 10% are rainy, entropy is low, as sunny days are much more predictable.

- **In Decision Trees:** Entropy is used to find the best way to split the data. A good split reduces entropy, creating subsets of data that are more "pure" or homogeneous (more predictable).

Information Gain:

- **Definition:** Information gain measures how much "information" (or predictability) a particular feature adds when it's used to split the data. Higher information gain means the feature does a better job of reducing uncertainty.

- **Formula:** Information Gain = Initial Entropy - Entropy after split.
- **Example:** In a decision tree, suppose you're trying to classify animals. If the "feathers" attribute creates groups of animals with high purity (birds vs. mammals), it has high information gain, so it's a good choice for splitting.

ID3 Algorithm:
- **Definition:** ID3 is a common algorithm used to build decision trees. It constructs the tree by choosing attributes that have the highest information gain at each step, which helps build a tree that can accurately classify new data.

- **Process:**
  1. Start with all data at the root.
  2. Calculate information gain for each attribute.
  3. Choose the attribute with the highest information gain for the first split.
  4. Repeat the process for each subset of data until the data is pure (contains examples of only one class) or you've run out of attributes.

Issues in Decision Tree Learning:
- **Overfitting:** Decision trees can sometimes fit the training data too closely, capturing noise instead of the general trend. This makes them less accurate on new data.

- **Underfitting:** Simplifying the tree too much can lead to underfitting, where the model is too simple and fails to capture important patterns in the data.

- **Bias and Variance:** Decision trees can have high variance (sensitivity to small changes in data) due to their flexibility, making them unstable on new data.

- **Data Requirements:** Decision trees need a large amount of data to accurately capture complex patterns, especially if the data has many attributes.

- **Attribute Selection:** In some cases, the decision tree may struggle to find meaningful splits if the attributes do not have clear predictive power.

---

k-Nearest Neighbors (k-NN) Learning:

- **Definition:** k-Nearest Neighbors is a type of instance-based learning where the algorithm classifies or predicts outcomes for new data points by looking at the "k" closest examples (neighbors) in the training data.

- **How It Works:** To make a prediction, k-NN finds the k data points closest to the new data point (based on distance, like Euclidean distance) and "votes" based on the outcomes of these neighbors. For classification, the most common class among neighbors is chosen, and for regression, the average outcome of neighbors is used.

- **Example:** Suppose you want to classify a new fruit based on its sweetness and size. k-NN would find the k closest fruits in the data and classify the new fruit based on the most common type among these neighbors.

Locally Weighted Regression (LWR):
- **Definition:** Locally Weighted Regression is a type of regression that focuses more on nearby data points when predicting the outcome for a new point. It is a form of local learning, where each prediction is made using only a local

subset of data that is most relevant to the point being predicted.

- **How It Works:** Instead of fitting a single model to all the data, LWR creates a unique model for each prediction. It assigns higher weights to points that are closer to the new point and fits a regression line (or curve) that best represents the nearby data.

- **Example:** In predicting house prices based on location, size, and other features, LWR would prioritize houses that are closer to the house of interest when making a price prediction, thus reflecting more accurate local trends.

Radial Basis Function (RBF) Networks:

- **Definition:** An RBF network is a type of neural network that uses radial basis functions (typically Gaussian functions) as activation functions to make predictions. It’s particularly useful for problems where data points are grouped in clusters.

- **How It Works:** RBF networks have three layers: an input layer, a hidden layer of RBF units, and an output layer. Each RBF unit calculates how close the input is to a “center” (usually a representative data point) and produces an output based on this distance. The output layer combines these results to make a final prediction.

- **Example:** In handwriting recognition, RBF networks can identify individual characters by measuring how close the strokes of each letter are to the stored prototypes (centers) of each letter shape.

Case-Based Learning:

- **Definition**: Case-Based Learning (CBL), also known as Case-Based Reasoning (CBR), is an approach where the algorithm solves new problems by referencing past cases (examples) and adapting their solutions. Instead of generalizing, it focuses on finding a relevant example from memory.

- **How It Works:** When a new problem arises, the model searches its library of cases to find the most similar one(s). It then uses the solution to this similar case as a basis for solving the new problem. CBL is especially useful in domains with well-defined cases, like medical diagnosis or customer support.
- **Example:** In medical diagnosis, if a doctor encounters a patient with symptoms that match a previous patient's case, they may use the previous case's treatment as a reference, adjusting it based on the current patient's specifics.

Summary of Differences:

- **k-Nearest Neighbors:** Uses nearby points to classify or predict outcomes by majority or average.

- **Locally Weighted Regression:** Fits a unique model for each prediction, focusing more on nearby points.

- **Radial Basis Function Networks:** Uses radial functions for predictions, focusing on distance from central points in a neural network structure.

- **Case-Based Learning:** Solves new problems by adapting solutions from similar past cases.