UNIT IV - TREE BASED AND PROBABILISTIC MODEL

Tree-Based Model — Decision Tree ✓ / Random forest

Decision Tree :- Supervised learning algorithm used for classification & regression both.
↳ It is a graphical representation of all possible solutions to a decision.

Decision tree terminologies :-
① Leaf Node - Node that cannot be further divided into child nodes (final o/p)
② Root Node - Starting Node & gets further divided into 2/more homogenous sets.
③ Child Node - Node formed by splitting of Root Node (Parent Node).
④ Branch / Subtree - Connection formed b/w nodes representing specific decision path.
⑤ Pruning - Removing unwanted branches/nodes from tree
⑥ Splitting - Dividing root node into different parts.
⑦ Impurity - Measure of how much mixed the target classes are given in node

Impurity Measures :- These measures help to identify best features for creating branches that lead to more accurate predictions.

① Gini Index / Gini Impurity — measures inequality in sample.
↳ Measures impurity of a node / how mixed classes are.
↳ It tells us how likely it is that a randomly choosen item from dataset will be incorrectly classified.
↳ Gini Index = 0 ⇒ all items belong to one class ⇒ perfectly pure
> 0 ⇒ more mixing of classes ⇒ more impure.

Goal is to - minimize Gini Index at each split.

$$Gini(D) = 1 - \sum p_i^2$$

$p_i$ — probability of data point in class 'i'

② Entropy
↳ measures the disorder/uncertainity in a dataset.
↳ It tells the impurity in a node by considering distribution of a classes.
↳ Entropy = 0 ⇒ items belong to 1 class ⇒ perfectly pure
> 0 ⇒ more disorder ⇒ more impure.

$$H(D) = -\sum (p_i \cdot \log_2(p_i))$$

Goal is to - minimize entropy

③ Information Gain
↳ measures how much uncertainty (or entropy) is reduced when data is split based on specific feature.
↳ It tells how much better we can predict the outcome, after splitting the data based on a feature. It shows how much useful info we can gain from the split.

$$IG = Entropy(parent) - \sum (p_i \cdot Entropy(child))$$

Goal is to - ↑ IG to get ↑ info, by splitting

Tree Pruning :- technique to improve performance by reducing overfitting
↳ Overfitting occurs when a model learns training data too well, incl noise & outliers, which makes it perform poorly on new/unseen data
↳ Pruning helps simplifying tree structure, making it generalizable & easier to interpret.

① Pre-Pruning (Early Stop) :- stops growth of tree, before it gets complex
↳ It sets criteria such as, max-depth / min samples per leaf to prevent further splits. Eg: 18+ can vote (why you need blood grp, area, height, weight, only age is required)

② Post-Pruning :- fully growing tree first and then removing branches that does not contribute to model's accuracy.

ALGORITMS:

①ID3 (Iterative Dichotomiser 3) Algorithm

Step 1 - Determine the Root of the Tree
typically the attribute that provides highest info gain
Step 2 - Calculate Entropy for the Classes.
Step 3 - Calculate entropy after split for each Attribute.
Step 4 - Calculate Information Gain for each Split.
Step 5 - Perform first split.
Step 6 - Perform further splits
repeat step 2 to 5 for each subset.
Step 7 - Complete the Decision Tree.

② C4.5 Algorithm
↳ developed by Ross Quinlan, as an extension to ID3 Algo.
↳ Dissadvantages of ID3 → Overfitting
→ Handling continuous attributes
→ Does not manage missing values
→ Lack of Pruning.
→ Worked for binary trees only.

C4.5 Algorithm uses "Gain Ratio", which adjusts IG by accounting for number of splits made.

$$\boxed{\text{Gain Ratio} = \frac{IG}{\text{Intrinsic Info}}}$$

Advantages of Decision Tree
↳ Easy to understand
↳ Works with diff. datatypes
↳ No need for scaling
↳ Resistant to Outliers
↳ Handles Missing Values
↳ Flexible

↳ Automatically selects imp features
↳ can do both Classification & Regression
↳ Handles multiple outputs
↳ Works well with other methods (like Random F)

## Disadvantages of Decision Tree

↳ overfitting
↳ Sensitive to changes
↳ Limited Complexity Handling
↳ Greedy Approach
↳ Bias towards complex features
↳ Need for Pruning
↳ Poor performance on Imbalanced data.

## Probabilistic Model

These models allows us to make predictions & inferences based on uncertainty

## Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Event A → probability we're trying to find
Event B → already happened/occured

## Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

P(A|B) → Posterior probability
P(B|A) → Likelihood
P(A) → Prior probability of proposition
P(B) → ~~Marginal likelihood~~ prior probability of evidence

## ** Naive Bayes Classifier / Naive Bayes Algorithm.

Step 1: Calculate Prior Probability
Step 2: find likelihood probability
Step 3: Use Bayes' formula
Step 4: Choose the class with Higher Probability.

Eg. Step 1 - 70% spam 80% not spam
P(spam) = 0.7
P(not spam) = 0.3

Step 2 - word "free".
P(free|spam) = 60 out of 70 = 0.857
P(free|not spam) = 5 out of 30 = 0.167

Step 3 -
$$P(\text{class}|\text{feature}) = \frac{P(\text{feature}|\text{Class}) \cdot P(\text{Class})}{P(\text{features})}$$

Step 4 - P(spam|free) = 0.8
P(not spam|free) = 0.2

∴ Email is spam.

↳ zero probability problem
when p=0, still it calculates which can be overcome through "smoothing techniques" such as Laplace estimation.

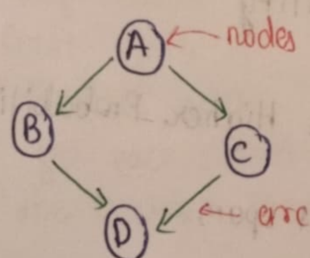
Advantages of NBC
↳ Simplicity
↳ Fast training & prediction
↳ Works well with large/small dataset
↳ Handles High Dimension Data
↳ assumes that features are independent
↳ Versatile

Disadvantages
↳ independence assumption
↳ sensitive to irrelevant features
↳ not suitable for all datatypes
↳ cannot capture complex relationships betn features

Application of NBC
↳ Spam / Not spam
↳ Sentiment Analysis (+ve/-ve/neutral) (feedback)
↳ Recommendation Systems.
↳ Infected / Not infected
↳ Weather prediction
↳ Face Recognition

Bayesian Network for Learning & Inferencing
↳ powerful tools used to represent and reason about uncertain knowledge.
↳ type of probabilistic graphical model that uses DAGs (directed acyclic graphs) to illustrate the relationships between variables.
↳ Each node represents a variable, & edges represent conditional dependencies between these variables.

Structure / Components
Nodes → each node represent random variable, which can be discrete (yes/no) or continuous
Edges → represents conditional dependencies between variables.
CPDT (Conditional Probability ~~Distribution~~ Tables) → Each node has an CPT that quantifies relationship betn node & its parent node.




How it works.
① Structure representation :- represented in DAG ie no cycles & you cannot return to a node once you have moved away from it.
② Joint Probability Distribution :- can be calculated using chain rule
$$P(X_1, X_2 \ldots X_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(x_i))$$
ie. joint probability is the product of conditional probabilities of each variable given its parent
③ Inference :- involves updating beliefs about certain variables based on evidences from other observed variables.
Eg. If you observe symptom in patient, you can use network to infer the probabilities of certain various diseases

## Learning in Bayesian Network.

① Parameter Learning -
Estimating CPT for each variable
2 common methods → Max. Likelihood Estimation
→ Bayesian Estimation

② Structure learning -
involves determining the network structure ie, figuring out how variables are connected to each other.

2 methods → Score-based
→ Constraint based

## Inference in Bayesian Network

↳ deriving new info from known data using structure & parameters of Bayesian Network

① Exact Inference
→ gives accurate/exact/precise probabilities.
→ works best for smaller networks.

i) Variable elimination
removes variables one at a time to simplify calculations
eg. picking up mess to clean room

ii) Belief Propogation
sends msg between connected nodes in the network
each node updates its belief's
eg. sharing info among friends

② Approximate Inference.
→ gives estimated probabilities when exact calculations are too complicated
→ works best for large networks

i) Monte Carlo Simulation

ii) Loopy Belief Propogation

---

Minority class - category in classification problem that has fewer instances compared to other categories

Eg. Medical diagnosis dataset

Minority class → Rare disease
Majority class → Healthy ppl.