**Machine Learning (MTCS 031)**

**UNIT-III**

**Syllabus:**

***EVALUATING HYPOTHESES*** – Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms;

***BAYESIAN LEARNING*** – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm;

**PYQ:**

**7 Marks:**

1. How is Naïve Bayesian Classifier different from Bayesian Classifier? Explain.
2. Explain the concept of Bayes theorem with an example.
3. Explain Bayesian belief network and conditional independence with example.
4. Explain the k-Means Algorithm with an example.

**2 Marks:**

1. Differentiate between Supervised and Unsupervised learning.
2. Describe accuracy with a suitable example.
3. Explain sample complexity of a Learning Problem?

**Bayes' Theorem Formula**

The theorem is expressed mathematically as:

P(A∣B)=P(B∣A)⋅P(A)

P(B)

Where:

* P(A∣B) is the **posterior probability**: the probability of event A occurring given that event B has occurred.
* P(B∣A) is the **likelihood**: the probability of event B occurring given that event A has occurred.
* P(A) is the **prior probability**: the initial probability of event A occurring, before considering event B.
* P(B) is the **marginal probability**: the total probability of event B occurring, considering all possible scenarios.

**Applications:**

* **Text Classification**: Spam detection, sentiment analysis, document categorization.
* **Medical Diagnosis**: Predicting the likelihood of diseases based on symptoms.
* **Recommendation Systems**: Filtering content based on user preferences.

**Naive Bayes Classifier:**

* **Definition**: The Naive Bayes classifier is a machine learning algorithm that applies Bayes' Theorem with the assumption of independence between every pair of features.
* **Assumption**: It assumes that all features (or attributes) are independent of each other given the class label, which is why it's called "naive."
* **Usage**: The Naive Bayes classifier uses Bayes' Theorem to predict the probability that a given data point belongs to a particular class based on the features.

**Bayesian belief networks**

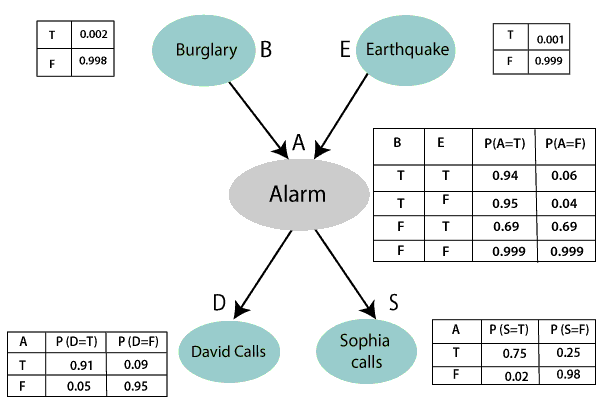
* **Bayesian Belief Networks (BBNs)**, also known as **Bayesian Networks** or **Belief Networks**, are graphical models that represent probabilistic relationships among a set of variables. These networks are used to model the uncertainty in complex systems by combining the principles of probability theory with graph theory.
* We can use a trained Bayesian network for classification.
* There are two components that define a Bayesian belief network :

***A. Directed acyclic graph*** :

* + Each node in a directed acyclic graph represents a random variable.
  + These variable may be discrete or continuous valued.
  + These variables may correspond to the actual attribute given in the data.

***B. Conditional Probability Table***:

* A CPT is a table that lists all possible combinations of values for a node and its parent nodes, along with the corresponding conditional probabilities. The table is used to specify the probability distribution of the node given its parents.



**Conditional Independence**

Conditional independence in the context of Bayesian Networks refers to a situation where two variables are independent of each other given knowledge about a third variable (or a set of variables). In simpler terms, if knowing the value of one variable does not change the probability distribution of another variable, given a third variable, they are conditionally independent.

**Formal Definition**: Variables A and B are conditionally independent given C if:

**P(A,B∣C)=P(A∣C)⋅P(B∣C)**

**Naive Bayes vs. Bayesian Network**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Naive Bayes** | **Bayesian Network** |
| Assumption | Features are conditionally independent given the class label | Allows for conditional dependencies between features |
| Complexity | Simpler model, computationally efficient | More complex model, can represent complex relationships |
| Structure | No explicit graphical structure | Uses a directed acyclic graph (DAG) to represent dependencies |
| Inference | Based on Bayes' theorem with independence assumption | Can use more complex inference algorithms (e.g., belief propagation) |
| Applicability | Suitable for problems with many features and limited computational resources | Appropriate for problems with complex relationships between features |
| Example | Spam filtering, text classification | Medical diagnosis, fault diagnosis |
| Data Requirements | Requires less data due to independence assumption. | Requires more data to accurately estimate joint probabilities. |
| Performance | Can perform well even if independence assumption is violated, especially in text classification. | Potentially more accurate if sufficient data and valid joint distribution. |
| Scalability | Scales well with a large number of features. | Less scalable due to the need to compute joint probabilities. |
| Ease of Implementation | |  | | --- | | Relatively easy to implement and understand. |  |  | | --- | |  | | More complex to implement due to comprehensive probability calculations. |

**Supervised vs Unsupervised Learning:**

|  |  |  |
| --- | --- | --- |
| Aspect | Supervised Learning | Unsupervised Learning |
| Definition | Learning from labeled data where the output is known. | Learning from unlabeled data without predefined outputs. |
| Training Data | Requires labeled data with input-output pairs. | Uses only input data, with no labeled responses. |
| Output | Predicts outcomes or labels for new data points. | Discovers hidden patterns or groupings in data. |
| Common Algorithms | Linear regression, logistic regression, decision trees, support vector machines, neural networks. | k-means clustering, hierarchical clustering, principal component analysis (PCA), association rules. |
| Goal | To map inputs to outputs based on example data. | To find structure, patterns, or relationships in the data. |
| Applications | Classification (e.g., email spam detection), regression (e.g., predicting house prices). | Clustering (e.g., customer segmentation), dimensionality reduction (e.g., data compression). |
| Data Dependency | Relies heavily on the quality and quantity of labeled data. | Can work with large amounts of unlabeled data. |
| Examples | Email classification, speech recognition, medical diagnosis. | Market basket analysis, anomaly detection, exploratory data analysis. |

**Accuracy**

**Accuracy** is a common metric used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of the total instances in the dataset.

**Accuracy Formula**

Accuracy is calculated using the formula:

Where:

* **True Positives (TP)**: Instances correctly predicted as positive.
* **True Negatives (TN)**: Instances correctly predicted as negative.
* **False Positives (FP)**: Instances incorrectly predicted as positive.
* **False Negatives (FN)**: Instances incorrectly predicted as negative.

**Example of Accuracy Calculation**

Let's consider a binary classification problem where we are classifying whether emails are "Spam" or "Not Spam" using a spam detection model.

Suppose we have a dataset with 100 emails, and the classification model makes the following predictions:

* **True Positives (TP)**: 40 emails correctly identified as Spam.
* **True Negatives (TN)**: 50 emails correctly identified as Not Spam.
* **False Positives (FP)**: 5 emails incorrectly identified as Spam (they are actually Not Spam).
* **False Negatives (FN)**: 5 emails incorrectly identified as Not Spam (they are actually Spam).

To calculate the accuracy:

1. **Number of Correct Predictions**:
   * TP+TN=40+50=90
2. **Total Number of Predictions**:
   * TP+TN+FP+FN=40+50+5+5=100
3. **Accuracy**:

* Accuracy=100/90​=0.90

So, the accuracy of the spam detection model is 90%. This means that 90% of the time, the model correctly classifies the emails as either Spam or Not Spam.