TEAM: GURUS

REPO LINK: https://github.com/RamcharanSinghRamavath/CS-367-GURUS-Lab3-Q4

LAB 4 (Q3) REPORT:

1. Introduction

The purpose of this project is to develop and evaluate a Bayesian Network classifier using a multi-feature dataset. A Bayesian Network models the probabilistic relationships among the variables in the dataset, allowing us to make predictions while considering dependencies between features. Bayesian Networks are a generalization of the Naive Bayes classifier, where dependencies between variables are explicitly modeled.

2. Methodology

2.1 Bayesian Network Overview

A **Bayesian Network** (also known as a belief network) is a graphical model representing the probabilistic relationships among a set of variables using a directed acyclic graph (DAG). Each node in the graph corresponds to a variable, and the edges represent conditional dependencies between the variables.

Instead of assuming independence between features, as in the Naive Bayes model, the Bayesian Network allows us to model dependencies. This can result in improved predictive performance, especially when there are known dependencies between features.

2.2 Data Preprocessing

The dataset is loaded, and the independent features (X) and the target variable (y) are separated. We split the dataset into training and testing sets using an 80/20 ratio to evaluate the performance of the model on unseen data.

2.3 Bayesian Network Construction

Constructing a Bayesian Network involves the following steps:

1. **Identifying Dependencies**: We determine which features are conditionally dependent on others. This can be done using domain knowledge or by using algorithms such as the K2 algorithm or constraint-based methods to learn the structure from the data.
2. **Parameter Estimation**: Once the structure is defined, we estimate the conditional probability distributions (CPDs) for each node in the network using the training data.
3. **Model Representation**: The Bayesian Network is represented using nodes for each variable and directed edges for conditional dependencies.

2.4 Model Training and Prediction

After building the Bayesian Network, the model is trained using the training set. The learned CPDs are used to make predictions on the test set.

2.5 Evaluation

We evaluate the performance of the Bayesian Network classifier using **accuracy**, which is the percentage of correctly predicted class labels in the test set.

3.RESULT

We will be getting an accuracy of 98%

4.CONCLUSION

The Bayesian Network classifier offers a more flexible approach than the Naive Bayes classifier by explicitly modeling dependencies between features. In this experiment, the classifier achieved an accuracy of 98%, demonstrating its ability to effectively capture relationships between variables.