## **Project Setup Instructions**

### **1. Cloning the Project**

Clone the project from the GitHub repository using the provided link.

### **2. Installation of Required Packages**

Ensure the following packages are installed in your Python environment:

* pgmpy
* Pandas
* Numpy
* sklearn
* Matplotlib
* seaborn

You can install these packages using pip:

!pip install pgmpy pandas numpy sklearn matplotlib seaborn  
  
**3. Uploading Dataset to Google Colab**

* Click on the folder icon in Google Colab.
* Click on the "Upload session" button.
* Select the dataset file downloaded from GitHub.
* After uploading, verify the name of the dataset file.

### **4. Step-by-Step Instructions to Run the Code**

#### **a. Install Required Packages**

First, install the required packages as mentioned in step 2.

#### **b. Load Libraries and Dataset**

* Import necessary libraries like pandas, numpy, pgmpy, sklearn, matplotlib, and seaborn.
* Load your dataset. Replace 'path\_to\_your\_dataset.xlsx' with the actual path to your dataset.

#### **c. Explore the Dataset**

Check the first few rows of your dataset to ensure it's loaded correctly.

#### **d. Define Bayesian Network Structure**

Define the structure of the Bayesian Network by specifying the directed edges between nodes.

#### **e. Fit the Bayesian Network**

Fit the Bayesian Network using Maximum Likelihood Estimator (MLE).

#### **f. Print Conditional Probability Tables (CPTs)**

Print the learned Conditional Probability Tables (CPTs) for each node in the network.

#### **g. Discretize Continuous Variables**

Discretize continuous variables like 'LogP' and 'Molecular Weight' into discrete categories. Adjust discretization strategy according to your dataset.

#### **h. Handle Missing States**

Provide evidence for certain states and perform inference to predict the target variable. Adjust evidence according to your dataset.

#### **i. Cross-Validation**

Split the data into training and testing sets using k-fold cross-validation. Fit the model for each fold and evaluate its performance.

#### **j. Generate ROC Curve**

Plot Receiver Operating Characteristic (ROC) curve to evaluate model performance.

#### **k. Visualize Data Distribution**

Visualize the distribution of 'LogP' and 'Molecular Weight' using histograms.

#### **l. Generate Confusion Matrix**

Generate a confusion matrix to evaluate classification performance.

#### **m. Calculate Metrics**

Calculate metrics like accuracy, precision, recall, and F1-score.

#### **n. Adjust Threshold for F1 Score**

Adjust the decision threshold to optimize F1 score.

#### **o. Final Cross-Validation and Evaluation**

Perform k-fold cross-validation again and evaluate the model's performance.

#### **p. Plot F1 Scores and Accuracies**

Plot F1 scores and accuracies for each fold.