**At first, we try to write python script to implement our idea. So we have three different python scripts:**

1. **Bayesnet.py**

**There are 2 classes defined in this script,** BayesNet and Node. BayesNet : Handles all the function of a Bayesian Network from initialization of Bayesian Network based on a given structure to computation of Conditional Probability Tables (CPTs) of each node. Also, has predict function to make predictions given a data sample.

Node: Contains information pertaining to a single node of a Bayesian Network. CPTs for each node is stored in this class

1. **kFold.py**

Contains function to perform k-fold cross validation

1. **GA.py**

The main file which executes Genetic Algorithm to find the best structure of the Bayesian Network.

As we mentioned before, the dataset that is being used is the Iris dataset. The measure of quality is done by comparing cross validated prediction accuracy. This is also the metric that is used for defining the fitness of Genetic algorithm.

For our case, the edges represented by the various positions in the array are :

Position 1 : 1 ->2

Position 2 : 1 ->3

Position 3 : 1 ->4

Position 4 : 2 ->3

Position 5 : 2 ->4

Position 6 : 3 ->4

The values in these positions have the following meaning:

-1 : Edge starting from second node to first node

0 : No edge

1 : Edge starting from first node to second node

For crossover, halves of two phenotypes are mixed and matched to get two new phenotypes.

Inference is not computationally expensive as it involves looking up values from computed Conditional Probability Tables. The only computationally expensive task is the calculation of the Conditional Probability Tables which is a onetime process.