Universitatea Tehnică ”Gheorghe Asachi” din Iași
Facultatea de Automatică și Calculatoare
Domeniul Calculatoare și Tehnologia Informației
Specializarea Tehnologia Informației

INTELIGENȚĂ ARTIFICIALĂ

Inferențe prin enumerare în rețele bayesiene

Proiect realizat de:
Maieczki Petronela-Sînziana
Butnaru Raimond-Eduard

CUPRINS:

1. Descrierea problemei

Rețele bayesiene sunt structuri probabilistice utilizate pentru a modela relații de dependență între variabilele aleatorii și pentru a calcula probabilități condiționate. Problema abordată constă în calcularea probabilităților condiționate într-o rețea bayesiană folosind algoritmul de inferență prin enumerare.

Obiectivul principal este de a calcula probabilitatea unei variabile de interogare, date fiind variabilele observate. Acest lucru implică marginalizarea variabilelor ascunse prin procesarea tuturor combinațiilor posibile ale acestora.

2. Aspecte teoretice privind algoritmul

2.1. Probabilități condiționate și teorema lui Bayes

Teorema lui Bayes stă la baza raționamentului probabilistic utilizat în rețele bayesiene. Formula generală este:

$P(I|E) = P(E|I) \cdot P(I) / P(E)$,

unde I este ipoteza, E este evidența (provenind din datele observate), P(I) este probabilitatea a-priori a ipotezei, adică gradul inițial de încredere în ipoteză, P(E|I) este verosimilitatea datelor observate, adică măsura în care s-a observat evidența în condițiile îndeplinirii ipotezei, iar P(I|E) este probabilitatea a-posteriori a ipotezei, dată fiind evidența.

2.2. Structura rețelelor bayesiene

Rețele bayesiene sunt reprezentate sub forma unor grafuri orientate aciclice (DAG), unde:

Nodurile reprezintă variabilele aleatorii (ex. Gripa, Abces).

Arcele indică relații de dependență condiționată între variabile.

Fiecare nod este asociat cu un tabel de probabilități condiționate (CPT) care descrie distribuția probabilităților pentru fiecare combinație de valori ale părinților săi.

2.3. Algoritmul de inferență prin enumerare

Inferența prin enumerare calculează probabilitatea totală a unei variabile interogate prin: Fixarea variabilelor observate și marginalizarea variabilelor ascunse prin sumarea tuturor combinațiilor posibile ale valorilor acestora.

3. Modalitatea de rezolvare

3.1. Definirea structurii grafului, incluzând nodurile și relațiile dintre acestea

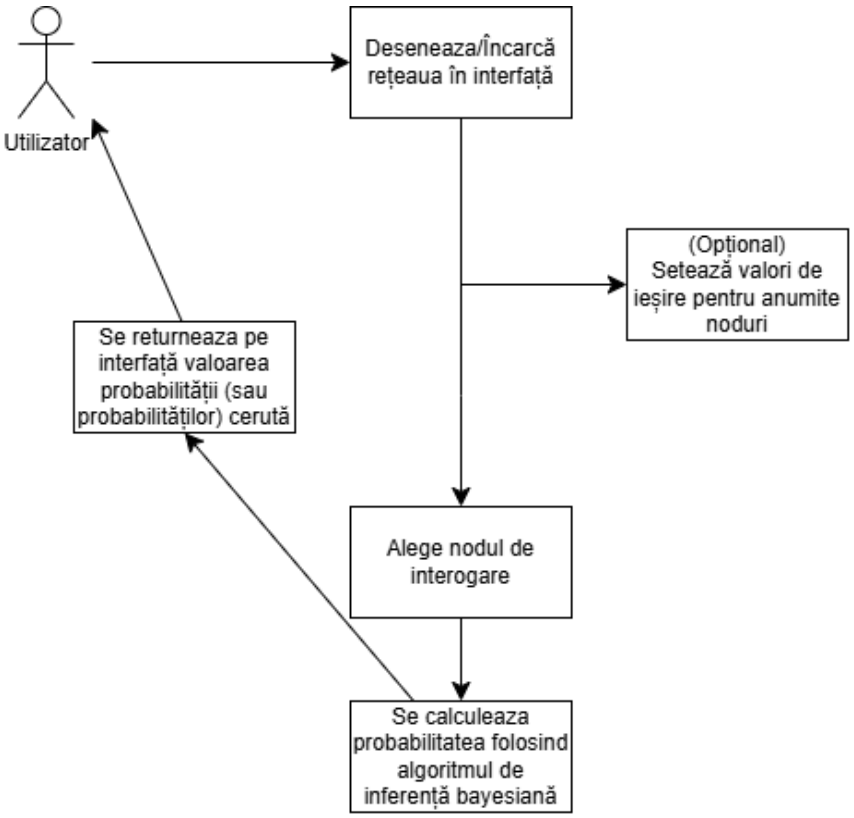
- Specificarea tabelelor de probabilitate condiționată pentru fiecare nod.
- Implementarea algoritmului de inferență prin enumerare:

3.2. Calcularea probabilității unei variabile țintă, având observații pentru unele noduri

- Integrarea observațiilor prin fixarea valorilor nodurilor cunoscute.
- Iterarea peste toate combinațiile posibile pentru variabilele necunoscute.

3.3. Diagrama de interacțiune între utilizator și aplicație

Utilizatorul va deschide aplicația, va desena rețeaua de probabilități (sau poate alege din suita predefinită de rețele), interoghează nodul de interes, după care primește rezultatul calculat folosind algoritmul de inferență prin enumerare în rețele bayesiene.



4. Funcționalitatea componentelor

4.1. Make Observation

Această componentă permite utilizatorului să selecteze un nod de pe interfața grafică și să-i aplice o valoare de ieșire a nodului garantată. Spre exemplu, conform JSON-ului de mai sus, utilizatorul ar putea selecta nodul intitulat „Gripă” care este părinte pentru nodul „Febră” și poate seta valoarea de ieșire fie pe „Da” fie pe „Nu”, oferind astfel calculului probabilităților fie valoarea probabilității pentru evenimentul „Da” fie cea de pe evenimentul „Nu”.

4.2. Query

Această componentă apelează funcția de calcul pentru probabilitatea interogată de pe nodul selectat. Utilizatorul apelează funcția „Query” selectând nodul țintă, în urma selecției acestui nod, se va urma algoritmul de inferență prin enumerare în rețele bayesiene, calculând astfel probabilitatea cerută în condițiile menționate de componenta „Make Observation”.

4.3. Modify Probability Table

Aceasta componentă permite utilizatorului să modifice tabela de probabilități pentru fiecare nod în parte, ba mai mult, permite modificarea fiecărei probabilități în cazul în care un nod are părinți.

4.4. Algoritmul de inferență prin enumerare implementat

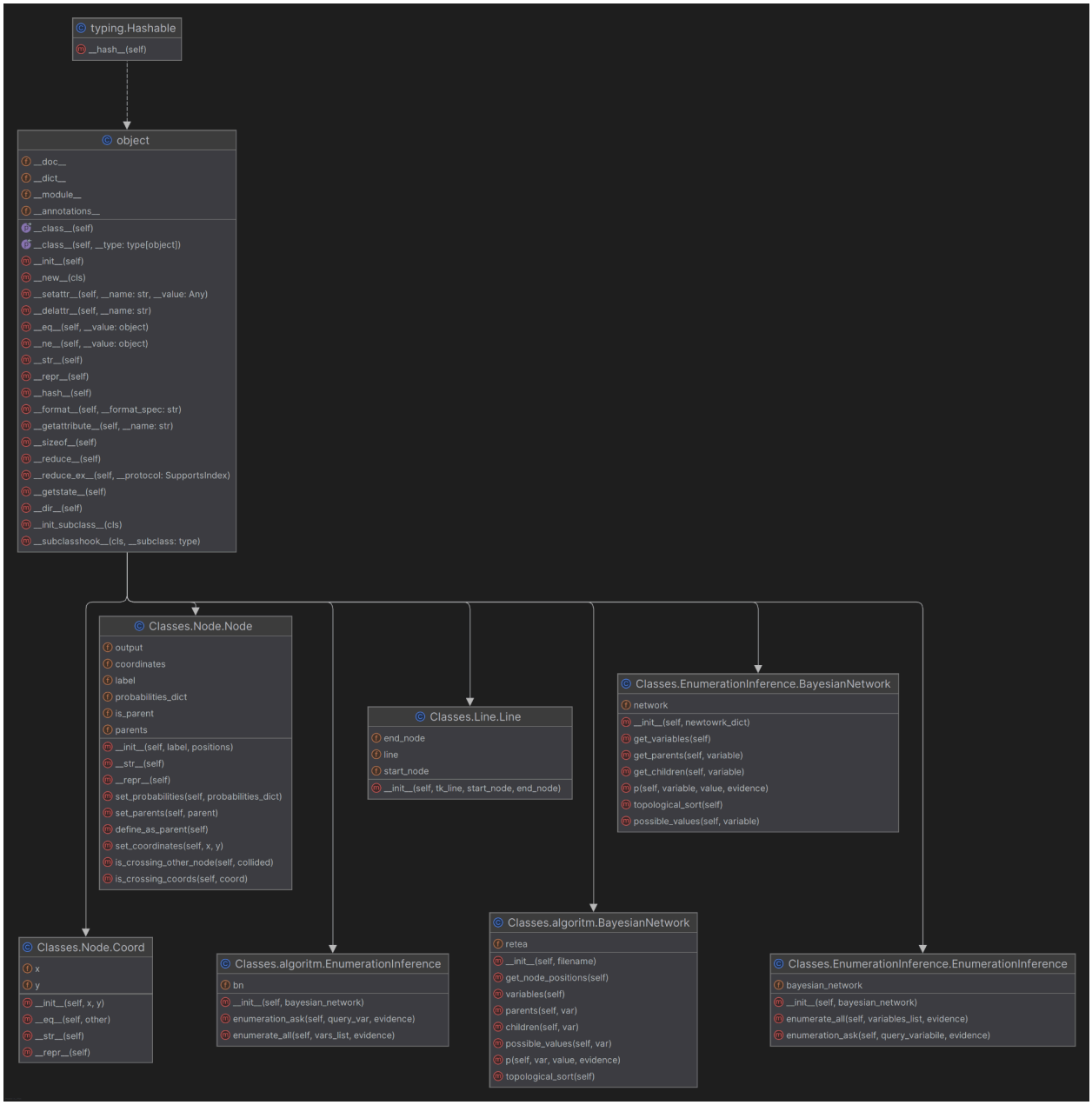
```
  def enumeration_ask(self, query_var, evidence):
      Q = {}
      vars_order = self.bn.topological_sort()
      for val in self.bn.possible_values(query_var):
          extended_evidence = evidence.copy()
          extended_evidence[query_var] = val
          Q[val] = self.enumerate_all(vars_order, extended_evidence)
      # Normalizare
      norm_factor = sum(Q.values())
      for val in Q:
          Q[val] /= norm_factor
      return Q

   def enumerate_all(self, vars_list, evidence):
      if not vars_list:
```

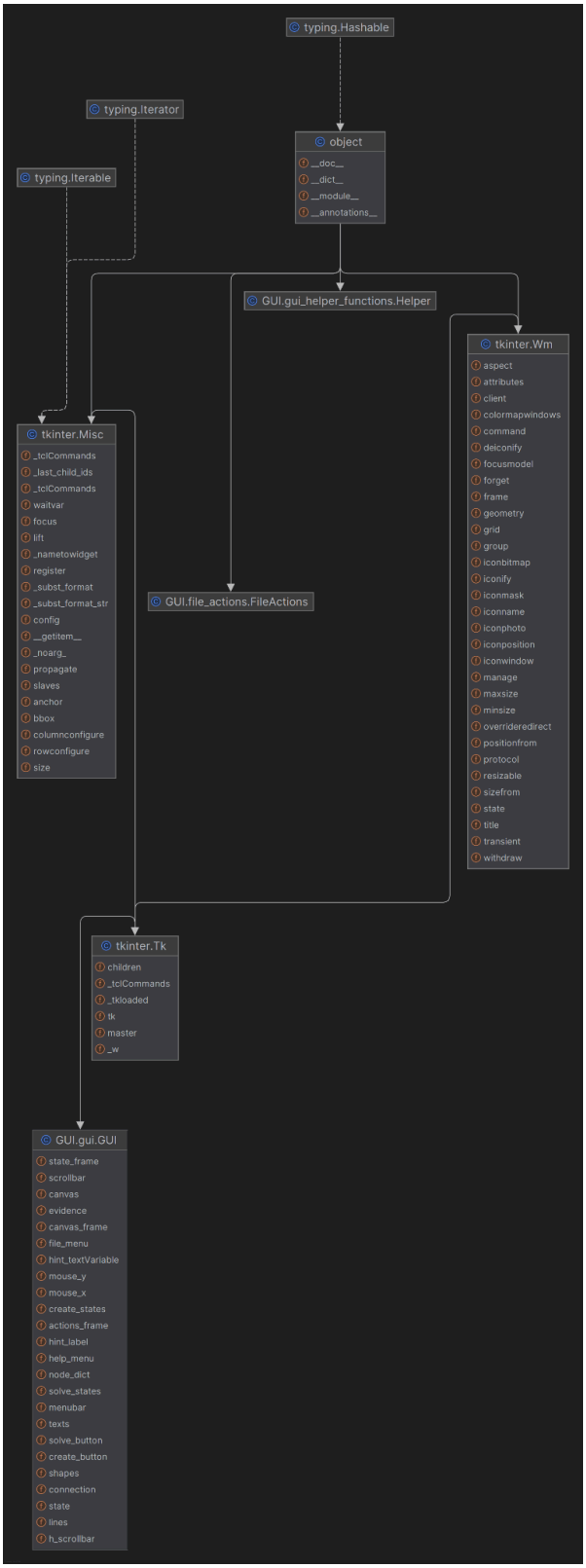
```
    return 1.0
Y = vars_list[0]
rest = vars_list[1:]
if Y in evidence:
    return self.bn.p(Y, evidence[Y], evidence) * self.enumerate_all(rest, evidence)
else:
    total = 0
    for y_val in self.bn.possible_values(Y):
        new_evidence = evidence.copy()
        new_evidence[Y] = y_val
        total += self.bn.p(Y, y_val, new_evidence)*self.enumerate_all(rest, new_evidence)
    return total
```

5. Diagrame UML

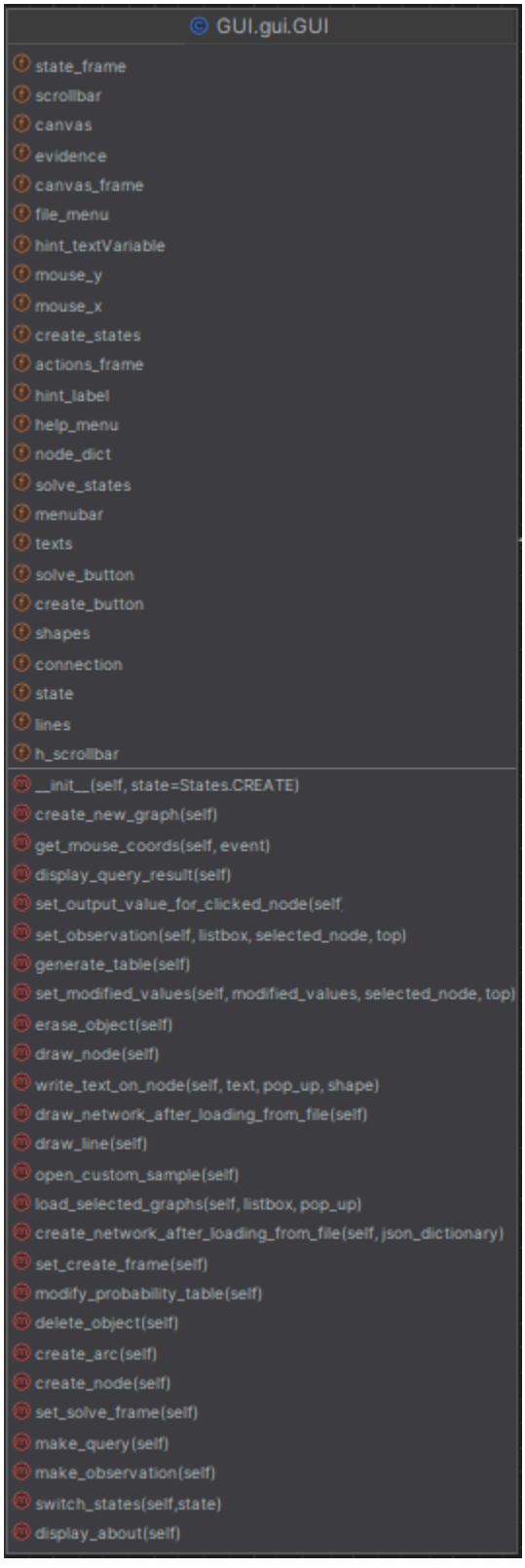
5.1. Diagrama Package Classes



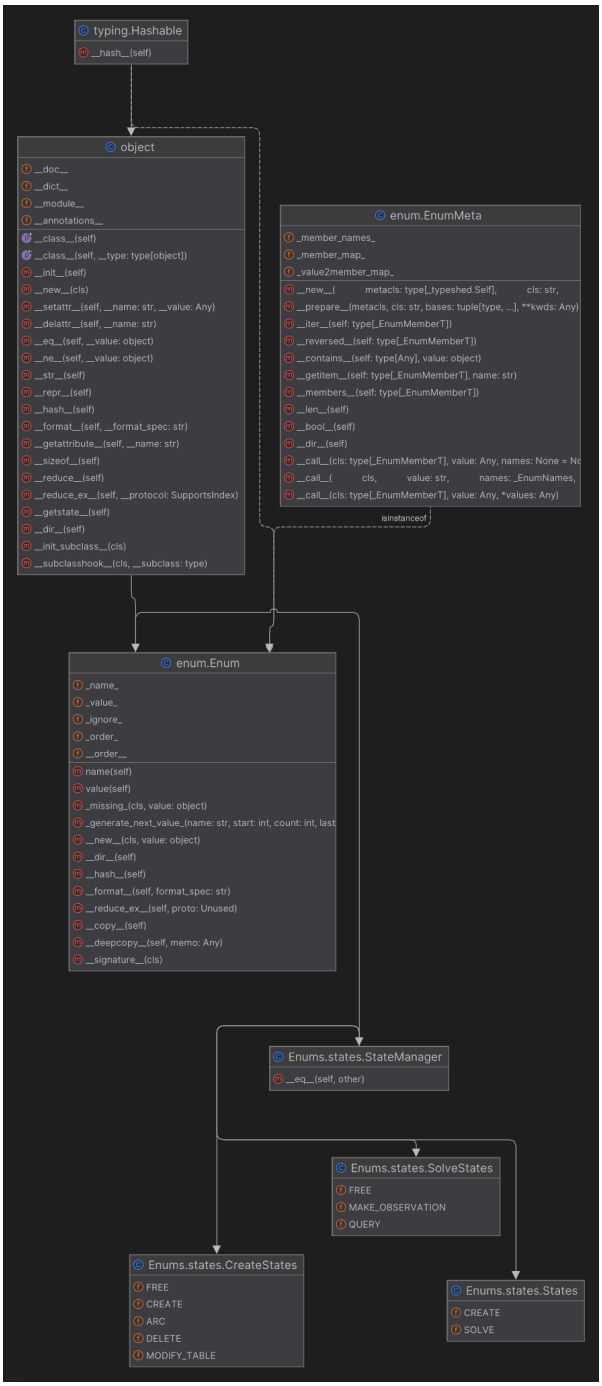
5.2. Diagrama Package GUI



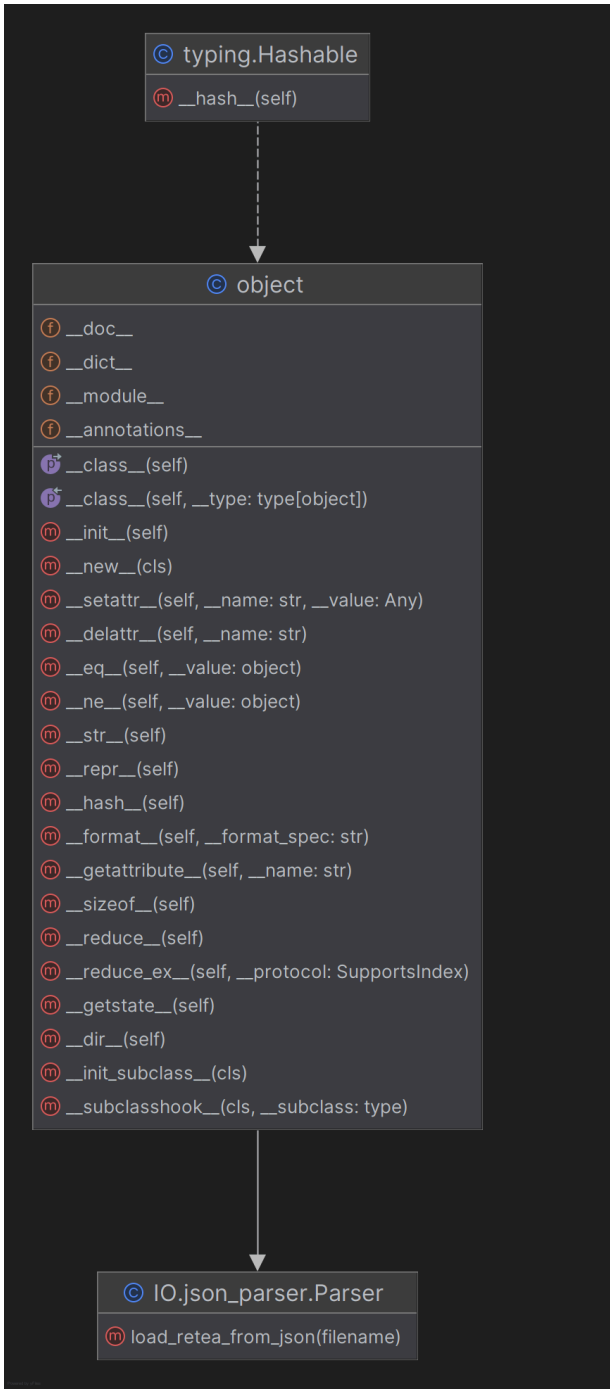
5.3. Diagrama Clasa GUI



5.4. Diagrama Package Enums



5.5 Diagrama Package IO

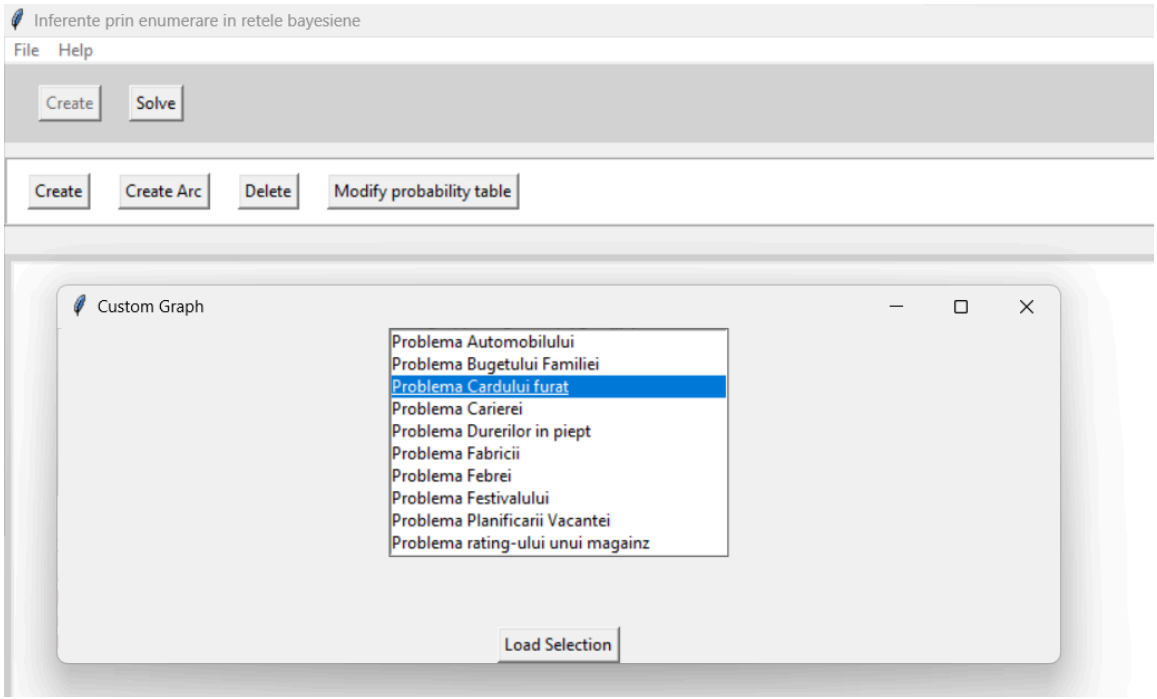


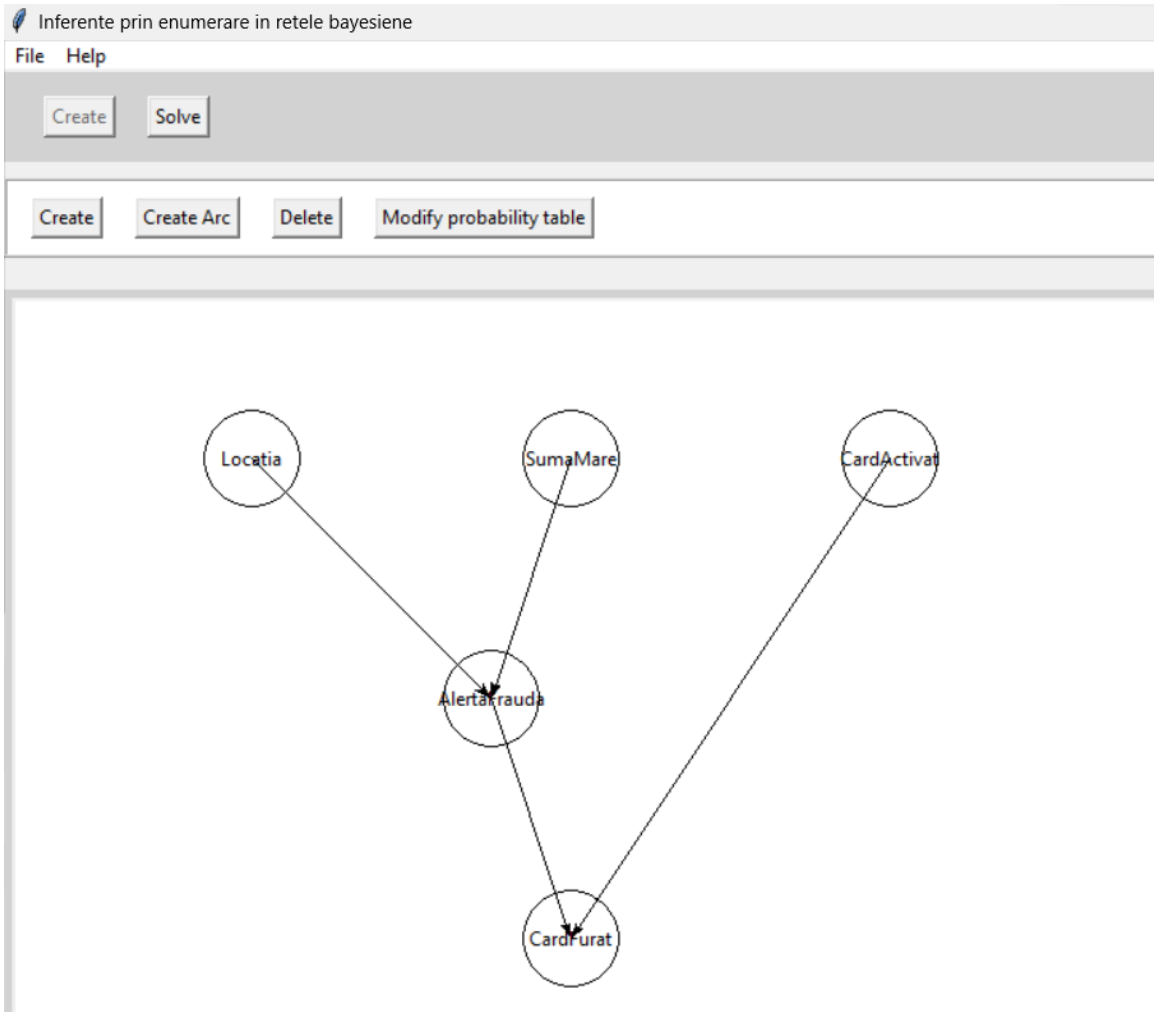
6. Exemplu funcționare aplicație

6.1. Load Selection

Programul citește dintr-un fișier rețeaua pe care se aplica algoritmul și să permită astfel utilizatorului să interogheze anumite noduri care să afișeze rezultatele finale.

File - Load Custom Graph - Select Graph - Load Selection

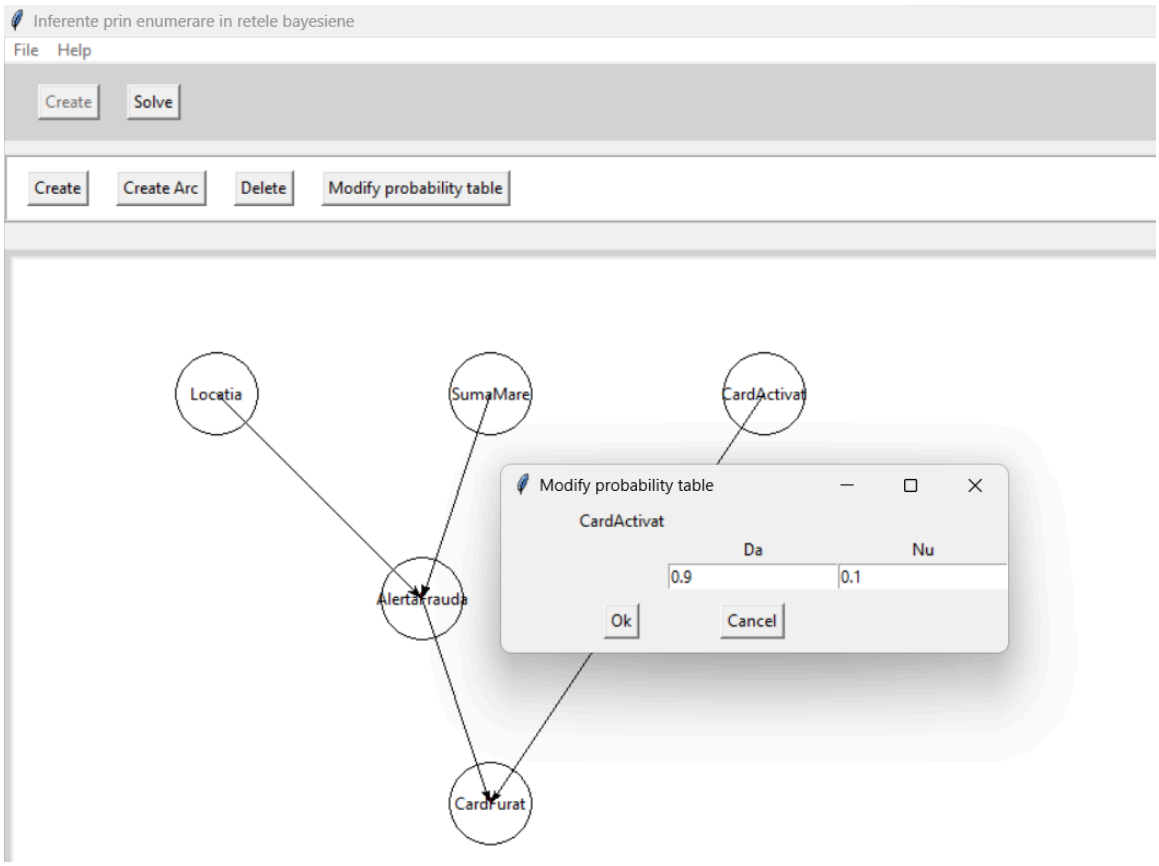



Inferente prin enumerare in retele bayesiene
File
Help
Create
Solve
Create
Create Arc
Delete
Modify probability table
Locatia
SumaMare
CardActivat
AlertaFrauda
CardFurat

6.2. Tabelul de probabilități condiționate

Permite utilizatorului sa modifice probabilitatile unui nod (acestea însumate trebuie să aibă valoare 1).

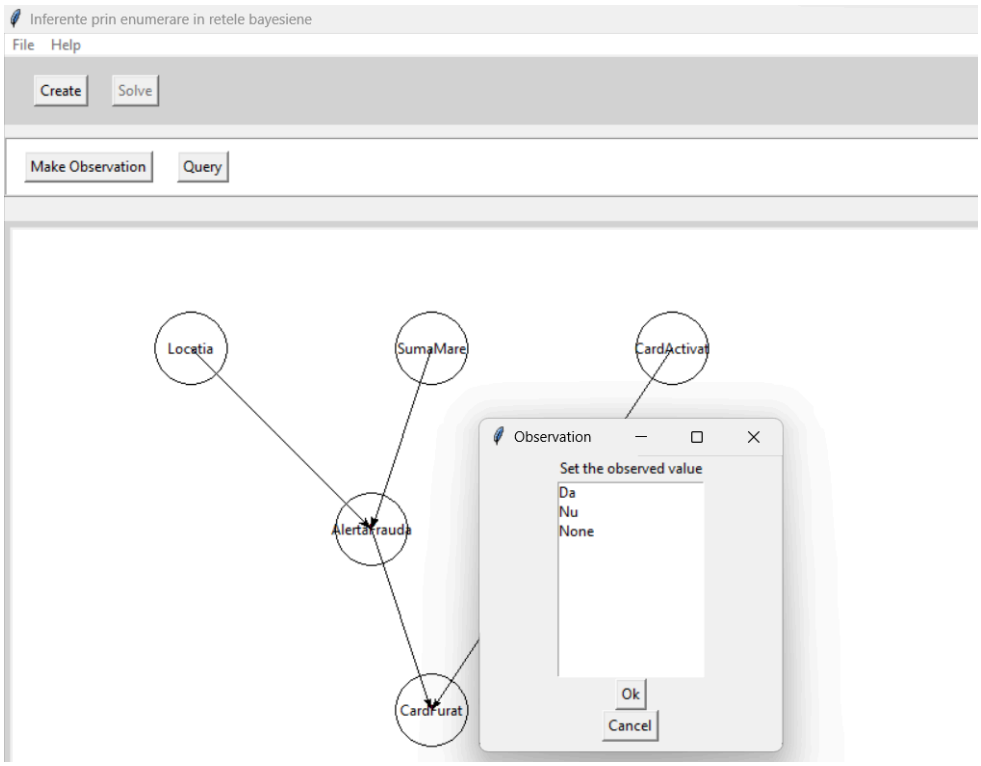
Modify probability table - Select Node



6.3. Make Observation

Permite setarea unei valori de adevăr pentru un nod specific (de exemplu, "Maladie = True"). Actualizează starea rețelei prin fixarea nodurilor observate.

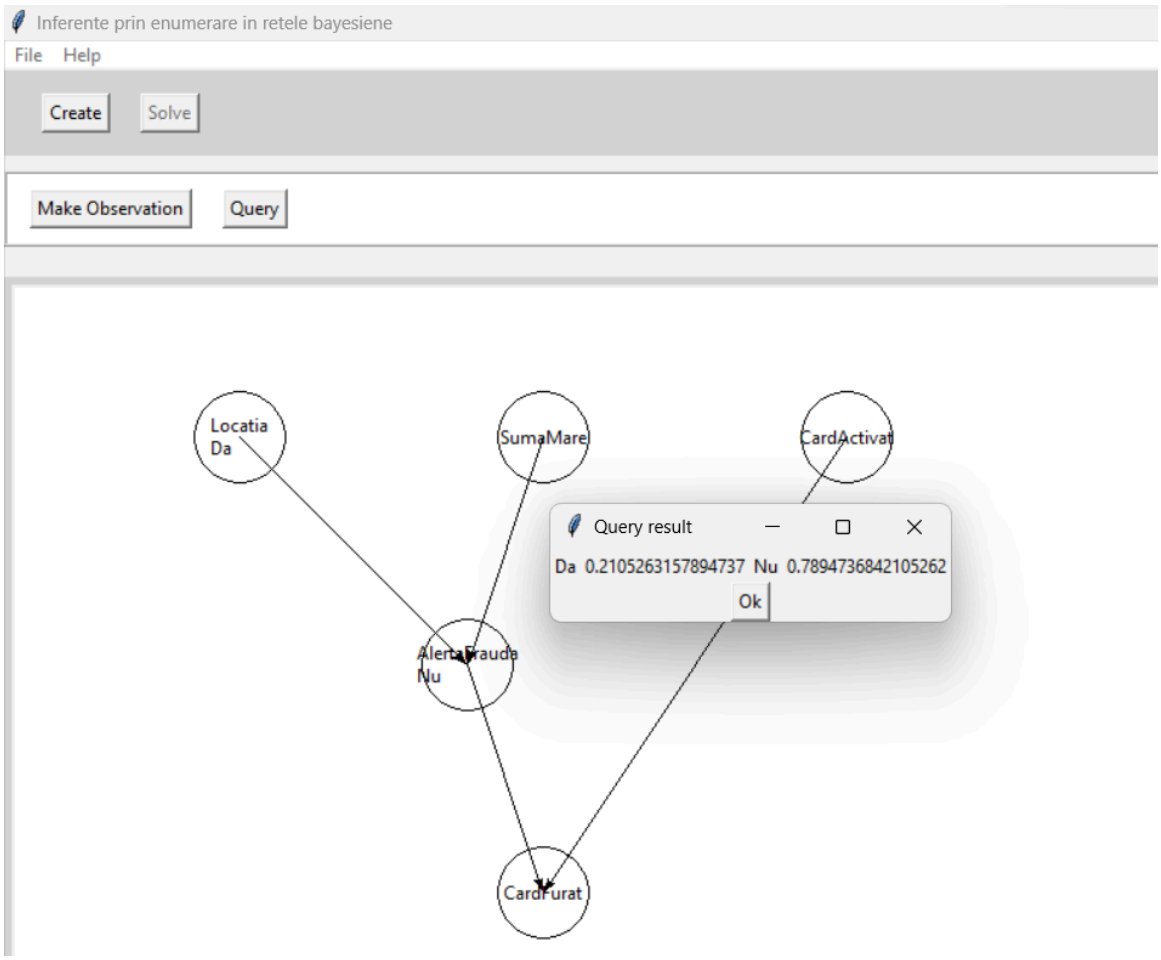
Solve - Make Observation - Select Node



6.4. Query

Permite interogarea unui nod pentru a calcula probabilitatea condiționată, ținând cont de observațiile curente.

Solve - Query



7. Rolul membrilor echipei

Maieczki Petronela-Sînziana:

- Întocmirea documentației
- Generarea de cazuri de testare
- Diagramele
- Proiectarea modelului JSON
- Implementarea componentelor „Query”

Butnaru Raimond-Eduard:

- Dezvoltarea testelor unitare
- Proiectarea arhitecturii aplicației
- Interfața grafică
- Reprezentarea rețelei în interfață
- Implementarea componentelor „Make Observation”

8. Concluzii

Proiectul demonstrează aplicarea algoritmului de inferență prin enumerare în rețele bayesiene, permițând calculul probabilităților condiționate printr-o interfață intuitivă. Utilizatorul poate crea rețele, seta observații, modifica tabele de probabilități și interoga variabile, oferind flexibilitate și eficiență.

9. Biografie

- Laborator 11 Inteligență Artificială - Florin Leon
- Curs 10 Inteligență Artificială - Florin Leon