Risk and Decision-Making for Data Science and AI

Week 6 Lab 5 Answers

Question 1: Absolute Vs Relative Risk

a) Probability of non-bacon eaters who died from Disease A: $\frac{24}{60} = 0.4\ or\ 40\%$

Probability of bacon eaters who died from Disease A: $\frac{10}{40} = 0.25\ or\ 25\%$

Relative risk $\frac{0.4}{0.25} = 1.6$

Relative risk increase = $(1 - 1.6) * 100 = 60\%$

b) Probability of non-bacon eaters who died from Disease A: $\frac{24}{60} = 0.4\ or\ 40\%$

Probability of bacon eaters who died from Disease A: $\frac{10}{40} = 0.25\ or\ 25\%$

Absolute risk increase = $0.4 - 0.25 = 0.15\ or\ 15\%$

Question 2: Utility, Expected Utility and Optimal decision

a) First, we calculate win and lose utilities.

Win utility = $10000 - 100 = 9900$

Lose utility = -100

Then we calculate the expected utility to determine the total expected utility:

Expected utility of win = $9900 * \frac{1}{36} = 275$

Expected utility of lose = $-100 * \frac{35}{36} = -97.22$

Total expected utility of playing = 275 – 97.22 = 177.78

b) Total utility of playing = 177.78

Total utility of not playing = 0

The optimal decision is for Mr. West to play the game since this maximises total expected utility.

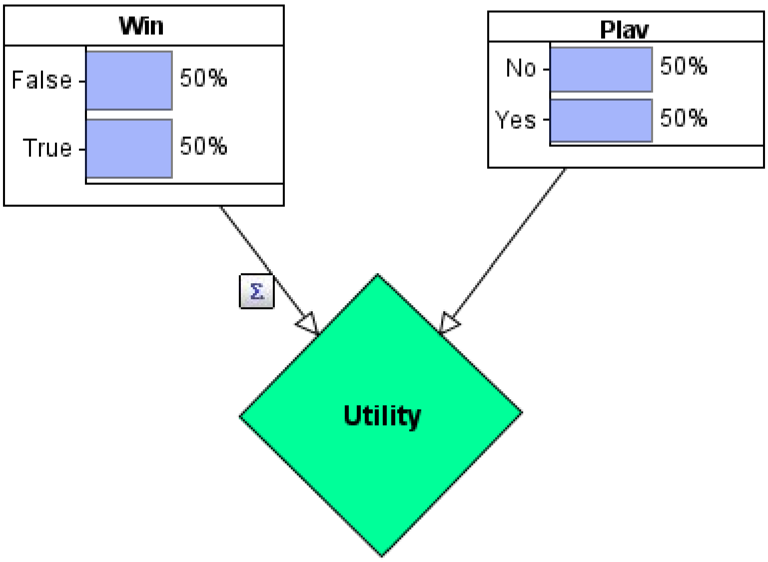
Question 3: Influence Diagram and Decision Tree

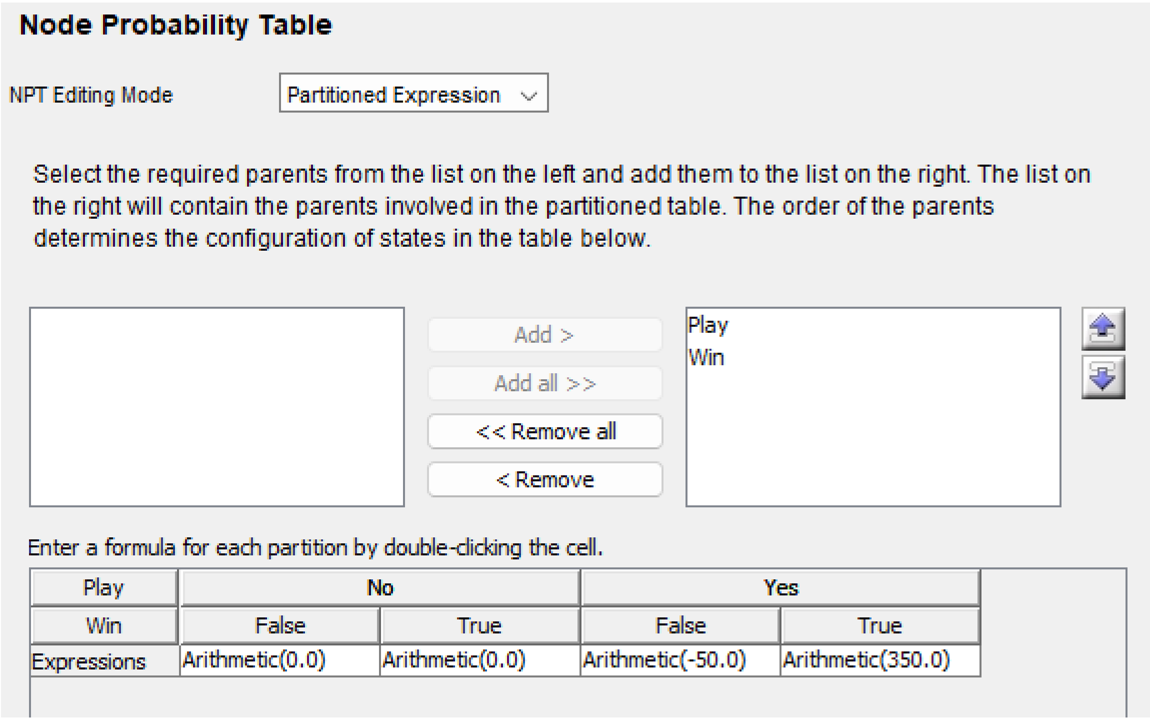
a)

Using Agena

Risk and Decision-Making for Data Science and AI

Week 6 Lab 5 Answers

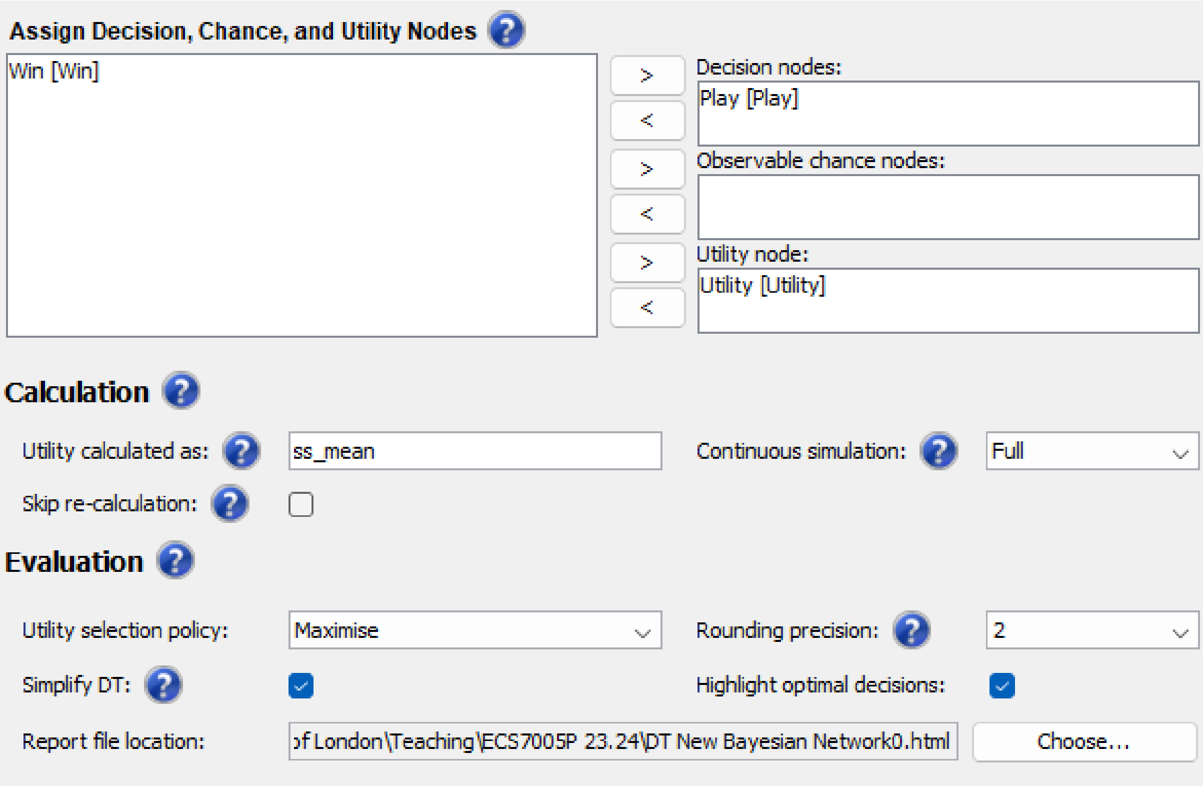


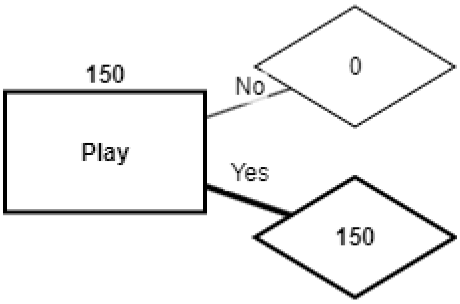


Play	No		Yes	
Win	False	True	False	True
Expressions	Arithmetic(0.0)	Arithmetic(0.0)	Arithmetic(-50.0)	Arithmetic(350.0)

Risk and Decision-Making for Data Science and AI

Week 6 Lab 5 Answers

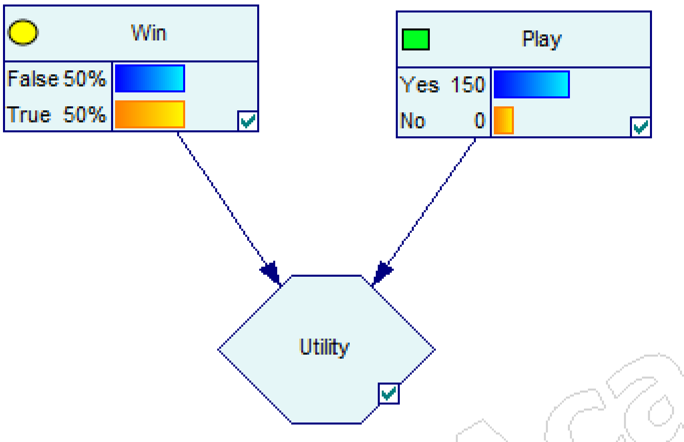




Using Genie:

Risk and Decision-Making for Data Science and AI

Week 6 Lab 5 Answers



Using Python:

```
import os

%matplotlib inline
from pylab import *
import matplotlib.pyplot as plt
from IPython.display import display,HTML

import math

import pyAgrum as gum
import pyAgrum.lib.notebook as gnb
```

[1] ✓ 0.8s

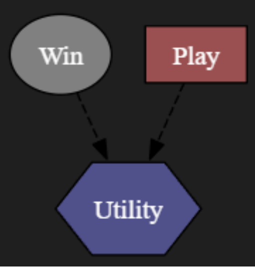
```
infdiag = gum.InfluenceDiagram()

win = infdiag.addChanceNode(gum.LabelizedVariable("Win","Win",["False","True"]))
play = infdiag.addDecisionNode(gum.LabelizedVariable("Play","Play",["Yes","No"]))
utility = infdiag.addUtilityNode(gum.LabelizedVariable("Utility","Utility",1))

infdiag.addArc("Win","Utility")
infdiag.addArc("Play","Utility")

infdiag
```

[2] ✓ 0.0s



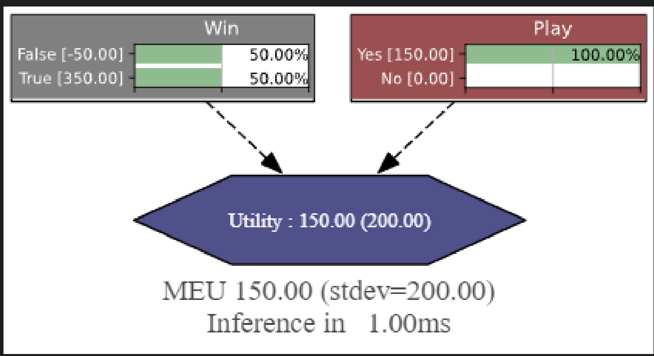
Risk and Decision-Making for Data Science and AI

Week 6 Lab 5 Answers

```
infdiag.cpt("Win").fillWith([0.5,0.5])

infdiag.utility("Utility")[{'Play':'No','Win':'False'}] = 0
infdiag.utility("Utility")[{'Play':'No','Win':'True'}] = 0
infdiag.utility("Utility")[{'Play':'Yes','Win':'False'}] = -50
infdiag.utility("Utility")[{'Play':'Yes','Win':'True'}] = 350
```
✓ 0.0s

```
gnb.showInference(infdiag)
```
✓ 0.1s



So the optimal decision is to play