Reasoning under Uncertainty: Sustainable Investment Analysis

Development and Implementation of a Bayesian Network

Johannes Van Cauwenberghe

Task 1. Conceptual Design of a DSS

This project focuses on designing a Decision Support System (DSS) to evaluate the sustainability and financial viability of investment projects. By using a Bayesian Network (BN), the system models key factors, such as market demand and government subsidies, to predict two critical outputs: environmental impact and return on investment (ROI). This approach allows investors to make informed decisions under uncertainty, a frequent challenge when evaluating green projects.

The purpose of the DSS is twofold: first, to predict the likelihood of achieving high ROI for green investment projects, and second, to estimate the probability of these projects having a positive environmental impact. This dual-output approach ensures that stakeholders can evaluate the financial and environmental trade-offs simultaneously. The DSS is applicable in areas like sustainable finance and ESG (Environmental, Social, Governance) investing. Stakeholders include the investment fund's partners, project managers, as well as society at large who may benefit from the positive environmental impacts.

- **Task**: Predict the probability of achieving both high ROI and positive environmental impacts, identify critical influencing factors, and provide a support system for evaluating investment strategies.
- **Objectives**: (1) Develop a conceptual design of a Bayesian Network to represent the dependencies between input factors and outputs; (2) implement and test the Bayesian Network in Python; (3) ensure that the DSS can accommodate real-world constraints and provide actionable insights.
- **Constraints**: Limit the Bayesian Network to 4-6 nodes, relying on estimates and informed assumptions when real-world data is unavailable.

The probabilistic relationships are informed estimates. They can be refined and updated dynamically. We can use Bayesian estimation or maximum likelihood estimation (MLE) to recompute the probabilities as more data becomes available.

System Architecture and Design

The DSS comprises three main components: data input, a processing engine, and outputs. The data input relies on a corporate investment fund database. Given the proprietary nature of such data, we will use estimates informed by publicly available research. The processing engine is a Bayesian Network that evaluates probabilistic dependencies among the input factors. The outputs are twofold: probabilities of high ROI and probabilities of positive environmental impact.

The Bayesian Network is designed to be standalone but can integrate with other systems, such as financial reporting tools or green investment platforms. This modularity ensures adaptability to different user needs while maintaining a clear focus on dual-output predictions.

Input Data and Knowledge

The system assumes the following inputs:

1. **Government Subsidies**: Subsidies are assumed to reduce costs and incentivise environmentally friendly practices. The availability of subsidies significantly impacts project success rates. For instance, it is assumed that projects that receive subsidies succeed 70% of the time.
2. **Market Demand**: High demand for green products, such as renewable energy and electric bicycles, improves project ROI. It is estimated that high demand correlates with high ROI in 65% of cases.
3. **ROI Benchmarks**: The system assumes a target ROI of 7%, reflecting typical returns for 'dark green' investments in this category. ROI depends on factors like demand and subsidies.
4. **Environmental Impact**: The investment strategy prioritises projects with positive environmental impacts. A project is considered to have a positive environmental impact if it either employs sustainable solutions to avoid emissions, generates a substantial portion of its energy needs from renewable sources, or purchases offsets to compensate for its negative impacts.

The primary outputs of the system are predictions of high/low ROI and predictions of positive/negative environmental impacts. These outputs are intended to help investors allocate green funds more effectively and achieve better environmental outcomes.

Development Risks

Several risks are associated with developing this DSS. Data quality is a primary concern, as the system depends on accurate and unbiased data inputs. Furthermore, the model's simplicity, constrained by the 4-6 node limit, may lead to oversimplified dependencies, reducing predictive power. These risks necessitate iterative refinement and validation.

Task 2. Development of a Simple Bayesian Network

To construct the Bayesian Network, we conceptualise a problem where an investor evaluates green projects. The investor assesses whether government subsidies are available, and whether market demand is high. These factors determine the ROI and the environmental impact. The goal is to evaluate these outputs based on the inputs: government subsidies and market demand.

Nodes in the Network

All nodes remain binary, as the network is built using discrete variables.

1. Government Subsidies (Yes/No)
2. Market Demand (High/Low)
3. Environmental Impact (Positive/Negative)
4. Regulatory Risks (High/Low)
5. Project Viability (Viable/Not Viable)
6. ROI (High/Low)

Dependencies and Links

- **Subsidies → Environmental Impact**:
  Subsidies increase the likelihood of positive environmental outcomes by incentivising sustainable practices and reducing project costs.
- **Subsidies → ROI**:
  Subsidies improve financial performance by lowering costs, directly contributing to higher returns on investment.

- **Demand $\rightarrow$ Environmental Impact**:
  High demand indicates scalability but can also put stress on resources, affecting environmental outcomes.
- **Demand $\rightarrow$ ROI**:
  High demand improves ROI by increasing revenue and scaling profits.
- **Regulatory Risks $\rightarrow$ Project Viability**:
  High regulatory risks decrease the likelihood of project viability due to stricter requirements, higher compliance costs, or legal barriers.
- **Project Viability $\rightarrow$ ROI**:
  Project viability determines whether the project can generate returns. Non-viable projects are less likely to achieve high ROI.

Conditional Probabilities

Subsidies

Subsidies are assumed to be available in 30% of cases:

- $P(\text{Subsidies} = 1) = 0.3$, $P(\text{Subsidies} = 0) = 0.7$

Market Demand

High market demand is still assumed in 65% of cases:

- $P(\text{Demand} = 1) = 0.65$, $P(\text{Demand} = 0) = 0.35$

Regulatory Risks

The likelihood of high regulatory risks:

- $P(\text{Regulatory Risks} = 1) = 0.4$, $P(\text{Regulatory Risks} = 0) = 0.6$

Project Viability

The probability of project viability depends on regulatory risks:

- $P(\text{Viability} = 1|\text{Regulatory Risks} = 0) = 0.8$
- $P(\text{Viability} = 1|\text{Regulatory Risks} = 1) = 0.5$

Environmental Impact

The likelihood of positive environmental impact is influenced by subsidies and demand:

Subsidies	Demand	Environmental Impact=0	Environmental Impact=1
1 (Yes)	1 (High)	0.3	0.7
1 (Yes)	0 (Low)	0.5	0.5
0 (No)	1 (High)	0.8	0.2
0 (No)	0 (Low)	0.9	0.1

ROI

The probability of achieving high ROI now depends on subsidies, demand, and project viability:

Subsidies	Demand	Project Viability	ROI=0	ROI=1
1 (Yes)	1 (High)	1 (Viable)	0.5	0.5
1 (Yes)	1 (High)	0 (Not Viable)	0.55	0.45
1 (Yes)	0 (Low)	1 (Viable)	0.6	0.4
1 (Yes)	0 (Low)	0 (Not Viable)	0.65	0.35
0 (No)	1 (High)	1 (Viable)	0.7	0.3
0 (No)	1 (High)	0 (Not Viable)	0.75	0.25
0 (No)	0 (Low)	1 (Viable)	0.8	0.2
0 (No)	0 (Low)	0 (Not Viable)	0.9	0.1

Task 3. Implementation of the Bayesian Network in Python

The Bayesian Network is implemented using the `pgmpy` library. The nodes, dependencies, and conditional probability distributions (CPDs) are encoded as follows:

```python
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from IPython.display import Image
import os; os.environ['NUMEXPR_MAX_THREADS'] = '28'

# Define the structure
model = BayesianNetwork([
    ('Subsidies', 'ROI'),
```

```
    ('Subsidies', 'Environmental Impact'),
    ('Demand', 'ROI'),
    ('Demand', 'Environmental Impact'),
    ('Regulatory Risks', 'Project Viability'),
    ('Project Viability', 'ROI')
])

# Plot
viz = model.to_graphviz()
viz.draw('ex.png', prog='dot')
Image('ex.png')
```

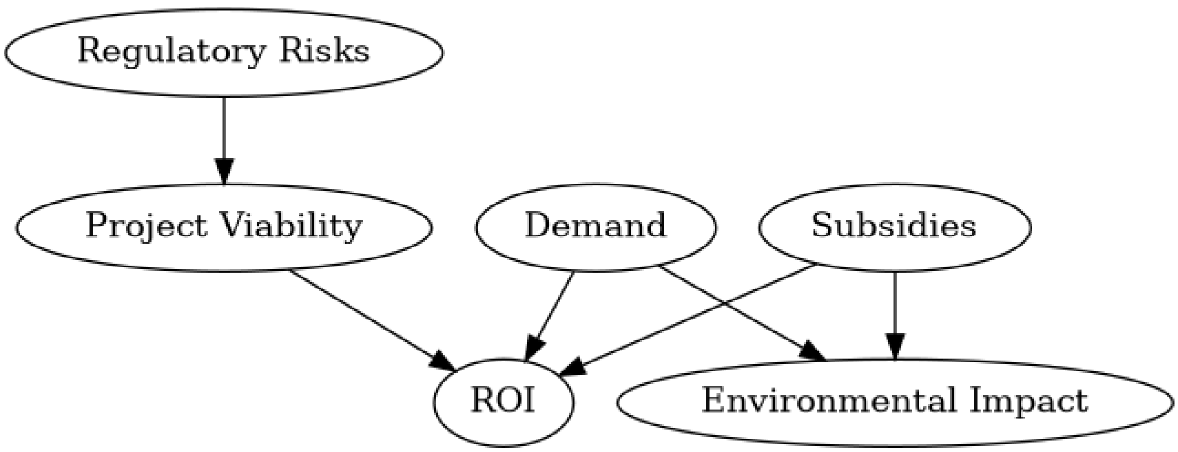


Figure 1: A visualisation of the Bayesian Network.

```
# Define CPDs
cpd_subsidies = TabularCPD(
    variable='Subsidies', variable_card=2,
    values=[
        [0.7],  # P(Subsidie=1)
        [0.3]   # P(Subsidie=0)
        ])
cpd_demand = TabularCPD(
    variable='Demand', variable_card=2,
    values=[
        [0.65],     # P(Demand=1)
        [0.35]      # P(Demand=0)
        ])
cpd_environmental_impact = TabularCPD(
```

```
    variable='Environmental Impact', variable_card=2,
    values=[
        [0.9, 0.8, 0.5, 0.3],  # P(Impact=0)
        [0.1, 0.2, 0.5, 0.7]   # P(Impact=1)
        ],
    evidence=['Subsidies', 'Demand'], evidence_card=[2, 2]
)
cpd_regulatory_risks = TabularCPD(
    variable='Regulatory Risks', variable_card=2,
    values=[[0.6], [0.4]]
    )
cpd_project_viability = TabularCPD(
    variable='Project Viability', variable_card=2,
    values=[[0.8, 0.5], [0.2, 0.5]],
    evidence=['Regulatory Risks'], evidence_card=[2]
)
cpd_roi = TabularCPD(
    variable='ROI', variable_card=2,
    values=[
        [0.9, 0.8, 0.75, 0.7, 0.65, 0.6, 0.55, 0.5],
        [0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]
    ],
    evidence=['Subsidies', 'Demand', 'Project Viability'],
    evidence_card=[2, 2, 2]
)

# Add them to the model
model.add_cpds(cpd_subsidies, cpd_demand,
    cpd_environmental_impact, cpd_regulatory_risks,
    cpd_project_viability, cpd_roi)

assert model.check_model()
```

Querying the Bayesian Network:

Queries can be performed to evaluate the probabilities of ROI and environmental impact given various scenarios. We start by setting up the model for inference. We do this by initialising it with `VariableElimination`.

```
from pgmpy.inference import VariableElimination

inference = VariableElimination(model)
```

The objective is now to ask three questions and use the implemented Bayesian Network to answer them. (1) Calculate the probability of a positive `Environmental Impact`, in the absence of evidence; (2) Introduce evidence, in the form of `Subsidies` and `High Demand` and ask the model to return its prediction for `ROI`; (3) Test a worst-case scenario in which the posterior is conditioned on an absence of subsidies, low viability and high risks.

Query 1: Likelihood of Positive Environmental Impact

Start by calculating the probability of Positive Environmental Impact:

```
# Query the inference object
query_result = inference.query(
    variables=['Environmental Impact'])
print(query_result)
```

```
+--------------------------+------------------------------+
| Environmental Impact     |   phi(Environmental Impact) |
+==========================+==============================+
| Environmental Impact(0) |                       0.7345 |
+--------------------------+------------------------------+
| Environmental Impact(1) |                       0.2655 |
+--------------------------+------------------------------+
```

We can plot this again, for a visual aid:

```
# Plot the `ancestor graph`
model.get_ancestral_graph('Environmental Impact')\
    .to_graphviz().draw('ex.png', prog='dot')
Image('ex.png')
```

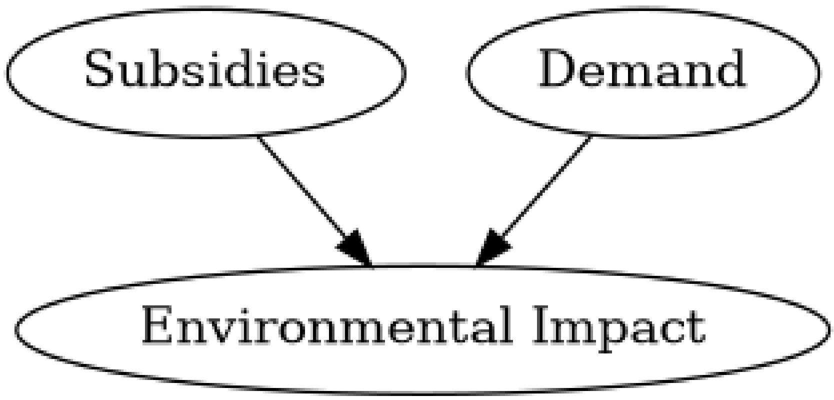


Figure 2: Visualisating the direct influence of `Subsidies` and `Demand` on `Environmental Impact`.

Query 2: Probability of High ROI Given Subsidies and High Demand

Query the probability of high ROI given `Subsidies` and high `Demand`:

```
query_result = inference.query(
    variables=['ROI'], evidence={
        'Subsidies': 1, 'Demand': 1
        })
print(query_result)
```

```
+--------+------------+
| ROI    |   phi(ROI) |
+========+============+
| ROI(0) |     0.5340 |
+--------+------------+
| ROI(1) |     0.4660 |
+--------+------------+
```

Query 3: Impact of Absence of Subsidies, Low Viability and High Risks on ROI

Lastly, query the probability of `ROI` given absence of `Subsidies`, low `Viability` and high `Risks`:

```
query_result = inference.query(
    variables=['ROI'], evidence={
        'Subsidies': 0,
```

```
        'Project Viability': 0,
        'Regulatory Risks': 1
        })

print(query_result)
```

```
+--------+------------+
| ROI    |   phi(ROI) |
+========+============+
| ROI(0) |     0.8475 |
+--------+------------+
| ROI(1) |     0.1525 |
+--------+------------+
```

Bonus Task: Causal Inference with Bayesian Networks

1. Extending the network with Public Perception

The model can be improved by including a node for Public Perception. A positive public perception depends on positive environmental impact and low regulatory risks. We can formalise the node and links as follows:

Additional Node in the Network:

7. Public Perception (Positive/Negative)

Added Dependencies and Links

- **Environmental Impact → Public Perception**: Positive environmental outcomes improve public perception.
- **Regulatory Risks → Public Perception:** Higher risks might lead to negative public sentiment due to perceived instability or concerns.

For each combination, we assume reasonable numerical probabilities for $P(\text{Public Perception} = 1)$ and $P(\text{Public Perception} = 0)$.

Environmental Impact	Regulatory Risks	Public Perception=0	Public Perception=1
0 (Negative)	0 (Low)	0.6	0.4
0 (Negative)	1 (High)	0.8	0.2

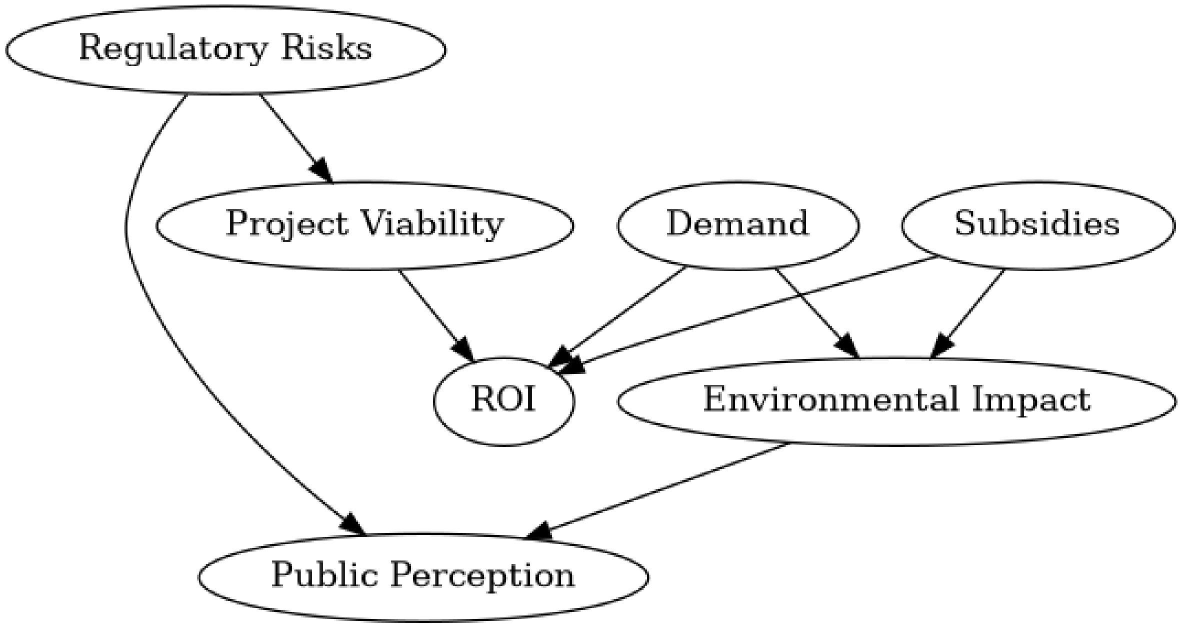
Environmental Impact	Regulatory Risks	Public Perception=0	Public Perception=1
1 (Positive)	0 (Low)	0.3	0.7
1 (Positive)	1 (High)	0.5	0.5

The additional dependencies are added to our Bayesian Network in `pgmpy` as nodes and edges, and the conditional probability table is added as a `TabularCPD` instance.

```
model.add_nodes_from(['Public Perception'])
model.add_edges_from([
    ('Environmental Impact', 'Public Perception'),
    ('Regulatory Risks', 'Public Perception')
])

# Add a CPD for Public Perception
cpd_public_perception = TabularCPD(
    variable='Public Perception',  # The node name
    variable_card=2,
    values=[
        [0.6, 0.8, 0.3, 0.5],  # P(Public Perception = 0 | Parents)
        [0.4, 0.2, 0.7, 0.5]   # P(Public Perception = 1 | Parents)
    ],
    evidence=['Environmental Impact', 'Regulatory Risks'],  # Parent nodes
    evidence_card=[2, 2]  # States for each parent
)

model.add_cpds(cpd_public_perception)
assert model.check_model()
viz = model.to_graphviz()
viz.draw('ex.png', prog='dot')
Image('ex.png')
```



2. Causal Effects

Best-case scenario in terms of Public Perception:

We can now calculate the best-case scenario in terms of Public Perception, assuming we cannot know the environmental impact in advance. In other words, we model the causal effect of Low Regulatory Risks, High Subsidies and High Demand on Public Perception:

```
query_result = inference.query(
    variables=['Public Perception'], evidence={
        'Regulatory Risks': 0,
        'Subsidies': 1,
        'Demand': 1
        })
print(query_result)
```

```
+----------------------+-------------------------+
| Public Perception    |   phi(Public Perception) |
+======================+=========================+
| Public Perception(0) |                  0.3900 |
+----------------------+-------------------------+
| Public Perception(1) |                  0.6100 |
+----------------------+-------------------------+
```

Worst-case scenario:

This is the causal effect of High Regulatory Risks, Low Subsidies and Low Demand on Public Perception:

```
query_result = inference.query(
    variables=['Public Perception'], evidence={
        'Regulatory Risks': 1,
        'Subsidies': 0,
        'Demand': 0
        })
print(query_result)
```

```
+----------------------+--------------------------+
| Public Perception    |   phi(Public Perception) |
+======================+==========================+
| Public Perception(0) |                   0.7700 |
+----------------------+--------------------------+
| Public Perception(1) |                   0.2300 |
+----------------------+--------------------------+
```

Note that the worst-case scenario is still achieving a public perception of 20%!

Worst-case alternative:

And lastly, we model the alternate case where `Environmental Impact` is known to be high, while the all other variables satisfy worst-case conditions. This is the effect of High Regulatory Risks, Low Subsidies and Low Demand on Public Perception:

```
query_result = inference.query(
    variables=['Public Perception'], evidence={
        'Regulatory Risks': 1,
        'Subsidies': 0,
        'Demand': 0,
        'Environmental Impact': 1
        })
print(query_result)
```

```
+----------------------+--------------------------+
| Public Perception    |   phi(Public Perception) |
+======================+==========================+
| Public Perception(0) |                   0.5000 |
```

```
+----------------------+-------------------------+
| Public Perception(1) |                  0.5000 |
+----------------------+-------------------------+
```

3. Interventions

Performing an intervention shows how setting certain variable to zero affects the output. Here we assess the impact of removing subsidies to ROI:

```
query_result_intervention = inference.query(
    variables=['ROI'], evidence={
        'Subsidies': 0
        })
print(query_result_intervention)
```

```
+--------+------------+
| ROI    |   phi(ROI) |
+========+============+
| ROI(0) |     0.8211 |
+--------+------------+
| ROI(1) |     0.1789 |
+--------+------------+
```

Conclusion

This project demonstrates the practical application of a Bayesian Network to evaluate sustainability and financial viability in investment projects. By focusing on two key outputs–environmental impact and return on investment–the DSS provides a dual-faceted approach to decision-making. The structured use of probabilistic reasoning ensures that stakeholders can account for uncertainties inherent in green investments while maintaining simplicity in design with a limited number of nodes.

For further development, the system could be enhanced by introducing feedback loops, as a high ROI may be due to enhanced trust and funding opportunities, in turn due to positive environmental impacts. However, the current capabilities of the pgmpy library do not support loops.

Future iterations of the DSS could refine and update conditional dependencies in the Bayesian Network dynamically. Techniques such as Bayesian estimation or maximum likelihood estimation (MLE) can be employed to recompute the probabilities, improving the accuracy of the system as more data becomes available.

Word Count

The word count is 1985.

References

Nielsen, A. (2016). *Pygotham talk*. New York. Retrieved December 18, 2024, from https://www.youtube.com/watch?v=DEHqIxX1Kq4

Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.