Credit Risk Prediction with a Bayesian Network

Use Case Description

Credit risk prediction refers to estimating the likelihood that a borrower will default on a loan (fail to repay). I will build a simplified Bayesian network to model credit risk for loan applicants. The network includes variables that are commonly considered by lenders when evaluating loans. In my example, I have used the following variables:

- **IncomeLevel** – The borrower's income category (High or Low). A higher income may indicate more capacity to repay.

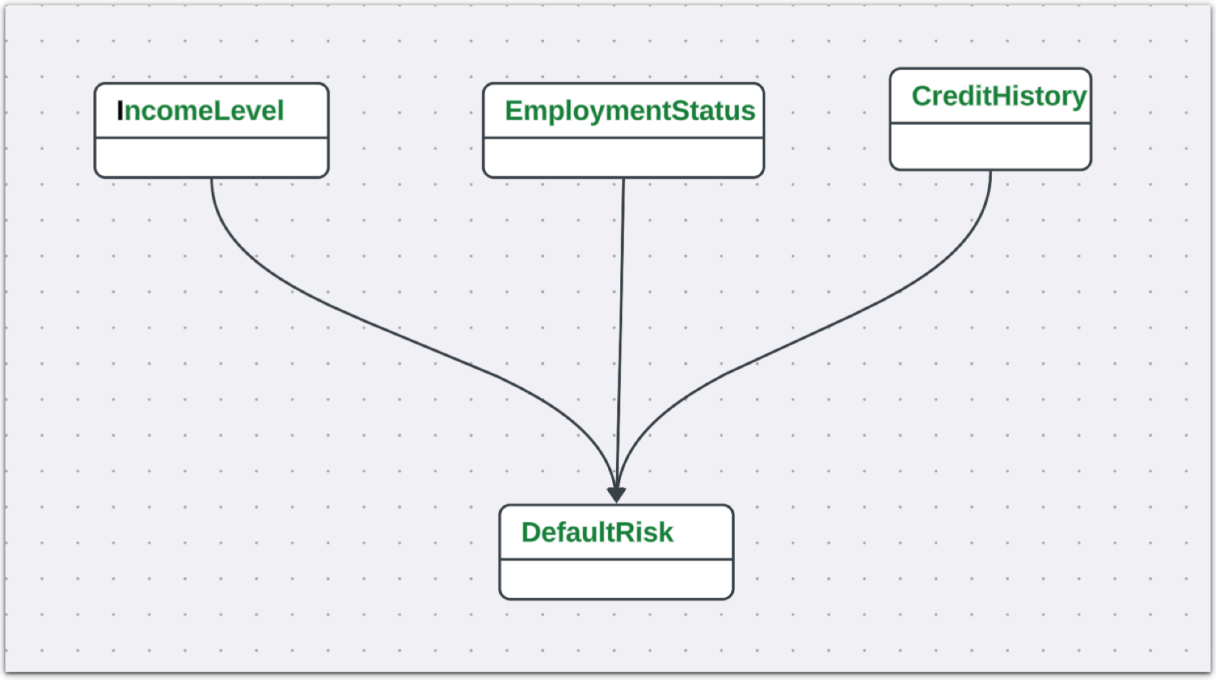
- **EmploymentStatus** – Whether the borrower is Employed or Unemployed. Stable employment usually reduces credit risk.

- **CreditHistory** – The quality of the borrower's past credit history (Good or Poor). A good credit history means they have a record of timely repayments, whereas a poor history indicates previous issues (late payments, defaults, etc.).

- **DefaultRisk** – This node represents the overall credit risk of the borrower defaulting on the loan. We can think of it as the probability of default (Yes or No default). For reporting purposes, we treat this as a binary variable: **Default = Yes** means a default occurs (high credit risk), and **Default = No** means no default (low risk).

Assumptions for my Network Structure

I assume that *IncomeLevel*, *EmploymentStatus*, and *CreditHistory* are the direct causes influencing the *DefaultRisk*. In other words, these three factors are treated as parent nodes, and *DefaultRisk* is their child node. I will assume the parent factors are mutually independent **a priori** (which is a simplifying assumption for illustration). The resulting structure is a classic "naive Bayes" configuration, where multiple evidence variables point into a single target variable. The diagram below shows the network structure:



This structure encodes my domain understanding: a person's income, employment, and credit history all have a direct impact on their likelihood of defaulting, and I assume no additional direct relationships among these factors themselves (for instance, I are not explicitly modeling any correlation between Income and EmploymentStatus in the network; any such correlation would be indirectly reflected through their influence on DefaultRisk).

In a more elaborate model, one could add links or additional nodes (for example, **EconomicIndicator** or **Debt-to-Income Ratio**).

Prior and Conditional Probabilities

After defining the structure, we need to specify the probability tables. For this illustration, we will assign hypothetical probabilities in a consistent manner.

Prior Probabilities (Root Nodes): Since *IncomeLevel*, *EmploymentStatus*, and *CreditHistory* have no parents, we define their prior distributions based on notional population statistics:

- **IncomeLevel**: P(IncomeLevel = High) = 0.6, P(IncomeLevel = Low) = 0.4. (We assume 60% of applicants have high income and 40% low income, given our lending context.)

- **EmploymentStatus**: P(EmploymentStatus = Employed) = 0.7, P(EmploymentStatus = Unemployed) = 0.3. (Perhaps 70% of applicants are currently employed.)

- **CreditHistory**: P(CreditHistory = Good) = 0.8, P(CreditHistory = Poor) = 0.2. (We assume most applicants have a good credit history, while 20% have a poor record.)

For clarity, we can tabulate these priors:

IncomeLevel	Probability
High	0.6
Low	0.4

EmploymentStatus	Probability
Employed	0.7
Unemployed	0.3

CreditHistory	Probability
Good	0.8
Poor	0.2

Conditional Probability Table for DefaultRisk: The *DefaultRisk* node has three parents, so we must define P(Default = Yes | IncomeLevel, EmploymentStatus, CreditHistory) for each combination of parent values. (For completeness, P(Default = No | ...) = 1 – P(Default = Yes | ...).)

Based on typical credit risk reasoning, we expect that a borrower with *Low* income, *Unemployed* status, and *Poor* credit history has a very high probability of default. Conversely, someone with *High* income, *Employed*, and *Good* history should have a low default probability. We assign the conditional probabilities as follows:

IncomeLevel	EmploymentStatus	CreditHistory	P(Default = Yes)	P(Default = No)
High	Employed	Good	0.10	0.90
High	Employed	Poor	0.50	0.50
High	Unemployed	Good	0.40	0.60

High	Unemployed	Poor	0.80	0.20
Low	Employed	Good	0.30	0.70
Low	Employed	Poor	0.70	0.30
Low	Unemployed	Good	0.60	0.40
Low	Unemployed	Poor	0.95	0.05

These numbers are hypothetical but chosen to be consistent with domain expectations: any presence of a risk factor (low income, unemployment, or poor history) increases the default probability. For instance:

- A **best-case profile** (High income, Employed, Good history) has only a 10% chance of default (so 90% chance of no default).

- A **worst-case profile** (Low income, Unemployed, Poor history) is assigned a 95% chance of default. We assume it's extremely likely such a borrower will default, though not absolutely certain.

- Each negative factor contributes to higher risk. For example, comparing High/Employed/Good (10% default) to High/Employed/Poor, the poor credit history raises risk to 50%. Likewise, High/Unemployed/Good is 40%, reflecting that unemployment alone significantly raises risk even if income was previously high. These conditional probabilities would ideally come from historical data or expert elicitation in a real model.

With the prior tables and this Conditional Probability Table (CPT) for DefaultRisk, the Bayesian network is fully specified. We could use it to compute any joint or conditional probability of interest regarding these four variables.

Inference procedure and the algorithm used

To answer probabilistic queries from the network, we can use the variable elimination algorithm (an exact inference method). The main idea is to systematically eliminate non-query variables by summing them out, rather than trying to enumerate the entire joint distribution. This leverages the independence structure to simplify. The process is as follows:

1. **Identify the query and evidence:** Decide which probability we want to compute. For example, *P(Default = Yes | EmploymentStatus = Employed, CreditHistory = Good)* is a query asking for the probability of default given that the applicant is employed and has a good credit history. Here, *Default* is the query variable and *EmploymentStatus*,

CreditHistory are evidence variables (observed).

2. **Collect relevant factors:** Write down the probability factors from the network that involve these variables. Each CPT or prior is a factor. In my case, the factors would be:

    - Factor for IncomeLevel (prior)
    - Factor for EmploymentStatus (prior, but we will incorporate evidence)
    - Factor for CreditHistory (prior, incorporate evidence)
    - Factor for DefaultRisk (conditional on Income, Employment, History)

3. **Apply evidence:** Whenever a variable is known (evidence), we can **reduce** the factors by that evidence. Essentially, we fix the evidence variable to its observed value in all factors.

    For example, for EmploymentStatus = Employed, we restrict the Employment factor such that P(Employment = Employed) = 1 (and Unemployed = 0 for the sake of calculations), or directly include the 0.7 value in computations.

    Similarly, for CreditHistory = Good, use P(CreditHistory = Good) = 0.8. We also restrict the DefaultRisk CPT to only the entries where CreditHistory = Good and EmploymentStatus = Employed (which effectively picks out the rows highlighted in the table above for those values, still as a function of IncomeLevel).

- **Eliminate hidden variables:** Identify the variables that are not queried or given as evidence, and sum them out one by one. In our example query, the only non-evidence, non-query variable is *IncomeLevel* (we are querying Default, given EmploymentStatus and CreditHistory evidence). So we will eliminate IncomeLevel.

    - To eliminate IncomeLevel, we **multiply** all remaining factors that include IncomeLevel, then sum over IncomeLevel values. This means we take the DefaultRisk CPT (which depends on IncomeLevel) and multiply it by the IncomeLevel prior. This yields a **combined factor** that no longer has Income as a variable (because we will sum over Income = High and Low).

    - Mathematically, this step computes:
      Sum of
      (IncomeLevelP(IncomeLevel))P(Default=Yes∣IncomeLevel,Emp=Employed, Hist=Good)

      This summation yields an unnormalized probability for Default=Yes given the evidence. We would do the same for Default=No if needed (or directly compute normalization as the next step).

- **Compute the result and normalize:** After eliminating Income, we will have a factor that gives P(Default, evidence). Since we want *P(Default = Yes | evidence)*, we take the portion of the factor where Default = Yes, and divide by the sum of probabilities for Default Yes and No (which is P(evidence), ensuring the distribution sums to 1). This normalization yields the final conditional probability. In variable elimination, if we eliminate all non-query variables, the remaining factor (after normalization) directly gives the answer for the query distribution.

By eliminating variables, we avoid enumerating all combinations explicitly, which is much more efficient when the network is large. In my small network, we can also solve queries by brute-force summation, but variable elimination follows a more scalable pattern.

Inference Scenarion 1

Query 1: *P(Default = Yes | EmploymentStatus = Employed, CreditHistory = Good)*.

- **Step 1:** Query variable = Default, evidence = {EmploymentStatus = Employed, CreditHistory = Good}.

- **Step 2:** Relevant factors: P(IncomeLevel), P(EmploymentStatus), P(CreditHistory), P(Default | Income, Employment, History).

- **Step 3 (apply evidence):** We set EmploymentStatus = Employed and CreditHistory = Good. We will incorporate these by using their values: P(Employment=Employed) = 0.7, P(CreditHistory=Good) = 0.8. (When computing the conditional probability, these evidence terms will appear in both numerator and denominator and cancel out, as we'll see.) We also restrict the Default CPT to the sub-table where EmploymentStatus is Employed and CreditHistory is Good:

  From the DefaultRisk CPT above, the relevant entries are:

  - Income High, Employed, Good: P(Default=Yes) = 0.10

  - Income Low, Employed, Good: P(Default=Yes) = 0.30

- (We took the rows matching Employed & Good. We will still consider Income being High or Low in the elimination step.)

- **Step 4 (eliminate IncomeLevel):** We sum out IncomeLevel. There are two possible IncomeLevel values (High or Low). Using the restricted CPT:

The joint probability of Default = Yes, EmploymentStatus = Employed, and CreditHistory = Good is calculated as follows:

We sum over all possible values of IncomeLevel:

P(Default = Yes, EmploymentStatus = Employed, CreditHistory = Good)
= P(EmploymentStatus = Employed) x P(CreditHistory = Good) x
[P(IncomeLevel = High) x P(Default = Yes | IncomeLevel = High, EmploymentStatus = Employed, CreditHistory = Good)
+ P(IncomeLevel = Low) x P(Default = Yes | IncomeLevel = Low, EmploymentStatus = Employed, CreditHistory = Good)]

Plugging in values:

For Income = High: P(Income=High) x P(Default=Yes | High, Employed, Good) = 0.6 x 0.10 = 0.06.

For Income = Low: P(Income=Low) x P(Default=Yes | Low, Employed, Good) = 0.4 x 0.30 = 0.12.

Plugging in values:

- For Income = High: P(Income=High) x P(Default=Yes | High, Employed, Good) = 0.6 x 0.10 = 0.06.
- For Income = Low: P(Income=Low) x P(Default=Yes | Low, Employed, Good) = 0.4 x 0.30 = 0.12.

- Summing these: $;0.06 + 0.12 = 0.18$. (The factors P(Emp=Employed) and P(Hist=Good) were omitted in this sum because they are common to all terms; if we included them, we would factor 0.7 * 0.8 in and then later divide by the same product in normalization. We can effectively ignore those constants when computing the ratio, as they cancel out.)

- **Step 5 (normalize):** The sum 0.18 is the joint probability of Default=Yes *and* the evidence. To get the conditional probability P(Default=Yes | evidence), we need to divide by the total probability of the evidence. In practice, we can compute the probability of *evidence* (Employed and Good history) by doing a similar sum for both Default outcomes.

To calculate the probability of the evidence, we use:

P(evidence) = P(Default = Yes, evidence) + P(Default = No, evidence).

Now, let's calculate the part for when Default = No:

- If Income is High:
  Multiply 0.6 by the probability of not defaulting given High Income, Employed status, and

Good Credit History.
That is: 0.6 × 0.90 = 0.54

- If Income is Low:
  Multiply 0.4 by the probability of not defaulting given Low Income, Employed status, and Good Credit History.
  That is: 0.4 × 0.70 = 0.28

Add those two results:
0.54 + 0.28 = **0.82**

We already know that the probability of Default = Yes and the same evidence is 0.18.

So:
P(evidence) = 0.18 + 0.82 = 1.00

This total remains consistent even if we normalize using likelihoods like 0.7 and 0.8.

Now, to find the probability of default given the evidence (Employed and Good Credit History), we calculate:

P(Default = Yes | Employed, Good History) = 0.18 / (0.18 + 0.82) = 0.18

This means there is an 18% chance of default in this scenario.

Thus, given an employed applicant with a good credit history (and no information about income), the model predicts an 18% probability of default. This makes sense because, even though employment and good credit are favorable, we haven't specified income, the probability of default is intermediate. (If we further knew the income was high or low, the probability would adjust to 10% or 30% accordingly from the CPT.)

The above manual calculation illustrates variable elimination: we eliminated Income by summation.

Inference scenario 2

To showcase another inference, let us consider

Query 2: *P(CreditHistory = Poor | DefaultRisk = Yes)*. This is a diagnostic query, we observe that a default happened (Default = Yes) and want to infer the probability that the borrower had a poor credit history. In other words, given that someone defaulted, how likely is it that their credit history was poor? We can use the network to answer this by applying Bayes' rule and summing out the other factors (Income and Employment). The computation would be:

- **Numerator:** P(History=Poor, Default=Yes) = sum over Income, Employment [P(Income) x P(Employment) x P(History=Poor) x P(Default=Yes | Income, Employment, History=Poor)].

- **Denominator:** P(Default=Yes) = sum over Income, Employment, History [P(Income) x P(Employment) x P(History) x P(Default=Yes | Income, Employment, History)].

Then P(History=Poor | Default=Yes) = Numerator / Denominator. Rather than plug in all 8 combinations here, we describe it qualitatively: we weigh each scenario where history is Poor by its prior probability and default likelihood, and divide by the overall probability of default. After performing this calculation (the sums can be done with the CPT values above), we find:

P(CreditHistory = Poor | Default = Yes) ≈ 0.38 (approximately 38%).

This result implies that if a default occurs, there is a 38% chance the borrower had a poor credit history. In other words, defaults are more likely to be associated with poor history than the 20% base rate, which makes intuitive sense, observing a default updates our belief toward a higher likelihood of a problematic credit history.

I have attached the Python implementations of the above 2 queries and comments to explain my approach in the implementation.