ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster Universitario en Big Data

TRABAJO FIN DE MASTER

Framework Development for Causal Inference with Machine Learning: Exploring Manufacturing and Software Use Cases

Autor: Daniel Bicand Fernández

Director: José Luis Gahete Diaz

Madrid

2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Framework Development for Causal Inference with Machine Learning: Exploring

Manufacturing and Software Use Cases en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Daniel Bicand Fernández Fecha: 06/ 07/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Jose Luis Gahete

Firmado digitalmente por Jose Luis Gahete
Fecha: 2025.07.10 12:03:46 +02'00'

Fdo.: José Luis Gahete Diaz Fecha: 09/ 07/ 2025

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster Universitario en Big Data

TRABAJO FIN DE MASTER

Framework Development for Causal Inference with Machine Learning: Exploring Manufacturing and Software Use Cases

Autor: Daniel Bicand Fernández

Director: José Luis Gahete Diaz

Madrid

2025

Memory Index

CHAPTER 1. INTRODUCTION

1.1 MOTIVATION: WHY CAUSAL INFERENCE MATTERS BEYOND PREDICTION

Machine-learning systems excel at spotting patterns, yet most industrial and product decisions hinge on answering what-if questions: Will shortening a paper-machine drying cycle cut waste without compromising tensile strength? Will a new onboarding flow actually raise seven-day retention, or merely correlate with it? Correlation-based models cannot separate genuine levers from noisy covariates, so organizations risk over-optimizing vanity metrics or deploying changes that backfire once confounders shift.

Digital transformation has amplified this tension. Modern manufacturing plants stream thousands of sensor variables per minute, while software teams ship code behind feature flags dozens of times a day. The sheer volume, velocity and dimensionality of data make controlled trials expensive, partial, or logistically impossible. Decision-makers therefore look to causal inference—augmented by scalable ML algorithms—to exploit observational data when experimentation is limited and to extract richer insights from the experiments they can run.

At the same time, regulators and customers are demanding transparent explanations of algorithmic decisions. Causal models provide narratives grounded in intervention logic rather than opaque correlations, aligning with emerging standards on AI governance, fairness and root-cause accountability.

Taken together, these drivers create an urgent need for a practical framework that embeds causal reasoning in everyday analytics for both physical and digital products.

1.2 RESEARCH GAP AND OBJECTIVES OF THE THESIS

Academic work on causal inference has surged, yet it remains fragmented: papers on Double Machine Learning target economists, physics-informed approaches serve process engineers, and product-analytics platforms market proprietary A/B engines with little methodological detail. Few resources walk practitioners from problem formulation through deployment across multiple verticals. Even fewer show how to reuse the same logic tree—hypothesis, DAG, identification, estimation, validation—in domains as disparate as papermaking and mobile apps.

Meanwhile, machine-learning research often treats causal discovery and effect estimation as standalone algorithmic challenges, overlooking the organizational steps that determine whether a project is even causally feasible (e.g., presence of an actionable treatment, data coverage, ethical constraints). Practitioners therefore face two symmetrical pain points: cutting-edge models without operational guidance, and well-meant checklists that ignore the latest modelling advances.

This thesis targets that dual gap. Its core objective is to develop and validate a reusable end-to-end framework for causal inference with ML, then demonstrate its versatility through two concrete industries—manufacturing and software. In doing so, it aims to provide both (i) a structured decision pathway that any analytics team can adopt and (ii) implementation blueprints that connect modern algorithms to real business questions.

1.3 CONTRIBUTIONS & SCOPE

This thesis sets out as an experiment in portability: can one practical framework guide causal reasoning when the "product" is a roll of tissue today and a mobile-app screen tomorrow? Rather than presuming a universal formula, it assembles a tentative playbook—checking for actionable treatments, sketching a defensible DAG, choosing an estimator, running refutation tests—and explores how far that sequence can travel across two contrasting product-making landscapes, one grounded in physical machinery and the other in digital experience design.

The journey is mapped in four stages. Chapter 2 distils the minimum causal theory needed to recognize interventions and counterfactuals; Chapter 3 turns those ideas into an operational guide of checklists and decision trees; and Chapters 4 and 5 try the guide in practice, first on a sensor-rich manufacturing line, then on a UI change measured by clickstreams and retention curves. Deep proofs, full code, and frontier topics such as network interference or real-time control are left to appendices and future work, keeping the main text focused on what can be learned—and perhaps reused—here and now.

CHAPTER 2. FOUNDATIONS OF CAUSAL INFERENCE

The goal of this chapter is to give the reader enough conceptual and practical footing to follow the framework in later chapters.

2.1 CORRELATION VS CAUSATION:

Correlation and causation are two concepts that are often used interchangeably, but they refer to different types of relationships between variables. Correlation refers to a statistical relationship between two variables, where the occurrence of one variable is related to the occurrence of another variable. Causation, on the other hand, refers to a relationship where one variable directly influences or causes a change in another variable. Whilst causation implies correlation between two variables, the inverse, need not hold true. It is possible that variables are correlated but there is no causal relationship between these.

Machine learning models are designed to identify patterns and relationships in data, and they rely heavily on association to make predictions and generate insights. These models use statistical algorithms to identify correlations between variables, and predict outcomes based on these correlations. However, while machine learning models can be effective at identifying associations between variables, they do not necessarily capture causation.

To see why the distinction matters in practice, consider three familiar pitfalls:

1. **Spurious coincidence — "Margarine divorces."**

Tyler Vigen's tongue-in-cheek chart shows a 99 % correlation between divorce rates in Maine and U.S. margarine consumption. The link is purely accidental; no plausible mechanism connects the two variables. Acting on such a pattern would be chasing noise.

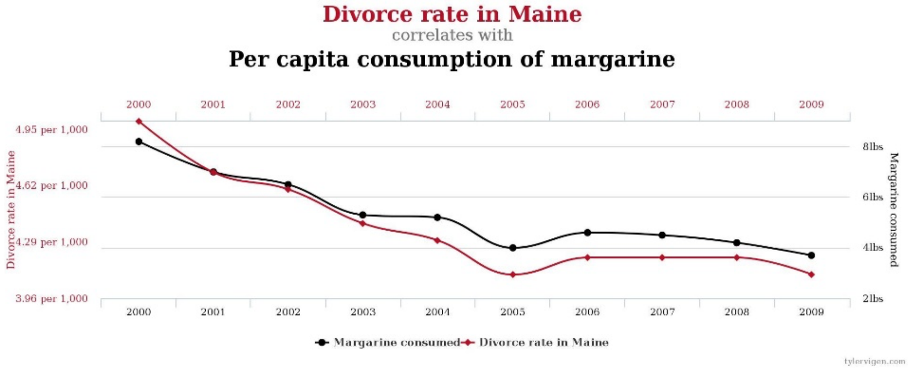


Figure 1. Correlation between divorce rate in Maine and per capita consumption of margarine [3]

2. Unknown direction — Corruption ↔ GDP.

Cross-country data reveal that low GDP and high corruption scores move together, but the arrow of influence is unclear: does corruption depress growth, or does poverty foster corruption—or both? Without a model that pins down direction, policy could target the wrong lever.

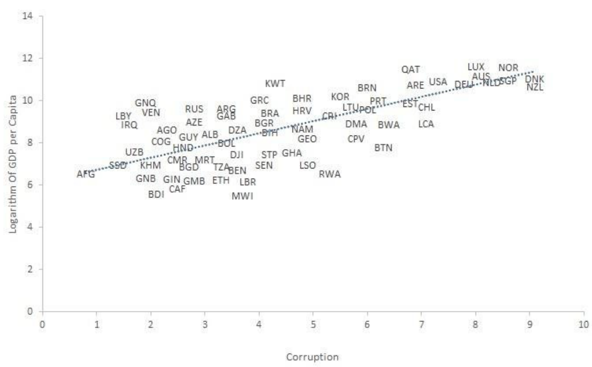


Figure 2. Correlation between corruption and GDP per capita of a country [4]

3. Hidden common cause (confounder) — Ice-cream sales and shark attacks.

Both rise in summer, yet banning ice-cream would not make beaches safer. The lurking variable is warm weather, which drives swimmers into the sea and customers to the ice-cream stand. Failing to account for this confounder opens a back-door path and produces a misleading association.

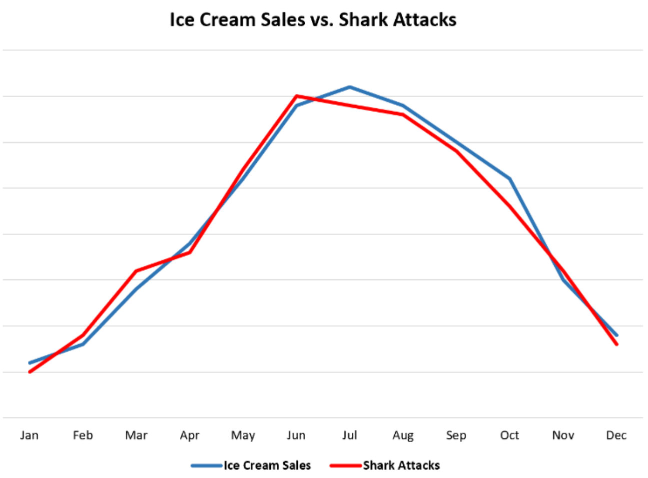


Figure 3. Correlation between shark attacks and ice cream sales

Throughout the rest of the thesis, every proposed change—be it lowering a drying-cylinder temperature or launching a new onboarding screen—is treated as a treatment whose impact must be separated from coincidence, reverse causality, and confounding. The framework introduced in later chapters therefore starts by asking three simple questions:

- Could the pattern be a random coincidence?
- Might the causal arrow point the opposite way?
- Is there an unmeasured variable opening a back-door path?

2.2 HOW CAUSAL RELATIONS ARE INFERRED:

There are multiple approaches to infer causal relationships between variables. The easiest way to understand the underlying logic behind causal inference is by examining the formula for the Average Treatment Effect (ATE):

$ATE = E[Y_{T=1} - Y_{T=0}]$. Y = Outcome T = Treatment

The ATE formula essentially calculates the difference in outcomes between two treatment variables. For example, let's consider an outcome where Y=1 represents a person getting cured and Y=0 represents a person not getting cured. Additionally, let´s assume T=1 represents giving a person a treatment and T=0 represents giving them a placebo. By comparing the outcomes of two groups separated by treatment variable, we can infer if the treatment is causing the desired outcome of curing people. The following graph illustrates this example.

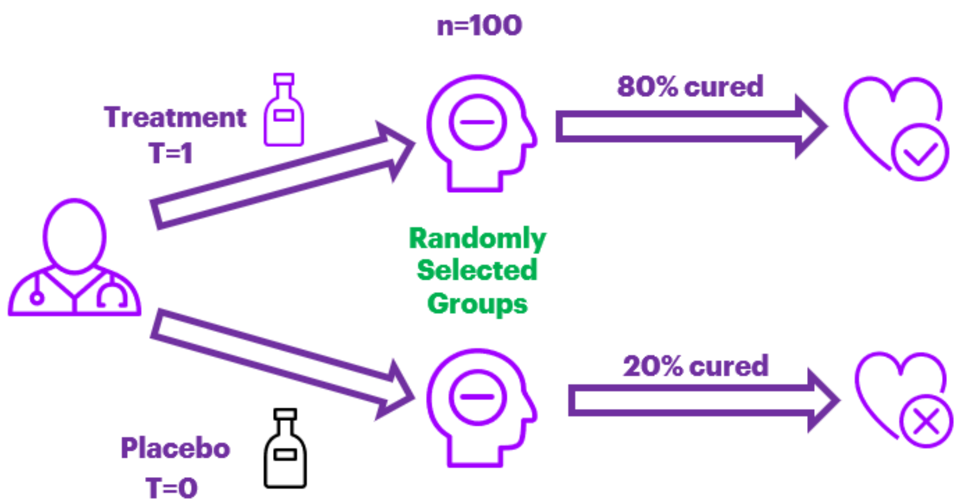


Figure 4. Treatment and Placebo diagram

By giving different treatments to two samples of the population of size 100, it is observed that 80% of the sample of people that received the actual treatment is cured vs the 20% of the people that received the placebo treatment. This implies that there is a difference of 60% between both groups and the ATE is of 0.6, which is significantly high so we could conclude that the treatment is causing people to get cured.

Randomized Controlled Trials (RCTs) remain the reference method for establishing causality because random assignment breaks every systematic link between a treatment T and potential confounders. If coincidence assigned all women to T=1 and all men to T=0, any measured difference might come from gender rather than the treatment itself; true randomization avoids that imbalance and lets us attribute outcome gaps squarely to T.

Yet full randomization is not always feasible. Cycling a paper machine through every recipe variant is costly and time-consuming, and some legacy software stacks cannot expose each user to a clean feature-flag split. When RCTs are off the table, we turn to observational data—sensor streams, production logs, click trails—and ask whether clever design (matching, back-door adjustment, instrumental variables) can approximate the missing experiment. These alternatives carry risks of bias and confounding, but, handled carefully, they still yield actionable insight for process optimization and product improvement. Section 2.4 surveys the algorithmic toolkit that makes such "quasi-experiments" possible in both the physical and digital domains explored in this thesis.

2.3 HOW CAUSAL STRUCTURES ARE REPRESENTED:

Causal mechanisms are most easily reasoned about when they are drawn. The standard language is a **Directed Acyclic Graph (DAG):** nodes stand for variables, arrows for putative causal influences. Two structural rules give a DAG its name and power.

- **Directedness –** every arrow points from cause to effect, breaking the symmetry that pure correlations permit.
- **Acyclicity** – the arrows may never loop back to the same node, preventing a variable from (directly or indirectly) causing itself and preserving a coherent time-ordering.

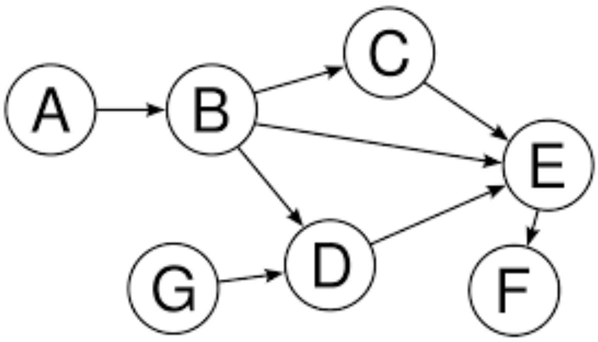


Figure 5. Example of a Directed Acyclic Graph (DAG)

Within this grammar three micro-patterns dominate:

- **Fork (confounding) A ← Z → B** — a common cause Z creates a back-door path between A and B.
- **Chain (mediation) A → M → B** — the effect of A on B flows through mediator M.

- **Collider (v-structure) A → C ← B** — arrows collide at C; conditioning on the collider opens, rather than blocks, a path between A and B.

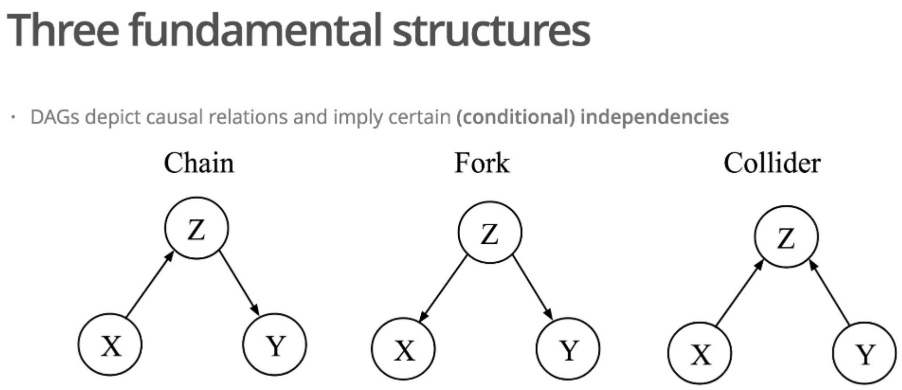


Figure 6. Chain, Fork and Collider structures in DAGs

Recognising forks, chains, and colliders allows us to decide which variables to control for and which to leave alone when estimating an Average Treatment Effect. In the chapters that follow, a first draft of the DAG—usually sketched with domain experts before any code is written—becomes the shared blueprint that guides identification strategies, algorithm choice, and ultimately the credibility of the causal claims.

2.4 HOW CAUSAL STRUCTURES PROVIDE INSIGHT:

Having drawn a tentative DAG, the next question is how to read an effect off that graph. Three classic routes appear again and again in practice:

Tactic	Intuition	Practical clues
Back-door adjustment	Find a set Z that blocks every path running *backwards* from treatment T to outcome Y. Condition on Z and the remaining association equals the causal effect.	In manufacturing, ambient temperature or raw-material grade often lands in Z; in software, user tenure or device class.

Front-door path	Sometimes no useful Z exists, yet a mediator M carries the whole effect of T on Y. If we can model T->M and M->Y cleanly, we recover the effect indirectly.	Rare but handy: *coupon* → *basket size* → *revenue* in e-commerce; *pressure set-point* → *sheet moisture* → *defects* on a paper line.
Instrumental Variable (IV)	A variable Z nudges T but influences Y **only** through T. IV logic treats Z as Mother Nature's randomiser.	Batch-to-batch feedstock moisture shifts T="dryer power" but not quality directly; hash-based user assignment nudges T="new UI" with no appeal to the user.

2.5 LIMITS OF CAUSAL INFERENCE WITH OBSERVATIONAL DATA:

Causal methods promise actionable insight, but only under assumptions that are easy to violate—especially when randomized trials are off the table.

- **Selection bias:** If treatment and control groups differ systematically (e.g., only senior operators use the new recipe; only power-users see the beta UI), estimated effects may reflect that imbalance, not the treatment itself. Diagnostics such as sample-ratio-mismatch checks and propensity-score balance help spot the problem, but cannot always fix it.
- **Unmeasured confounding:** Variables that influence both T and Y but stay off the data lake (maintenance culture, social influence) open back-door paths the analyst cannot block. Sensitivity analyses and instrumental-variable designs are partial remedies, not panaceas.
- **Measurement error & data quality:** Noisy sensors or mis-tagged events blur treatment assignment or outcomes, biasing estimates much like

unmeasured confounding. Both case studies include a data-quality audit to illustrate the impact.

- **Generalizability:** A causal effect estimated on one production line or one user segment may not transport to others. External validity demands replication, domain expertise, and sometimes explicit transportability analysis—topics noted but not solved in this thesis.

Recognizing these limits early shapes every step of the framework: which variables must be logged, which identification tactic is plausible, and how cautiously results should be applied beyond the original study setting.

2.6 GENERAL APPROACHES TO INFERRING CAUSAL RELATIONS WITH OBSERVATIONAL DATA:

Before any back-door adjustment or instrumental-variable trick can be applied, we need at least a working sketch of the underlying DAG. In domains such as papermaking or UI design parts of that sketch come from subject-matter intuition—engineers know steam pressure precedes moisture; product managers know an onboarding screen precedes retention. Yet even seasoned experts rarely see the whole picture, especially when hundreds of sensor channels or user events are in play. Data-driven structure learning fills those gaps by proposing, ranking, and refining candidate graphs that are compatible with the observed statistics.

The task is formidable: with d variables there are super-exponential many possible DAGs, and scoring each one exactly is often intractable. Modern algorithms therefore rely on heuristics that trade mathematical guarantees for practical speed and robustness. Four families have proved particularly useful:

The four groups of approaches to solutions are:

Constraint-based methods:

These algorithms begin with the most inclusive hypothesis—every variable is connected to every other—and then delete edges that contradict the data. Deletions are triggered by statistical independence tests: if two variables remain independent after we control for a third, there is no direct arrow between them. Repeating this logic across many variable triples whittles the dense graph down to a sparse skeleton, after which a set of logical rules orients the surviving edges. The PC and FCI procedures are longstanding exemplars. They work quickly on moderate problem sizes and are transparent enough for domain experts to follow, but their accuracy hinges on the reliability of the underlying independence tests.

Score-based methods:

Score-based methods for causal discovery aim to identify the best causal structure of a given system by iteratively generating candidate graphs, evaluating how well each one explains the data, and selecting the best one.

These methods start with an initial graph structure, such as a fully connected or empty graph, and then modify the graph by adding, removing, or reversing edges. After each modification, the resulting graph is evaluated based on how well it explains the observed data, using some criterion or scoring function.

Common scoring functions are the Bayesian Information Criterion (BIC), the Akaike Information Criterion (AIC), or the Maximum Likelihood Estimation (MLE).

One well-known example of a score-based method for causal discovery is the Greedy Equivalence Search GES algorithm.

The GES algorithm starts with an empty graph and iteratively adds, removes, or reverses edges based on a set of conditional independence tests until it arrives to a high score where the structure had converged on the best causal structure for the system. The score-based approach is a powerful tool for causal discovery, but it is resource intensive, meaning that it is computationally expensive, and it requires a careful tuning of the scoring function to converge to an optimal solution.

Functional methods:

Functional methods for causal discovery aim to identify the underlying causal relationships between variables by analyzing their functional dependencies. This involves estimating the parameters of a statistical model that describes the relationship between variables, and then using this model to infer the causal structure of the system. Functional approaches also use a score function to estimate the goodness of the solution, but the mechanics of functional methods are different from score-based methods. Rather than searching through candidate graphs to find the best fit to the data, functional methods make use of the distributional imbalances present in the data to detect the causal connections between variables.

The Linear Non-Gaussian Acyclic Model (LiNGAM) algorithm is a classic functional method that uses the non-Gaussianity of the data to identify the causal structure. LiNGAM exploits the non-Gaussian nature of the data to determine the probable causal relationships between variables. More specifically, when two variables have a non-Gaussian correlation, it implies a potential direct causal connection between them. On the other hand, if two variables have a Gaussian correlation, it indicates a possible indirect or confounding relationship between them.

Gradient-based methods:

The newest wave of structure-learning algorithms turns the combinatorial search for a DAG into one smooth optimisation problem. The flagship example, NOTEARS (Zheng, Aragam, Ravikumar & Xing), fits edge weights while adding a differentiable penalty that forbids cycles; standard gradient-descent updates then drive the solution toward a sparse, acyclic graph. Because the whole objective is differentiable, GPUs and other deep-learning tooling can be brought to bear, a feature we exploit in Chapter 4 via the CausalNex implementation contributed by QuantumBlack. Gradient methods rarely guarantee the single "best" graph, but they scale gracefully and inherit the optimisation tricks that power modern image and language models.

Recent growth in data volume and sensor density has pushed traditional discovery algorithms to their computational limits, motivating a new generation of large-scale learners that try to short-circuit the combinatorial search altogether.

How NOTEARS works:

1. **Data and weights:**

Place the data in a matrix X (n observations × d variables) and initialise a weight matrix W (d × d), where each entry wij proposes a causal influence from variable j to i.

$$X \in \mathbb{R}^{n\times d}$$
$$W \in \mathbb{R}^{d\times d}$$

2. **Fit term:**

Compute F(W), the squared error between X and XW, plus an ℓ_1 penalty that pushes minor edges toward zero.

$$F(W) = \frac{1}{2n} \| X - XW \|_F^2 + \lambda \| W \|_1$$

3. **Acyclicity term:**

Compute:

$$h(W) = Tre^{W\odot W} - d$$

a smooth surrogate that equals zero only when W encodes a cycle-free graph.

4. **Joint objective:**

Minimise L(W) with gradient descent (or Adam). The fit term drives explanatory power; the acyclicity penalty steers the solution away from cycles.

$$L^{\rho}(W,\alpha) = F(W) + \frac{\rho}{2}|h(W)|^2 + \alpha h(W)$$

5. **Outcome:**

The optimiser returns a sparse, directed graph compatible with the data and free of cycles. In Chapter 4 we use the QuantumBlack CausalNex implementation of NOTEARS to learn the manufacturing case study's structure.

New large-scale learners:

These methods harness deep-learning architectures—most often transformers—pre-trained on vast libraries of synthetic causal graphs. Once trained, the network can propose an entire diagram in a single forward pass, a strategy known as amortised discovery. Early studies show promising accuracy on problems with thousands of variables or on fast-streaming telemetry, suggesting a route around the memory and run-time walls that stop classical algorithms. The catch is cost: pre-training demands heavy compute, and the tooling is still research-grade, so for the moment these models remain "promising but experimental" rather than part of the everyday analyst's kit.

CHAPTER 3. GENERAL FRAMEWORK FOR CAUSAL INFERENCE WITH MACHINE LEARNING

This chapter distills a portable playbook for causal inference that can be reused in any data-rich environment—whether the "product" is a roll of tissue paper or a mobile-app screen. Rather than prescribing a domain-specific recipe, we draw on both the academic literature and field practice to outline the essential steps, checkpoints, and design choices that let analysts turn observational data into credible what-if answers. Manufacturing and software will serve as running examples, but only to illustrate how the same logic tree—hypothesis → DAG → identification → estimation → validation—travels across contexts.

The resulting framework is intentionally modular: it can plug into different algorithms, data volumes, and organizational constraints while still enforcing the causal reasoning needed to separate actionable levers from spurious correlations.

3.1 HIGH-IMPACT APPLICATION AREAS ACROSS DOMAINS:

Whenever organizations ask "what will happen if we pull this lever?" yet cannot—or will not—run a clean experiment, causal inference becomes the decisive lens. In practice, the same four kinds of questions appear again and again, whether the backdrop is a kilometer-long production line or a cloud-hosted mobile service.

Causal inference pays off whenever a team must decide which lever to pull and how hard to pull it—rather than just describing patterns. The specifics of the lever vary by industry, but three ingredients reappear everywhere:

Ingredient	Why it matters
Intervention-ready levers	A knob you can actually turn (e.g., spindle speed, paywall copy, ad budget). Without a lever, causal questions collapse into passive "forecasting."
Rich observational exhaust	Logs, sensors, or records that capture both when/where the lever moved and what else was happening. Depth beats width: timestamps, versions, and context fields are often more valuable than extra rows.
A plausible path to identification	Whether via natural experiments, instrumental variables, or careful adjustment sets, you need at least one assumption you are willing to defend in the boardroom—and test in hold-out slices.

Manufacturing: the canonical proving ground:

On a factory floor a single mistuned servo can choke throughput for hours, yet its fingerprints are buried under dozens of coupled feedback loops. Operators develop theories — "the night shift always sees more jams"— that are rich in anecdotes and poor in counterfactuals. Causal discovery pins those stories to data: a spike in vibration amplitude no longer merely co-occurs with scrap; it sits on a directed edge that explains how the fault propagates and which intervention —slowing a feed rate, retiming a tool change—will break the chain.

Because every extra minute of downtime has an explicit dollar tag, manufacturers feel the cost of misdiagnosis immediately. That urgency explains why we use a production line as the running example throughout this chapter and return to a concrete case study in Chapter 4. It also foreshadows the twin obstacles tackled in 3.2: hidden confounders lurking in exogenous inputs (Was today's batch of raw material subtly different?) and the computational chaos that arrives when hundreds of sensors balloon the search space of possible graphs.

SaaS: the same logic behind a softer façade

Swap conveyor belts for software deployments and the causal puzzle stays intact. A product manager rolls out an algorithmic recommendation tweak to half the user base. A/B tests are the gold standard, yet network spillovers, release embargoes or sheer velocity often make fully controlled trials impractical. Here, regression-kink designs at API version thresholds or synthetic control cohorts recreate the missing counterfactual. The win is measured not in parts-per-million yield but in retention curves and lifetime value—but the epistemic game is identical: learn which knob truly moves the metric, not which trend line happens to fluctuate in parallel.

3.2 BUILDING THE CAUSAL STRUCTURE:

There are several things that one must keep in mind when building any causal structure given a set of data. To build a robust and resilient causal structure, a two-step approach that tries to minimize the bias and errors in causal structures defining any manufacturing process is going to be explained. The first step focuses on the reduction of the error introduced by confounding effects. The second step focuses on the reduction of the run-time and complexity of the algorithm that oversees building the causal structure. In chapter 4, a real use-case of this methodology is provided.

3.2.1 THE DANGERS OF CONFOUNDERS:

Confounders are variables that affect both the cause and the effect variables in a study. Since these affects simultaneously both variables, if these are not considered, the causal analysis can be greatly biased. Therefore, it is important to account for confounders in causal inference to ensure that the causal relations extracted from the data are representative of the reality and it is not just a relationship caused because there is a hidden confounder that hasn't been considered.

Together with the shark and ice cream example, there is another example that is widely used to explain the confounders effect. This example states the following: sleeping wearing shoes is causing people to wake up with a headache.

Figure 12. Confounder variable example

A statistical analysis was made to find if sleeping wearing shoes was causing people to wake up with headaches. When observing just the cause and effect, it seems to be clear that sleeping with shoes causes headache, but the reason both variables are related is because there is a common confounder, in this case drinking alcohol the previous night. If this confounder is not considered, then the inferences obtained conducting causal inference are incorrect. This example clearly shows the importance of including all the possible confounders in a study when conducting causal inference.

The way to fight against the negative effect of confounding variables in a manufacturing system is to enclose the system by adding all the variables and features that affect the system in any possible way. When including all the variables in a system, then, all the system confounders are observed, and the causal structure includes them in the graph removing the confounding effect.

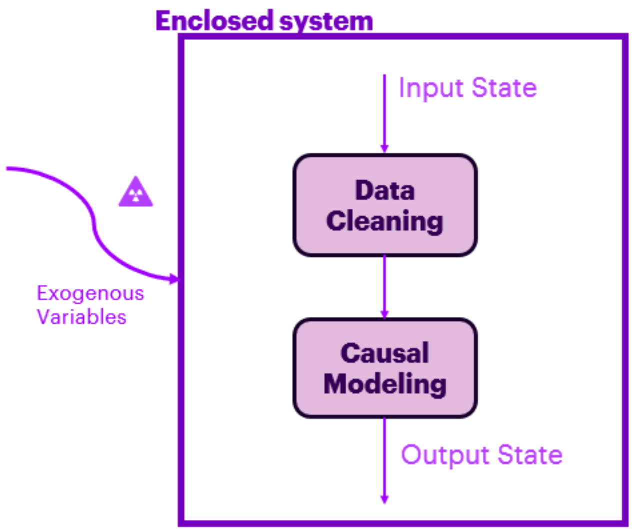


Figure 13. Enclosed system diagram

Figure 11 explains how by enclosing the system, the causal structure built on observational data is more resilient to biases caused by not including confounding variables. There is also a possibility that an exogenous variable, which is not considered as a variable of the system, may be influencing the system in an unobserved manner. For example, imagine you are analyzing a factory that oversees manufacturing chairs. You measure the quality of the chairs at the end of the process and track how all the machines are working. If the quality of the chairs depends solely on how the machines behave, then your analysis is fair. But if how the wood was chopped an important factor in making a good chair, and you don't consider it in your analysis, then you'll have a biased result due to the confounding effect. How the wood was initially chopped is an example of an exogenous variable. By paying attention to these exogenous confounding variables, you can improve your analysis and reduce errors and biases in your results.

Therefore, before building a causal structure based on observational data, it is important to **enclose the system subject to study**. This means including all the system variables and the possible exogenous variables that can have a confounding effect in the target variable. This is the first step of the general methodology for conducting causal inference on manufacturing processes.

3.2.2 THE COMPLEXITY PROBLEM:

As this thesis has explained in chapter 2, algorithms that build causal structures just relying on observational data are complex algorithms that take some time to converge to a solution. The complexity of these algorithms depends on the quantity of nodes (variables) that are forming the graph. This is because most of the algorithms work by analyzing the relationships between variables one to one, so the number of iterations increases exponentially with the number of variables.

Some manufacturing process need large complex machines to work and the amount of sensors that parametrize the system can be really large. In those cases, pre-processing techniques are required to reduce the number of variables. Pre-processing and variable reduction is a typical step that must be done in almost every data science project, but there are differences in what to consider when doing pre-processing for a prediction algorithm and doing it for a causal inference algorithm.

In Chapter Two, the distinction between association and causation in the relationship between variables was explained. In the context of predictive modeling, association relations are sufficient, which means that grouping associated variables during pre-processing does not affect the accuracy of the model, even if the variables are not causally related. One example of this approach is seen in the application of principal component analysis (PCA) to a dataset, where the principal components are formed by combinations of the original variables, and their combinations depend on the original associations between them. However, it is important to note that this approach is not suitable for analyzing causality. In causality it is important to keep the raw variables to understand what is really happening. If different variables are combined, the causation between those variables will be diluted and the whole purpose of the analysis is lost.

So, to obtain un-biased results, it is important to keep in mind that creating artificial variables by combining them can reduce information on causal relationships. Normally, the best way to deal with this situation is with the help of a Subject Matter Expert (SME), we will explain in the next section (3.3) of this chapter how an SME can help to empower causal analysis.

After studying convergence times and the complexity of the NoTEARS algorithm (which is the one that was used for the case study that will be explained in chapter 4), the recommended maximum number of variables to include into the analysis is in the order of less than 100. If more than 100 variables are included, the convergence time breaks and increases exponentially.

3.3 THE IMPORTANCE OF SMEs:

Once the data set is in shape and a discovery algorithm has produced its first directed acyclic graph, the real work of interpreting—and correcting—that structure begins. Algorithms that infer causality from pure observation are solving problems in the NP family: they can check whether a candidate graph fits the conditional independences in polynomial time, yet finding the single "true" graph may take longer than the lifetime of the project. In practice, the optimizer lands on a whole Markov equivalence class—many graphs that explain the data equally well—so the machine's answer is, at best, a plausible draft.

Enter the Subject-Matter Expert (SME). An SME is the engineer who knows the thermodynamics of a dryer section by heart, or the product manager who has watched thousands of users churn at the same onboarding screen. Their domain intuition turns three crude steps into a disciplined review loop.

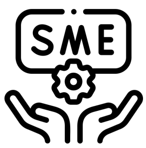


Figure 14. SME icon

Firstly, they can help with the selection of variables and the data pre-processing. As it was explained in last section, cleaning the dataset, and reducing the amount of variables is a complex task that needs precision and knowledge of how the system subject to study works. The SME can help with the selection of important variables and cleaning the independent variables that do not affect the system. It is important the data scientist and the SME work collaboratively, as the data scientist can clean and prune a dataset without losing information, the SME can help and guide the decisions with his knowledge of the system.

It is important to note that the data cleaning and pre-processing phase is common to almost all the use cases, but when the number of variables surpass the hundredths then it becomes a risky phase because eliminating important information can affect the whole causal structure.

Secondly, the SME can also guide the causal structure algorithm to converge to optimal solutions. This is done by providing constrains to the algorithm, therefore decreasing the number of possible solutions and reducing the number of operations (algorithm converges faster). The SME can add constrains to the causal structures in three different ways:

- **Adding a causal relation**: The SME can force a relation to appear in the causal structure. Maybe, it is the case that it is known for sure that the behavior of a variable directly causes the behavior of another variable.

(E.g., Variable X1 is causing variable X2)

Figure 15. Adding a causal relation between X1 and X2

- **Removing a causal relation**: The SME can also remove causal relations in the causal structure. It can be known that a variable is certainly not causing other variables to behave differently.
  (E.g., Variable X1 is not causing variable X2)

Figure 16. Removing the causal relation between X1 and X2

- **Changing causal directions**: The last action an SME can force into a causal structure is to change the direction of causation between two variables. This action is helpful when correcting the problems with Markov equivalent classes. Sometimes, if the NoTEARS algorithm is used to build the causal structure, it can happen that some of the directions are wrongly encoded. It is not frequent because there are not too many combinations to a possible solution that ensures the DAG property and has equivalent Markov classes. The SME can identify some relations that do not make sense and rearrange the direction.

(E.g., Variable X1 is not causing variable X2, it is the other way around)

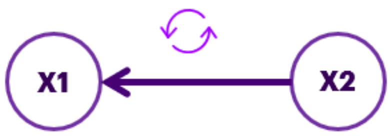


Figure 17. Changing the direction of the causal relation

3.4 STEP BY STEP METHODOLOGY:

In this section (3.4) the thesis is going to explain the overall methodology and the different steps that must be followed to ensure the development of a correct and unbiased causal structure when studying a manufacturing system.

- **Step 1 – Comprehensive data collection:**

Begin by inventorying every process variable, sensor reading, log field, and contextual attribute that could act as a cause, an outcome, or a confounder. Use structured elicitation sessions with subject-matter experts (SMEs) to surface latent drivers and plausible proxies and record the temporal resolution and known gaps of each source. The objective is coverage: a causal claim can only be as complete as the variables observed.

- **Step 2 – Targeted data curation:**

Reduce dimensionality without erasing causal signal. Merge exact duplicates, impute missing values only when the mechanism behind the gaps is understood, and flag features with near-zero variance or obvious data leakage. When the candidate list approaches or exceeds roughly one hundred variables, switch from automated filters to SME-guided pruning to avoid discarding critical confounders. As a rule of thumb, a structure-learning algorithm benefits from at least three thousand high-quality observations, but sample quality outweighs raw count.

- **Step 3 – Initial structure learning:**

Choose a score- or constraint-based discovery method—NoTEARS and GES are common choices—and run it with a reproducible random seed. Persist the resulting adjacency matrix and produce an initial visual rendering of the directed acyclic graph (DAG). At this point the graph represents a statistically plausible explanation of the observed conditional independences, but it is unlikely to be mechanistically perfect.

- **Step 4 – Export-driven graph refinement:**

Invite SMEs to scrutinize the draft DAG. They may (i) force an arrow to exist, (ii) forbid an arrow, or (iii) reverse an implausible direction. Encode these domain constraints in the learning algorithm and re-estimate the structure. Iterate between estimation and review until the graph stabilizes and passes face-validity checks; each loop narrows the search space and accelerates convergence.

- **Step 5 – Identification and estimation of target effects:**

With a credible structure in hand, formalize the estimand—average treatment effect, conditional effect, mediation path, and so forth—using do-calculus or back-door/front-door criteria. Select an estimator consistent with the identification strategy (for example, doubly-robust learners, targeted maximum likelihood, or Bayesian structural models). Subject every estimate to robustness diagnostics such as placebo tests, sensitivity analysis for unobserved confounding, and refutation by synthetic interventions.

- **Step 6 – Identification and estimation of target effects:**

Translate technical findings into decision-ready insight. Provide an interactive view of the final DAG, overlay key effect sizes and confidence intervals, and supply a concise narrative linking proposed interventions to expected shifts in business or process KPIs. Document residual uncertainties and the assumptions on which identification rests, thereby satisfying both managerial stakeholders and audit requirements.

CHAPTER 4. MANUFACTURING USE CASE – TISSUE MACHINE

Chapter 3 presented a general methodology for developing causal inference analysis. In this chapter, we will explore how this methodology was implemented in a real-life manufacturing project. Firstly, the context of the project will be established, giving an overview of its purpose and objectives. Then, we will detail the steps taken to apply the methodology, providing a comprehensive description of the process. Finally, the results obtained from the application of the methodology will be presented, providing an evaluation of the project's success. Overall, this chapter serves to provide an in-depth analysis of the practical application of the methodology presented in chapter 3 and its effectiveness in achieving project objectives.

4.1 GENERAL CONTEXT:

A tissue paper manufacturer wanted to develop several projects to improve their production of roll paper. The manufacturer had a problem because there were times during the production phase that the roll paper broke, delaying production and increasing scrap. The initial project consisted of developing a predictive model to make predictions on when the roll paper was going to break 5 minutes before the break occurred. In addition to that, a **causal prescriptive model** was going to be developed to help understand the client which variables were causing the breaks to occur and help the workers achieve a better understanding of the whole system.

The roll paper machines are around 80 meters long, and the production is separated in many different phases. The first step is the creation of paper paste. This is done by processing wood pulp and refining it with different additives. Once the pulp is refined, it is sent to the tissue machine. The tissue machine oversees converting the paper paste to actual roll papers. This is the part that was analyzed during the project. The tissue machine is formed with several parts.

- **The headbox**, initial step where the pulp is formed into a paper sheet.
- **The press section**, it finishes the job of the headbox and completely press the pulp into a sheet. It squeezes out water and compresses the fibers together.
- **The dryer section** is where the paper is dried out and the rest of the water is taken out of the paper. The paper is dried up in big rotating cylinders inside the Yankee. In the Yankee dryer, the steam-heated surface of the cylinder dries the sheet as it rotates. The Yankee dryer can be several meters in diameter and is designed to produce a high-quality, uniform finish on the paper sheet.
- The next step is the **creping process**. In the creping process, the dried paper sheet is scraped off the Yankee dryer with a sharp blade or creping blade, which causes the fibers in the paper sheet to loosen and stretch. This process creates the unique soft, fluffy texture and stretchy properties that are characteristic of tissue products.

- Finally, the paper is rolled in big reels preparing for the final step which consist of cutting and forming the rolling paper.

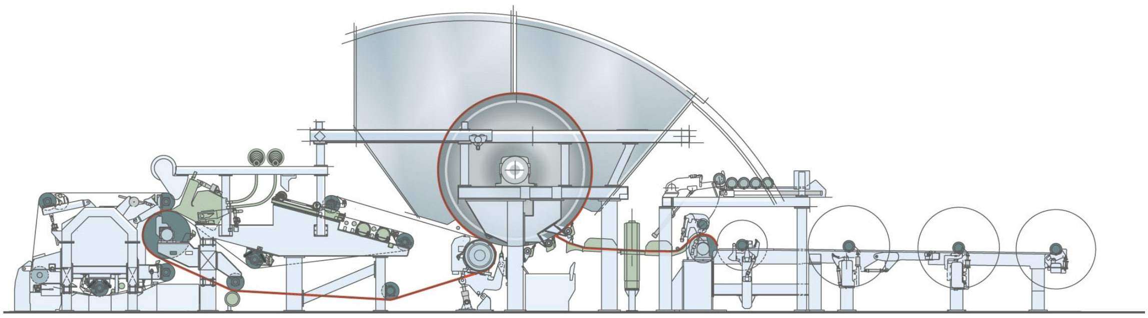


Figure 16. Tissue paper machine diagram

All these steps inside the tissue machine were sensorized and the process was parametrized from start to finish. Records of sensor data were taken each 5 seconds. From time to time, the process failed and the paper that was getting rolled in the big reels broke before arriving to an optimal length. When this occurred, the process had to be stopped and the paper rolled up to that moment had to be recycled and reprocess again, so breaks caused high costs to the client.

Machine operators were in charge of labeling when the breaks occurred. With the timestamps provided by the machine's operators and all the data retrieved from sensors, the idea was to create a predictive and prescriptive model to help the client optimize the manufacturing process. To support the prescriptive model, a causal structure map was provided to explain which variables were causing the process to break. Some of the production conditions could be modified by machine operators so the causal map would provide insights on how to tweak them to reduce the number of breaks.

4.2 DATA PREPARATION:

Since the tissue machine is a large complex machine with streaming data from instrumentation, the available data to construct the causal model consisted in more than 1100 entries recorded from different sensors. In some way, these are good news because it is ensured that most of the variables affecting the system are tracked. This helps to solve the problem of confounder bias in causal structures. But there was a problem to solve because building a causal map with that many data entrances is not possible due to the complexity of the algorithms that infer causal structures on observational data. As the thesis explained on chapter 3, the first step when building the causal structure is to get all the available data, and the second step is pruning and rearranging the data so it can be adapted for the algorithms. In this case, step 1 was easily completed since the manufacturing plant was greatly monitored. Step 2 was harder to complete since the number of variables had to be reduced from around 1000 to a list of around 100.

The approach that was followed was first to reduce redundant data by merging into one variable correlated measures of different sensors that measured the same phase of production. Also, by carrying out uni-variate analysis, data that has no variance was removed because it provided little insight. Then, with the help of the SME, other variables were discarded since they were not important and did not provide additional information.

After carefully merging variables and working on feature engineering, a set of 80 variables with one binary target variable indicating if a break occurred was provided to build the causal structure.

One of the variables, indicated the type of tissue paper that was being built during the time the sensors were collecting data. It was a categorical variable. In total, 12 different types of paper were being built, each one with significantly different characteristics and hence, different causal structures. The decision to split the data in 12 different datasets was made, one for each grade (type of tissue paper). In total, 12 different causal structures had to be built.

4.3 STRUCTURE TRAINING:

Having the 12 different datasets prepared, the next step was building the structures and iterating with the SMEs to obtain a reasonable optimal solution. With the first iteration, we realized that the causal weights pointing to the break variable were not high enough. We expected to obtain high causal weights pointing to the break variable, but that was not the case. We realized that the break variable in the dataset was not optimally tagged.

Break variable was set to 0 when the process was running correctly, and it was set to 1 five minutes before each break occurred. NoTEARS algorithm works better with continuous data, so having a binary target variable was limiting the algorithm to perform well. The problem was solved by making the break variable to increase in small steps from 0 (five minutes before break) to 1 (when the break occurred). The results obtained with this modification were significantly better.

There were several iterations over the causal structures adding and deleting causal relations. The results obtained consisted of a bundle of various DAGs, the biggest one contained the break variable. This means that there were other causal structures independent of break causing. Those structures were not considered for the project. Some of the causation weight between variables were small, so to clean the causal graph from spurious relations we decided to apply a threshold on causal weight.

Once the SMEs were in agreement with the obtained results, a custom visualization was created where the relations were shown in a clear manner.

4.4 VISUALIZATION:

It was decided that there were two graphs that could show the insights extracted from the causal maps. The first one was directly to show the resulting causal graph with the different connections and causal relations. The second one was conducting a Pareto analysis and showing the top variables that were causing the break to occur in each grade.

4.4.1 CAUSAL MAPS:

The causal map visualization consisted of making an easy to understand and clean representation of the DAGs. To do so, a color code was used to separate the variables in the different stages of production. Also, the thickness of the arrows showing the causal direction was dependent on the causal weight. The thicker the arrow, the bigger the causal weight. Finally, if a variable was causing the break (directly or indirectly), the edges and arrows coming out from that node were painted in blue. This representation was done for each of the 12 different causal graphs we obtained. Here is an example of one of them:

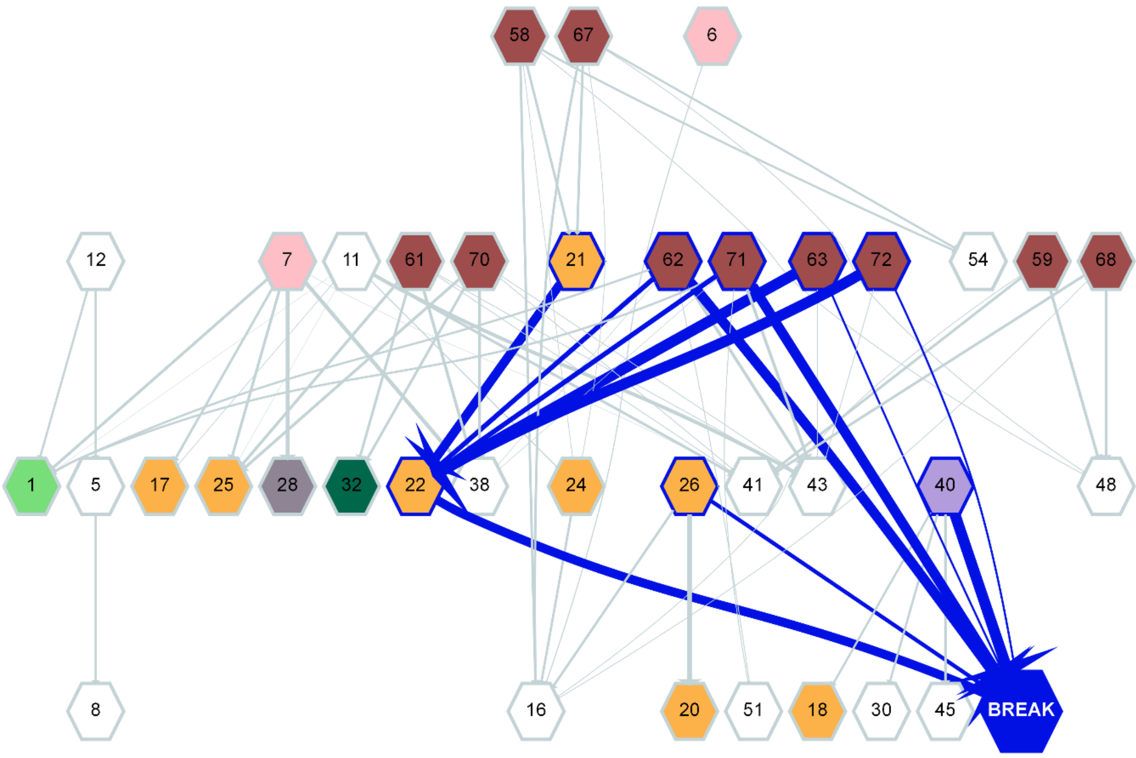


Figure 17. Causal graph showing relations of the tissue making process.

As *Figure 17* shows, there are nodes with different colors, each one representing the different stages of production. The white variables are general variables or variables that didn't belong to a specific stage. Also, it can be observed how the arrows change in size depending on the strength of the causal relation. The most important ones were pointing directly to the variable break (target variable).

Thanks to this visualization, SMEs and manufacturers could gain deep insights on how the system really worked and the variables that were affecting the misbehavior of the manufacturing line. This visualization also serves as a starting point for looking into modifying some of the flexible parameters in the production line to increase the productivity of the system and reduce the break occurrence.

4.4.2 PARETO ANALYSIS:

The pareto analysis visualization served as a summary of graph visualization. This visualization compiles the 20 variables that were causing the biggest impact in break occurrence for each type of tissue paper and for all the tissue papers in common.

It showed a bar plot of the causal weights towards the target variable, a line plot with the causal weight accumulation. Also, a color code was added to differ between variables that were common across all the different types of papers (dark blue) and the specific variables for each type (light blue).

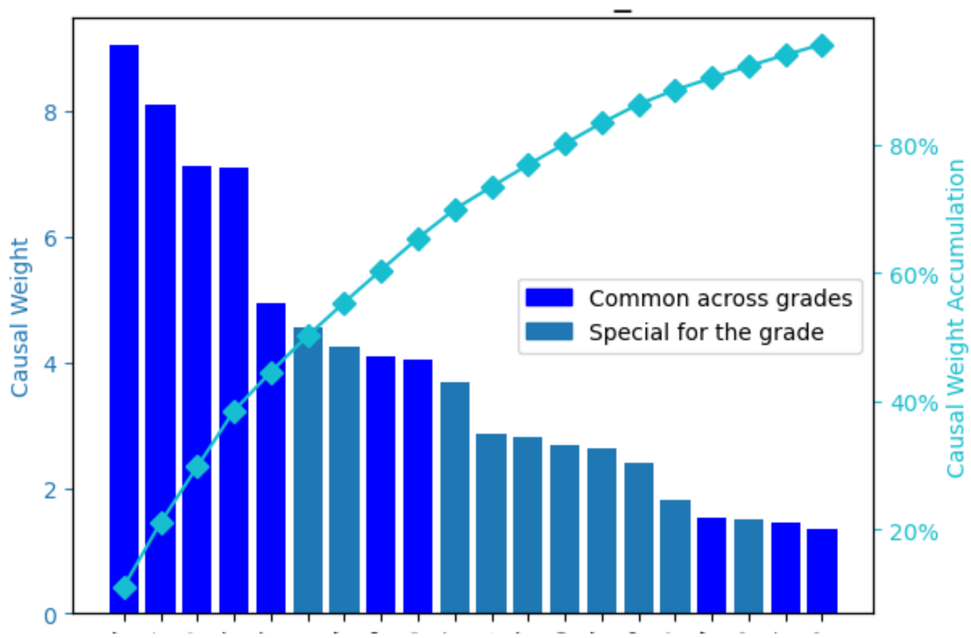


Figure 18. Pareto analysis visualization

CHAPTER 5. SOFTWARE USE CASE – INTELLIGENT DOCUMENT-AUTOMATION SAAS

Chapter 3 laid out a domain-agnostic workflow for causal inference, and Chapter 4 demonstrated its viability on a sensor-rich manufacturing line. This chapter shifts the spotlight to a purely digital setting—an intelligent document-automation SaaS—to test how well the same framework travels when the "production process" is a stream of user events rather than rolls of paper. We begin by situating the product and business questions that motivate the analysis, then trace each step of the workflow as it is adapted to click-stream data, latency logs, and CRM attributes. The chapter concludes with a quantitative assessment of the causal effects identified and a discussion of their practical implications for product strategy.

5.1 GENERAL CONTEXT:

The second empirical study transports the framework from the noisy, sensor-laden floor of a tissue machine to the asynchronous, event-driven world of a cloud application. The focal product is a Software-as-a-Service platform for intelligent document automation. Customers upload PDFs that the system (i) parses to extract structured metadata, (ii) offers automatic redaction and first-draft generation, and (iii) exposes through a conversational interface that answers questions against both internal and external knowledge bases.

Management plans to launch a new "context-aware drafting assistant" that surfaces clause suggestions in real time. Before committing engineering resources, the product team needs credible answers to three causal questions:

1. **Adoption:** Will enabling the assistant to increase the proportion of drafts generated per session?
2. **Retention:** Does initial exposure to the assistant raise seven-day active usage?
3. **Latency–Satisfaction Trade-off:** Does the additional inference time incurred by the assistant erode user satisfaction scores, and if so by how much?

Randomized rollout is feasible for a subset of users, yet heavy enterprise customers decline to be "guinea pigs," arguing that productivity stakes are too high. As a result, the experiment would cover only 30 % of traffic—insufficient for precise measurement in the enterprise tier where revenue concentrates. Observational methods, fortified by the causal workflow developed in Chapter 3, therefore become essential.

5.2 DATA PREPARATION:

The construction of a causal graph starts with a data foundation that is both complete and chronologically coherent. We first captured every event emitted during a user session—page views, API calls that launch extraction or redaction jobs, millisecond latency stamps, and granular click actions inside the drafting pane. A five-point satisfaction survey closes each session and will later serve as one of the main outcomes.

Raw behavior, however, rarely tells the full causal story. Adoption of the new drafting assistant can be tangled with who the customer is, how complex their documents are, or even when they work. To surface those latent influences, we enriched the session stream with log- and CRM-derived covariates: customer segment, billing tier, a document-complexity index, and broad time-of-day buckets. Each of these factors plausibly drives both treatment exposure (assistant on/off) and outcomes (latency, satisfaction), making their presence in the DAG essential for blocking back-door paths.

Cleaning followed three quick passes. We removed duplicate sessions created by retry loops, aligned all timestamps so that every putative cause strictly precedes its effect, and, with SME guidance, pruned deprecated or collinear metrics that add noise but no causal leverage. The resulting analytical frame—a tidy 92 variables covering 4 800 sessions over six pre-launch weeks—supplies a balanced mix of user actions, system performance metrics, and contextual covariates. It is small enough for rapid structure learning yet rich enough to capture the principal confounding channels the causal workflow must disentangle.

5.3 STRUCTURE TRAINING:

In line with the protocol from Chapter 3, we began by running NoTEARS on the curated 92-variable data set to obtain an unconstrained draft DAG. Three successive review rounds with product and infrastructure SMEs then followed:

1. **Edge validation:** Any arrow that violated basic system logic was removed. For example, survey scores recorded after a session were barred from influencing real-time latency metrics.
2. **Mandatory relations:** Arrows representing physical or business necessities were enforced—e.g., "document page count → extraction duration."
3. **Direction flips:** Ambiguous pairs (such as "manual edits" and "draft adoption") were resolved by SME judgment, shrinking the Markov equivalence class.

Each iteration re-trained the graph under the new constraint set, trimming implausible paths and speeding convergence. The final model stabilized at 87 directed edges and improved penalized log-likelihood by ≈ 14 % over the unconstrained baseline—evidence that domain knowledge both simplified the search space and raised statistical fit.

5.4 VISUALIZATION:

Figure 18 displays the final, SME-validated causal graph: 22 variables connected by 87 directed edges. Reaching this point—rather than the numerical estimates themselves—was the core objective of the exercise. The repeated loop of draft-DAG → SME review → constraint update forced the team to articulate assumptions, surface missing data sources, and converge on a shared mental model of how the service actually works. Even if that model is still incomplete, the discipline of drawing and debating each arrow already adds operational value.

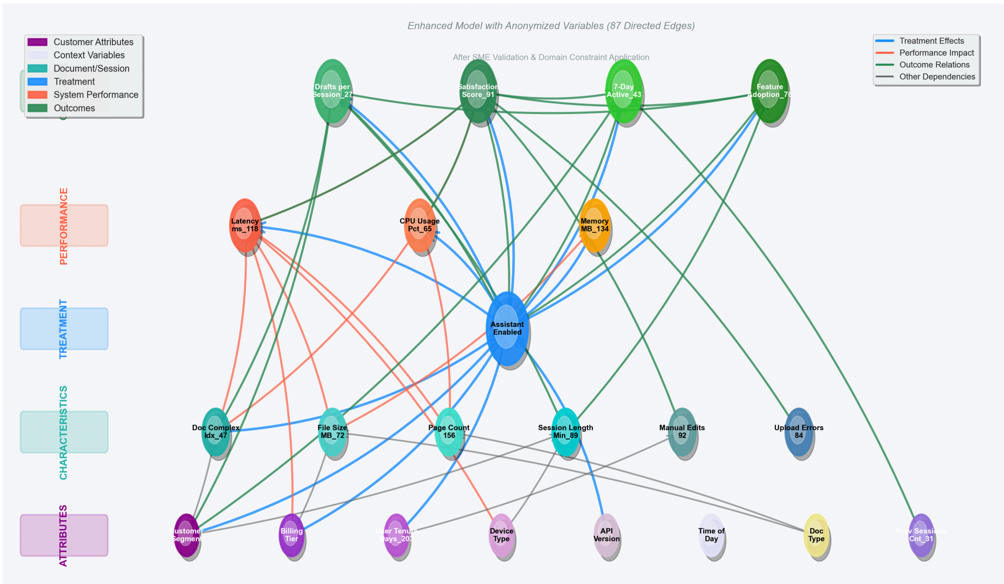


Figure 18. Saas use-case DAG

Using the graph as an identification scaffold, we estimated three effects of enabling the context-aware assistant:

- uplift in draft adoption ≈ + 8 pp (± 1.7)
- uplift in seven-day retention ≈ + 4 pp (± 1.2)
- satisfaction penalty per extra 250 ms latency ≈ – 0.05 (± 0.02)

These numbers should be read with caution. The DAG cannot rule out influences from unmeasured factors—e.g., network congestion at client sites or undisclosed document encryption methods—that could bias both latency and satisfaction. Sensitivity analyses suggest the qualitative conclusions are robust to moderate hidden confounding, yet large, coordinated shocks could still tilt the estimates.

What the structure unquestionably delivers is clarity of discussion. The graph makes it explicit that:

- Customer segment affects assistant adoption through multiple channels—document complexity, infrastructure quality, and baseline engagement—not through a single "segment" coefficient.
- Latency emerges as a central hub, mediating the path from assistant enablement to user sentiment; any optimization effort aimed at satisfaction must therefore address performance, not just interface design.
- Manual-editing time partly mediates adoption gains, indicating that product teams should track editing metrics during future rollouts.

Because these relationships are visible—not buried in weights of a black-box model—the engineering, product, and compliance teams can challenge, refine, or accept them in light of domain knowledge and new evidence. The numerical effects may evolve as more variables are instrumented, but the practice of causal-structure building—iterative, constraint-aware, SME-in-the-loop—provides a durable framework for decision-making in a high-dimensional environment.

CHAPTER 6. CONCLUSIONS

Causal inference is not the only route to interpretability—simple linear or generalized-linear models have long delivered transparent coefficients that domain experts can translate into cause-and-effect stories. However, as soon as the system under study becomes high-dimensional, nonlinear, or riddled with feedback loops, those classical tools reach their limits. In such settings—modern production lines with hundreds of sensors, cloud applications instrumented by dozens of latency and engagement metrics—causal-graph techniques offer a principled way to keep both the multivariable complexity and a clear narrative of "what drives what."

The practice remains demanding. Data coverage must be near-exhaustive, iterative SME reviews are indispensable, and graph discovery is computationally expensive. The gains, moreover, may appear incremental relative to the effort involved. Yet these methods shine when three conditions hold:

1. **Complex driver set:** many interacting variables whose relationships are opaque to purely human reasoning.
2. **Need for traceable logic:** regulated or safety-critical environments where decision audits are mandatory.
3. **Long-term lever pulling:** interventions repeated daily or embedded in automation loops, so even small effect estimates compound.

While today's algorithms still wrestle with NP-hard search spaces, the toolchain is improving rapidly—witness recent releases of NOTEARS-RL, Invariant Risk Minimization variants, and Microsoft's DoWhy. Crucially, the framework laid out in Chapter 3 does not hinge on any single algorithm. Its backbone—hypothesis → DAG → identification → estimation → validation—remains applicable as compute budgets grow and new estimators emerge.

Therefore, the thesis focuses on causal methods not because other explainable models are obsolete, but because for complex, multivariate systems they extend the reach of explainability without surrendering rigor. They are an evolving, resource-intensive toolkit—yet one well worth mastering when the stakes demand both insight and accountability.

CHAPTER 7. BIBLIOGRAPHY

[1] Correlation vs. Causation | Difference, Designs & Examples https://www.scribbr.com/methodology/correlation-vs-causation/

[2] Correlation vs Causation: Understand the Difference for Your Product https://amplitude.com/blog/causation-correlation

[3] Spurious correlations: Margarine linked to divorce? https://www.bbc.com/news/magazine-27537142

[4] What are the costs of corruption? https://blogs.worldbank.org/governance/what-are-costs-corruption

[5] Strengthening causal inference from randomized controlled trials of complex interventions. - https://gh.bmj.com/content/7/6/e008597

[6] Causal Structure. https://www.sciencedirect.com/topics/mathematics/causal-structure

[7] Causal inference & directed acyclic diagrams (DAGs). - https://bookdown.org/jbrophy115/bookdown-clinepi/causal.html

[8] Unlock the Secrets of Causal Inference with a Master Class in Directed Acyclic Graphs.

https://towardsdatascience.com/unlock-the-secrets-of-causal-inference-with-a-master-class-in-directed-acyclic-graphs-f2d3b40738e

[9] The Limits of Causal Inference. https://visionsinmethodology.org/wp-content/uploads/sites/4/2016/05/Wise_VIM_05_09_2016.pdf

[10] Causal Discovery Learning causation from data using Python. - https://towardsdatascience.com/causal-discovery-6858f9af6dcb

[11] Assessing bias: the importance of considering confounding. - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3503514/

[12] Machinery Used In Tissue Paper Making. https://www.pulpandpaper-technology.com/articles/machinery-used-in-tissue-paper-making

[13] NOTEARS and scale-invariance critique (“Unsuitability of NOTEARS for Causal Graph Discovery”).

https://arxiv.org/abs/2104.05441

[14] “Invariant Risk Minimization.”

https://arxiv.org/abs/1907.02893

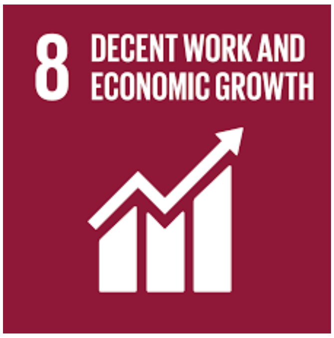
[15] Back-door Criterion Tutorial (Medium: Data 4 Sci series).

https://medium.data4sci.com/causal-inference-part-xi-backdoor-criterion-e29627a1da0e

ANNEX I: ALIGNMENT OF THE PROJECT WITH THE SDGS

This project, with the aim of increasing the efficiency and productivity of manufacturing and software can contribute significantly to the development of Sustainable Development Goals (SDGs), particularly Goals 8, 9, and 12.

- **Goal 8**: Decent Work and Economic Growth



Causal analysis provides the evidence base for interventions that raise efficiency and productivity in both physical plants and digital product teams. When a manufacturing study pinpoints which process levers truly lower scrap—or a software study shows how a new deployment workflow cuts mean-time-to-release—those verified improvements translate into cost savings that firms can reinvest in people. Expanded capacity and healthier margins give organisations room to hire, upskill, and retain workers, thereby increasing full and productive employment while advancing overall economic growth. In short, by isolating the genuine "drivers of efficiency," causal studies help convert technical gains into the broader social benefits envisioned by Goal 8.

- **Goal 9:** Industry, Innovation, and Infrastructure

Causal-inference projects that disentangle which process changes truly raise throughput or cut energy use supply the hard evidence needed to scale resilient, low-impact infrastructure. In manufacturing, a validated causal graph might reveal that moderating kiln temperature—not simply adding insulation—drives the largest drop in carbon intensity. In software operations, it could show that edge caching, rather than additional servers, delivers the biggest latency reduction per watt. By pinpointing the levers that matter, these studies steer capital toward innovations that both boost productivity and shrink the resource footprint. The result is industrial growth that is not only faster but also cleaner and sturdier—precisely the trajectory envisioned in Goal 9.

.

- **Goal 12:** Responsible Consumption and Production

Goal 12 aims to ensure sustainable consumption and production patterns. Manufacturing processes often have significant environmental impacts due to energy consumption, waste generation, and resource depletion. By increasing efficiency, the project can reduce the environmental footprint of manufacturing activities. It can optimize material usage, minimize waste generation, and implement recycling and reuse practices. By promoting sustainable manufacturing practices, the project contributes to responsible production, supports sustainable supply chains, and encourages the efficient use of resources.