Ivan Tsekov

Causal Inference in Digital Twins: Bridging the Gap Between Data and Understanding in Machine Learning with Data Intervention

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Roman Kern, Ass.Prof. Dipl.-Ing. Dr.techn.
Graz University of Technology

Graz, September 2024

Abstract

Real-world systems and processes can be represented as digital twins via data. A lack of causal understanding poses a challenge in correctly modeling systems such as digital twins. Causal inference and learning hold promise in enhancing predictive accuracy within digital twins employing machine learning approaches and their robustness. This thesis investigates the impact of causal changes on machine learning algorithms and proposes techniques for estimating causal effects by building causal model objects with the dowhy Python library. The methodology employs synthetic data and deliberate changes on this data, enabling broad applicability in an experimental approach. The experiments cover data generation, machine learning, evaluation of machine learning algorithms and estimation of causal effects. The results from the experiments are presented through plots and values. Their interpretation and comparison shed light on the importance of causal understanding and the applicability of causal learning techniques to business. Using fitted models on data with different causal relationships and distributions leads to worse performance. For instance, the mean squared error of 3.24 increases to 20.42 after an intervention on the causal relationships and the distribution of a variable. Using the same model on data that follows the same causal relationships as the train data but has different distributions leads to an error of just 4.43. The extended experiment shows that the tools for drawing interventional samples provided by the dowhy library give results close to predictions made with SVR and LinearRegression. This research contributes to the cross section between causal learning, machine learning and digital twins.

Kurzfassung

Reale Systeme und Prozesse können durch Daten als digitale Zwillinge dargestellt werden. Ein Mangel an kausalem Verständnis stellt eine Herausforderung bei der korrekten Modellierung solcher Systeme wie digitale Zwillinge dar. Kausale Inferenz und Lernen versprechen, die Vorhersagegenauigkeit von digitalen Zwillingen, die maschinelle Lernansätze verwenden, sowie deren Robustheit zu verbessern. Diese Arbeit untersucht die Auswirkungen kausaler Veränderungen auf maschinelle Lernalgorithmen und schlägt Techniken zur Schätzung kausaler Effekte vor, indem kausale Modellobjekte mit der Python-Bibliothek dowhy erstellt werden. Die Methodik verwendet synthetische Daten und absichtliche Veränderungen dieser Daten, um eine breite Anwendbarkeit im experimentellen Ansatz zu ermöglichen. Die Experimente umfassen die Datengenerierung, maschinelles Lernen, die Bewertung von maschinellen Lernalgorithmen und die Schätzung kausaler Effekte. Die Ergebnisse der Experimente werden durch Diagramme und Werte präsentiert. Deren Interpretation und Vergleich verdeutlichen die Bedeutung des kausalen Verständnisses und die Anwendbarkeit kausaler Lerntechniken auf betriebliche Fragestellungen. Die Verwendung angepasster Modelle auf Daten mit unterschiedlichen kausalen Zusammenhängen und Verteilungen führt zu einer schlechteren Leistung. Beispielsweise steigt der mittlere quadratische Fehler von 3,24 auf 20,42 nach einem Eingriff in die kausalen Zusammenhänge und die Verteilung einer Variable. Die Verwendung desselben Modells auf Daten, die den gleichen kausalen Zusammenhängen wie die Trainingsdaten folgen, aber unterschiedliche Verteilungen aufweisen, führt zu einem Fehler von nur 4,43. Das erweiterte Experiment zeigt, dass die von der dowhy Bibliothek bereitgestellten Werkzeuge zur Erstellung interventioneller Stichproben Ergebnisse liefern, die den mit SVR und LinearRegression gemachten Vorhersagen nahekommen. Diese Forschung trägt zum Schnittpunkt zwischen kausalem Lernen, maschinellem Lernen und digitalen Zwillingen bei.

Acknowledgements

I would like to express my gratitude to my thesis supervisor, Prof. Roman Kern, for their unwavering patience and guidance throughout the entirety of this research. Their expertise and mentorship have been invaluable in guiding me through the challenges of academic research.

I am also grateful to the research community in the field of Computer Science for sharing their progress which greatly helped my understanding.

To my family, I extend my thanks for their belief in my abilities and unwavering support.

Contents

1 Introduction 17

2 Literature Review 19

2.1 Causality 19

2.2 Directed Acyclic Graphs (DAGs) 23

2.2.1 Structure Discovery Methods 24

2.2.2 Combinatoric/Search based methods 25

2.2.3 Continuous optimization based methods 25

2.3 Domain knowledge 26

2.4 Digital twins 26

2.4.1 Fault detection 28

2.5 Predictive maintenance with Digital Twins 28

3 Methodology 31

3.1 Predictions 31

3.2 Literature Review 33

3.3 Implementation 33

3.4 Validity and Reliability 35

4 Approach 37

4.1 Data 37

4.1.1 Simple synthethic data for a small store 37

4.1.2 Synthetic data for a store. Adding noise, outliers and complexity to the causal model 39

4.2 Initial experiment 39

4.3 Introducing complexity to the initial experiment 45

5 Results 51

5.1 Initial experiment 51

5.2 Extended experiment with additional complexity 55

6 Discussion 63

7 Conclusion 67

Bibliography 69

List of Figures

2.1 Common DAGs: (a) chain, x causally affects y through z; (b) fork, z is a common cause for both x and y; (c) collider, z is caused by x and y and the treatments have no causal relation. 20

2.2 Digital model, shadow and twin. The Figure visualizes the difference between a digital model, shadow and a twin. The difference is characterized by the flow of data between the physical object and its digital twin. The figure is adapted from [Ful+20]. 27

2.3 Example of behaviour categorization in a Digital Twins system[SM18]. . . 30

3.1 This chart illustrates the flow of the experiment process. The cells with black text are executed for each dataset and the cells with white text are executed only once for the first dataset in each experiment. 32

4.1 Histograms showing the gamma distributions of values for demand, marketing and sales used in the first experiment. 38

4.2 Scatter plots depicting the relationship of demand and marketing with the sales. The plots indicate similar positive linear relationships. 38

4.3 Histograms showing the distributions of variables for the generated data in the second dataset for demand, external marketing, temperature, marketing, promotion and sales. The temperature data has a base of 14 to which are added daily variation, seasonal variation and noise. The peaks are the result of the daily variation. 40

4.4 Scatter plots showing the relationship between sales and the other variables. The plots reveal outliers and a positive linear relationship between sales and demand, marketing and promotion. 41

4.5 Directed acyclic graph representing the causal relationship between demand, marketing and sales. Demand plays the role of a confounder, marketing is the treatment and sales is the outcome. 42

4.6 Directed acyclic graph representing the causal relationship between demand, marketing and sales. In this case, demand is assumed to affect just the sales, where marketing is the treatment and sales is the outcome. . . . 43

4.7 Extended directed acyclic graph representing the causal relationship between demand, external marketing, temperature, marketing, promotion and sales. Demand and external marketing are confounders and can be considered environmental variables. Temperature is an instrument variable unaffected by anything else in the causal model and affects only the treatment. Marketing is the treatment and has an indirect effect on sales through the mediator promotion. 46
4.8 Extended directed acyclic graph representing the causal relationship between demand, external marketing, temperature, marketing, promotion and sales. The causal model is the same as on Figure 4.7 but without the confounding effect that demand and external marketing have on marketing. 48

5.1 Scatter plots of the test data and the predicted values for that data. The plots indicate that each model was able to capture the relationships between the variables well. Original values are blue and predicted values are red. 52
5.2 Scatter plot of the test set and the predicted values. The test set is split from the second dataset. This dataset has the causal relationships shown on Figure 4.6. The test set sales values are plotted in blue and the predicted values in red. 53
5.3 Scatter plots of the test data and the predicted values. The test set is split from the third dataset. This dataset has the same variable relationships as the first dataset and distribution of demand identical to the one in the second dataset. The test set sales values are in blue and the predicted values in red. 54
5.4 Scatter plot of sales as outcome and marketing as treatment. The plot contains the causal effect estimated for marketing via the dowhy package. 55
5.5 Scatter plots of the predicted values against the test values. The predicted values are in red and the blue values are the values from the test set. The data is not preprocessed and outliers are visible on the plots. The XGBRegressor and the RandomForestRegressor seem to be affected more by the outliers. 56
5.6 Scatter plot of the predicted values against the test values. The predictions are made after preprocessing for outliers with the Interquartile Range method. 57
5.7 Scatter plots of the predicted values against the test values. The test set is from the second synthetic dataset and the data follows the causal variable relationships from Figure 4.8. The distribution for demand is normal and the mean value is doubled. 58
5.8 Scatter plots of the predicted values against a test set split from the third dataset whose variable relationships are depicted on Figure 4.7. The distribution for demand is normal and the mean doubled compared to the first dataset. The regression algorithms are fitted with a train set from the first synthetic dataset. 59

5.9 Scatter plot of sales as an outcome and marketing as a treatment for the first dataset. The plot contains the causal effect estimated for marketing via the dowhy package. 60

List of Tables

2.1 Nomenclature by R. Guo, L. Cheng, J. Li, P.R. Hahn and H. Liu (2020) . 19

3.1 Main software tools and libraries used in the implementation. 34

4.1 Excerpt of the first dataset. The first 5 rows containing values for demand, marketing and sales. 37

4.2 Descriptive statistics for the synthetic data used in the first experiment. . 37

4.3 Excerpt of the first dataset in the extended experiment. The table contains the first 5 rows of the created DataFrame object. 39

4.4 Descriptive statistics for the synthetic data used in the second experiment. 39

5.1 The table contains the mean squared error for each regression algorithm and the mean baseline. The values are similar, indicating that each algorithm was able to capture the linear relationship. 51

5.2 The table lists sales predictions for a given constant demand and increasing marketing values. Sales are predicted with XGBRegressor, SVR, RandomForestRegressor and LinearRegression. 51

5.3 The table contains the mean squared error values calculated for the predictions on the test set from the second dataset. In this dataset the distribution for demand is changed to normal, the demand mean is doubled and the causal relationships are changed. 52

5.4 The table contains the mean squared error values calculated for the predictions on the test set from the third dataset. In this dataset the distribution for demand is normal and the mean value given in the generation function is doubled, however, the causal relationship is the one shown on Figure 4.5 52

5.5 The table contains interventional samples for sales based on the same values for demand and marketing used in the simulated situation, shown for predictions in Table 5.2. 55

5.6 The table contains the mean squared error values calculated for the predictions on the test set from the first dataset in the extended experiment. One set of errors is calculated with models fitted with the preprocessed data and another set after preprocessing. 56

5.7 The table contains sales predictions for given constant demand of 10, external marketing of 4 and temperature of 20. The values for the treatment marketing and the mediator promotion increase. 57

5.8 The table contains the mean squared error values calculated for the predictions on the test set from the second dataset in the extended experiment. The third dataset has the same causal relationship as the first and demand distribution as the second dataset. 58

5.9 The table contains the mean squared error values calculated for the predictions on the test set split from the third dataset in the extended experiment. The third dataset has causal relationships like the first dataset and normal distribution for demand as in the second dataset. 59

5.10 The table contains drawn interventional samples (via a StructuralCausalModel) for sales, given constant demand, external marketing and temperature. The values for marketing and promotion change similarly. The table shows data that can be used to make a data-driven decision. 61

1 Introduction

Digital twins are virtual translations of physical objects or systems which can use data to predict and optimize their real-world counterparts[Ful+20]. Industrial demands drive the need for better optimization of resources and systems which results in a growth of the predictive maintenance market[Liu+23]. A lack of causal knowledge makes modeling properly a complex system challenging. Causal learning and inference describe the process of understanding cause-effect relationships within a system. Causal learning discerns not only correlations but underlying mechanisms or principles driving a given phenomena. In the context of digital twins and machine learning, causal learning has the potential to improve prediction accuracy in different conditions. Causal learning requires domain knowledge. Such knowledge gives the potential for building more optimized, better controlled and more robust causal models which in turn pass those characteristics to the whole system enabling digital twins to adapt to new scenarios by understanding the causal mechanisms underlying the changes.

Researching the change of causal effects and their impact on machine learning algorithms contributes to the field of digital twins and machine learning as a whole by emphasizing the importance of causal reasoning in systems. This thesis aims to investigate the effect of causality changes on machine learning algorithms and describes techniques for estimating causal effects. This is achieved with the help of existing software tools such as Python libraries for machine learning algorithms and evaluation. Furthermore, the dowhy[1] library is used to estimate causal effects. The thesis focuses on using generalized synthetic data which makes the methodology applicable to any field. The importance of domain knowledge and correct identification of causal relationships is further mentioned.

The background information covers causality, causal graphs, domain knowledge, digital twins and the application of digital twins. Causality is the driving concept in the literature review describing different variable relationships and their meaning. Presenting such relationships in an understandable form is quite important to building causal models and Directed Acyclic graphs are another part of the background. A review of digital twins literature uncovers the history, purpose and first implementations of digital twins in manufacturing, healthcare and smart infrastructures. The review highlights the difference between a digital model, a shadow and a twin by using the flow of data. The background check of digital twins mentions applications for fault detection and predictive maintenance.

The research methodology is experimental and focuses on the implementation of existing resources. The methodology chapter analyzes the synthetic data used in the

[1] https://www.pywhy.org/dowhy/v0.9/index.html

experiments and describes the structure of the experiment. The implementation chapter describes the generation of data and how the data is used for machine learning algorithm training and evaluation. The creation of causal models and their causal graphs is described in the implementation. The result chapter contains the results of the two experiments in the form of plots and calculated values. The plots and the values of mean squared errors are interpreted and compared in the discussion chapter together with the process of estimating a causal effect. Furthermore, results from machine learning approaches and results from structural causal models are compared.

2 Literature Review

2.1 Causality

Causality describes a relationship between a cause and the effect raised by it. Causality is often approached intuitively. Discovering causality from data requires a distinction between statistical associations and causation. With limited data, a prerequisite to discover causality is a solid prior causal knowledge. A survey done by R. Guo, L. Cheng, J. Li, P.R. Hahn and H. Liu (2020) aims to review methods in discovering causality and discuss existing problems in the process.

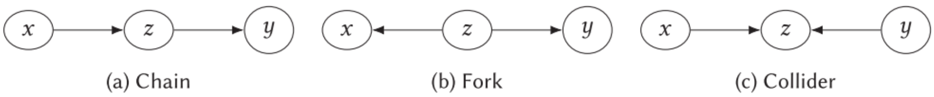
Causal inference aims to find out how much a certain variable changes if another specific variable is changed. Causal discovery is about finding variables whose values change and as a result another value of a variable is changed. The causal effects are investigated by looking into the extent with which changing a value of a cause influences an assumed effect. The manipulated variable is typically called the treatment and the responding variable is referred to as the outcome. More terminology can be seen in Table 2.1[Guo+20].

Table 2.1: Nomenclature by R. Guo, L. Cheng, J. Li, P.R. Hahn and H. Liu (2020)

Nomenclature		
Terminology	Alternatives	Explanation
causality	causal relation, causation	causal relation between variables
causal effect	-	the strength of causal relation
instance	unit, sample, example	an independent unit of the population
features	covariates, observables pre-treatment variables	variables describing instances
learning causal effects	causal discovery, causal learning, causal search	inferring causal graphs from data
causal graph	causal diagram	a graph with variables as nodes and causality as edges
confounder	confounding variable	a variable causally influences both treatment and outcome

It is mentioned that interventional data and a combination of observational and interventional data are used in learning causality. A variable's value in observational data is established by its causes. At least one variable in interventional data, on the other hand, has a value that is determined by intervention. Causal effects or causal linkages are frequently included in the ground truth that is used to train or assess causal learning algorithms. Through randomized trials, average causal effects may be ascertained with ground truth. For instance, an A/B test is frequently used to determine the average effect of a new feature in a recommendation system. Randomized experiments struggle collecting ground truths of individual causal effects, due to counterfactual knowledge prerequisite. To address this matter, domain knowledge is used to acquire ground truths through simulations.

Causal models are used to compose causal knowledge. A model describes a causal relations between variables as a mathematical abstraction. Structural causal models (SCMs) and the potential outcome framework are foundational due to their consistent representation of estimates, assumptions and causal knowledge. A structural causal model includes structural equations and a causal diagram (graph). A causal diagram is a directed graph that shows causal effects between variables. It is routine to consider only directed acyclic graphs (DAGs) in the field. In these graphs, paths do not start and end at the same node. The three most common DAGs are shown on Figure 2.1[Guo+20].



(a) Chain

(b) Fork

(c) Collider

Figure 2.1: Common DAGs: (a) chain, x causally affects y through z; (b) fork, z is a common cause for both x and y; (c) collider, z is caused by x and y and the treatments have no causal relation.

A frequent way to find causal effects in SCMs is to remove paths that represent peripheral causal effects. To identify causal effects in SCMs, back-door paths are blocked and irrelevant causal effects are removed. Back-door path, given a treatment and outcome, is a path incoming to the treatment that is not a directed path and is not blocked with a collider. Potentially, this path is coming from a confounder between the treatment and the outcome. The back-door criterion is satisfied by features if conditioning on the same features can block all back-door paths of the given treatment-outcome pair.

Even though, SCMs and potential outcomes framework have conceptual distinctions, they are similar in logic. This enables the transition or translation of assumptions from one to the other. A specific advantage of potential outcome framework is the lack of definitions for causal effects of variables which are not the special variables (e.g. instrumental variables) and the treatment. This advantage of the framework allows modeling causal effects without a complete causal graph. When a narrow estimation of a given treatment effect is required, it might be preferable to use potential outcomes. SCMs are typically chosen when the goal is to study causal relations among a group of variables, since the framework allows for learning the causal effect of any variable.

The techniques for comprehending and measuring causal effects in a data-driven way are covered further in the article. Determining how an outcome variable is anticipated to vary in response to treatment modifications is the primary objective. There are many different subgroups that may be of interest, including whole populations, particular subgroups that are defined by certain criteria, unknown subgroups, or even individual impacts.

A commonly used indicator is the Average Therapy Effect (ATE). ATE is very insightful when making decisions about implementing a therapy for a population. A range of assessment criteria are presented for evaluating ATE-learning models, including mean absolute error (MAE). It should be highlighted that in circumstances where several groups react to therapy in different ways, ATE may be deceptive. The idea of the Conditional Average Treatment Effect (CATE) is presented in order to overcome this. CATE takes into account the variations in results under various treatments for a certain combination of traits. A suggested function for the estimation of CATE is assessed for quality using mean squared error (MSE), which is also known as Precision in Estimation of Heterogeneous Effect (PEHE). PEHE is highlighted as a useful metric for analyzing estimated Individual Treatment Effects (ITE) in addition to CATE. The mean squared error of the estimated ITEs is used to describe the core of PEHE, offering a thorough grasp of its relevance in evaluating the accuracy of estimations, particularly in situations with diverse effects.

A common assumption is that all confounding variables are observed. [Guo+20] A paper by F. Eberhardt and R. Scheines (2007) describes this assumption as causal sufficiency. To reach causal sufficiency all common causes of any pair of variables are considered. It is further noted that the assumption is unrealistic but due to the huge reduction of model spaces under consideration, the effect of the assumption is substantial. [ES07] An adjustment is presented as a means to eliminate confounding bias. Confounding bias exists if the probabilistic distribution that represents the statistical association is not always equal to the interventional distribution.

The main groups of adjustments are regression adjustment, propensity score methods, and covariate balancing. Fitting a function to predict the probability distribution of the outcome given the characteristics and treatment is known as regression adjustment. There are two approaches that are covered: one where the Individual Treatment Effects (ITE) are inferred by fitting a single function to estimate and the other where distinct models are fitted for every possible outcome. When dividing cases into strata for randomized controlled trials, matching techniques are regarded as a specific example of propensity score approaches. Propensity score stratification makes the assumption that stratification is flawless, meaning that instances within a group are identical save for treatment and possible outcomes, and that groups are characterized by a set of attributes. Weighting techniques handle the absence of overlap in groups receiving a single treatment type, and supervised learning techniques are used to estimate the propensity score. Propensity score matching (PSM), propensity score stratification, inverse probability of treatment weighting (IPTW), and modification based on propensity score are the four kinds of propensity score approaches. Since propensity score stratification expands PSM and propensity score-based adjustment combines regression adjustment with propensity score approaches, the focus is on PSM and IPTW.

One technique for estimating the Average Treatment Effect (ATE) in observational research is Propensity Score Matching (PSM). In order to determine the ATE, treatment cases with comparable propensity scores are compared to control instances, and the ATE is then determined using the matched pairings. One method is the Greedy One-to-One Matching, in which the propensity score of the other treatment group's treated instance is matched with each treated instance. By comparing the results of the matched pairings, the ATE is calculated.

Propensity score stratification is an expansion of PSM in which the number of strata or specified propensity score thresholds are used to divide instances into groups. A weighted average across all strata is used to compute stratum-specific ATE, taking into account the total number of occurrences in each stratum.

A synthetic Randomized Controlled Trial (RCT) is created by weighing instances according to their propensity scores using the Inverse Probability of Treatment weighing (IPTW) covariate balancing approach. A weighted average of the factual outcomes for the treatment and control groups is computed to estimate the ATE. The weights are defined to balance the two treatment groups.

By taking into account imbalances in observed variables between treatment and control groups, these methods strive to estimate causal effects while providing alternative approaches to managing confounding in observational research. The specifics of the data and the technique assumptions determine which of PSM and IPTW to use.

Statistical dependencies are used to investigate causality, sets of causal graphs are learned by algorithms and assessed as candidates by comparing them to ground truth graphs. Equivalency classes are introduced to compare causal graphs.

Metrics used for evaluation can be put into two categories - the correctness of causal links found and the distances between learnt and ground truth graphs. For the correctness of causal links, metrics like the Frobenius norm and structural Hamming distance are utilized. For distances between learnt and ground truth graphs, metrics like false positive rate, recall, and accuracy are employed.

High-dimensional and heterogeneous data provide obstacles for learning causal relations from large data. Conventional constraint-based (CB) techniques, such the PC algorithm, have proven to be consistent and scalable for datasets including thousands of variables when applied to high-dimensional data. On the other hand, problems with the findings' variability depending on changeable order emerge. To solve this issue, Colombo and Maathuis provide a skeleton search algorithm that may be used with other CB methods. On the other hand, in high-dimensional environments, score-based (SB) algorithms have unresolved concerns regarding consistency and scalability. Another difficulty is managing mixed data, which includes both continuous and discrete variables. Independence tests appropriate for mixed data, such as conditional Gaussian tests and multinomial logistic regression, can be used to modify CB methods. According to empirical research, these exams greatly increase recollection for identifying causal relationships. Furthermore, as a preprocessing step, the joint distribution of continuous and discrete variables is modeled using mixed graphical models (MGMs). The Degenerate Gaussian (DG) score for scalable causal discovery on large-scale mixed data is constructed and shown to be consistent. On artificial datasets, DG performs better than other methods. Nonetheless, a common constraint across current methods is their suitability for variables detected in the training data, hence, creating models that may be extended to additional variables remains an unsolved issue.

By assigning causal links to the data, the learning of causal relations issue may be re-framed as a prediction problem. This allows the employement of supervised machine learning techniques. The task is to derive the causal direction label given labeled training data. One method involves using a causal regularizer to direct prediction models in the direction of discovering causal relationships between labels and characteristics. A penalty term in the objective function with the causal regularizer pushes the model to concentrate on characteristics that are more likely to be the source of the label. Using sample weights, another causal regularizer seeks to equalize the distribution of two groups with regard to each treatment. This method assists in determining the causative elements and building strong prediction models in many fields. Causal knowledge can enhance machine learning and how machine learning methodologies can contribute to causal discovery.

The process of modifying models that have been trained in one domain to function well in another, especially when labeled data is plentiful in the source domain but insufficient in the target domain, is known as domain adaptation. Invariant prediction assumes uniform conditional probability across domains. Deep Global balance Regression (DGBR) and related models use auto-encoders and causal regularizers to adapt to the domain, using low-dimensional representations to forecast the outcome and achieve global balance. Transportability is focusing on the reuse of causal knowledge across domains and is explored in the context of concept drift. Causal transportability aims to discover conditions under which causal knowledge learned from experiments can be applied to different domains. This leads to reusable causal knowledge.

In conclusion, efforts have been made to expand learning causality with big data and limited prior knowledge. These groundwork data-driven studies on causal effects and relations are foundational. The common ground between causality and machine learning brings the potential for mutually beneficial solutions in both causal and predictive problems[Guo+20].

2.2 Directed Acyclic Graphs (DAGs)

Causality is deemed important in many scientific fields. It is especially so in machine learning and artificial intelligence due to the development of effective reasoning. Gaining causal insights with randomized experimentation is proven to be costly and challenging. To address this, researchers develop and employ methods for causal discovery and inference from data. These methods function under assumptions and use the data to infer effects and structures with causal effects. An article by M. Vowels. and N.Camgoz and R. Bowden (2023) covers the theory and methods for causal induction structures. The paper focuses on combinatoric approaches[VCB22].

SCMs are a common and useful way to arrange a functioning definition of causality. Some of the causality definitions seem to be compatible in the context of SCMs. The definition given by Lewis (1973) elaborates that cause is something that makes a difference and the difference is what wouldn't have happened if the cause was not there. The Pearlian school of causal reasoning uses a Structural Equation/Causal Model

(SEM/SCM) indicating assignment of treatment value to a function of the structural parents and external noise. Furthermore, intervention is used to set the value of the treatment to a certain quantity. This structure, including the parents, can be shown with DAGs. Keeping in mind the assumptions, scientific questions can be answered with observational data.

Graphs consist of undirected paths, directed paths, parents, ancestors, immorality or v-structure, and colliders. Directed cycles are relevant in modeling cyclic properties and feedback in natural phenomena.

The recursive decomposition of the joint distribution is specified by the Markov Assumption. The distribution is a common attribute of Bayesian networks. Separation indicates conditional independents between vertices. Faithfulness ensures that conditional independents in the graph are reflected in the distribution. This assumption is critical in keeping the consistency between the graph structure and the real distribution.

The Markov Equivalence Class (MEC) concept finds a use in scenarios where multiple graphs fulfill the same conditional independents. The graphs have the same skeleton. Completed Partially Directed Acyclic Graphs (CPDAGs) represent MECs, with directed edges only if their direction is consistent across all graphs in the class. In other cases, the edges are not directed.

A crucial assumption states that all relevant data has already been observed. An unobserved confounding leads to Acyclic Directed Mixed Graphs (ADMGs) and Maximal Ancestral Graphs (MAGs). Hidden confounding and bidirected edges means ADMGs. Absence of directed paths between spouses and absence of spouses or parents for neighbors suggests acyclicity - MAGs. As discussed in the previous article [Guo+20], to ensure consistency between implied conditional independencies and the real distribution in MAGs, an assumption of faithfulness has to be established[VCB22].

2.2.1 Structure Discovery Methods

The article reviews several structure discovery methodologies. Constraint-based methods identify causal relationships by testing for conditional independencies in the empirical joint distribution. It is common for the output to contain Markov Equivalence Classes (MECs). The classes represent graphs with the same conditional independencies. Conditional independence require large sample sizes to be reliable[VCB22].

The second type of methods is Score-based. These methods allow for prior knowledge inclusion and a different approach to causal structure discovery. These methods validate graphs with the help of score functions with the aim to discover the graph with the maximum score. Common scores are Bayesian Information Criterion (BIC) and Minimum Description Length.

The third type of methods make specific assumptions about the functional or parametric forms of underlying data-generating structures to identify causal directionality. The approach is called Exploiting Structural Asymmetries. These methods test edges one at a time or triples, an unobserved confounder is involved. An example can be an additive noise model that looks for asymmetries in regression residuals and uses the results to infer causal directionality. Exploiting Structural Asymmetries methods leverage specific structural properties to determine direction of causal relationships.

Another type of methods are Intervention-based methods. These methods cover variable manipulation through interventions to search for causal relationships. In cases with limited or hard to collect observational data, these methods show great value. Intervention-based methods can be used on whole graph candidates or only edges. In this context, they provide flexibility in design.

Furthermore, causality can vary with time varying variables. Granger causality test can be used in such scenarios. In dynamic systems, Dynamic-causality methods are an alternative to Granger causality to overcome some of its limitations. For instance, Granger causality is not dependable when it comes to interdependency between two or more variables.

The causal inference algorithms, mentioned so far, can be assessed with metrics - True Positive Rate (TPR), False Positive Rate (FPR), Area Over Curve (AOC), Structural Hamming Distance (SHD), and Structural Interventional Distance (SID)[VCB22].

2.2.2 Combinatoric/Search based methods

The difficulty of finding DAGs raises exponentially with the number of variables. An approach to this problem is continuous optimization methods for graph learning. These methods use mostly combinatoric or search-based strategies to discover the causal structure. The assumptions made by such methods include sufficiency, faithfulness and acyclicity. Sufficiency is the assumption of potential existence of hidden variables. Faithfulness indicates the level of achieved form of faithfulness, be it relaxed or severe. Finally, there is acyclicity, which indicates the ability of a method to discover feedback loops and cycles[VCB22].

2.2.3 Continuous optimization based methods

Continuous optimization-based methods for structure discovery are covered in the paper. The combinatoric graph-search issue is transformed by the continuous optimization method into a continuous problem - Equality Constrained Program. The incorporation of continuous optimization into causal structure learning is beneficial in AI. The benefits of disentangled and structured symbolic representation enhance the performance across domains and the interpretability. An example is the CMS algorithm, which identifies directions in causal relations when it comes to dynamic systems. This is achieved with neural networks and shadow embeddings. Another example is NO TEARS (Non-combinatoric Optimization via Trace Exponential Augmented lagRangian Structure learning). NO TEARS is believed to be the first algorithm to handle the combinatoric graph-search as continuous optimization. NO TEARS uses an acyclicity enforcement function.

These methodologies range from continuous optimization-based techniques, neural network models, adversarial training, meta-learning, and reinforcement learning strategies. The different techniques address causal structure learning challenges such as super-exponential growth of DAGs due to many variables, the hard nature of the discovery problem and handling of hidden variables. A commonality is that methods enforce acyclicity with the purpose of ensuring causal relationships and to navigate concessions between optimal results and computational cost. Furthermore, the different approaches accommodate diverse data types, be it time series or data of high dimensions. Continuous optimization based methods are striving to provide interpretability and disentanglement of causal factors.

In conclusion, causal discovery methods lead to unequivocal insights into the nature of reality. These methods depend on assumptions like faithfulness and acyclicity, which may be untestable in some cases leading to uncertainty. Sensitivity to model structure, unobserved confounding, and the oversimplification of concepts like gender or race may inaccurately represent causal relationships. Biases in data may lead to skewed results. Understanding the limits of a given knowledge and the meanings of variables add extra layers of complexity to the methods. Despite these challenges, the spectrum of methods shows a joint effort in tackling them and offering valuable insights in the nature of causal learning[VCB22].

2.3 Domain knowledge

An arcticle by J. Lin and Y. Zhao and W. Huang and C. Liu and H. Pu (2020) researches the knowledge representation and its development in regard to domain knowledge graphs. Knowledge representation is labeled as an important step to build domain knowledge graphs. Representation of knowledge can be a digital model representation of the natural world or an object. Such models have limitations and are considered data structures. The properties of these models have values that describe the characteristics of an entity. Values can describe relationships used to connect two or more entities. Entities are units of knowledge graphs and have an independent existence. Knowledge covers facts, rules and principles acquired by observing and thinking about various situations in the world[Lin+21].

Another article by J. Grundspenkis (1998) discusses knowledge acquisition in regard to deriving causes. In general, rules of the structure IF-THEN are formed. However, this is considered ”shallow knowledge” because of the limitations. A serious limitation is the absence of an explanation what is happening in a system when faults occur. In this time, a shift can be observed from rule-based to model-based systems. Model-based systems introduced reasoning and explanation of behaviors to bring more understanding. This level of understanding requires the inclusion of causal knowledge to the domain knowledge[Gru98].

2.4 Digital twins

An article by A. Fuller and Z. Fan and C. Day and C. Barlow (2020) labels Digital Twins as a core element in the Industry 4.0 revolution. Digital Twins find usages in manufacturing, healthcare and smart infrastructures by integrating Internet of Things (IoT) and data analytics. By showing the state of physical objects or processes virtually, Digital Twins create an interconnected environment[Ful+20].

Digital Twins concepts started coming along in the early 2000s[EBA20]. The first of the definitions for Digital Twins was introduced in 2003, in a presentation. The National Aeronautical Space Administration (NASA) introduced another Digital Twins paper in 2012. The descriptions over the years focus on the application of the concept. For instance, NASA (2012) defined digital twins as simulation of vehicle or system that uses fleet history or sensors to mirror a physical object. Another definition from 2017 states that a digital twin is a computerized model of a physical object represented by its features and links with other working elements. Other definitions use terms like a living model of a physical objects that adapts continuously, representation of a physical item utilizing simulations and service data, and a virtual instance of a physical entity continuously update by its physical twin[Ful+20].

The distinction between Digital Twins and general computing models can be complicated. Three concepts are outlined - Digital Model, Digital Shadow and Digital Twin[EBA20]. A Digital Model covers a one-way relationship. This relationship is defined by the lack of automatic data exchange with the physical model, due to this the digital model is not impacted by the physical one[Ful+20].

Digital Shadows take an additional further step from Digital Models by implementing a flow of data from the physical model to the digital object. In this scenario, the physical twin makes alterations to the digital twin[Ful+20][EBA20].

Digital Twins have a bidirectional integration. This means that the digital twin can influence the physical twin, and vice versa. This is labeled as the true concept of Digital Twins[Ful+20].

Figure 2.2 visualizes the difference between a digital model, shadow and a twin. The main factor in the difference is the direction and existence of an automatic data flow.

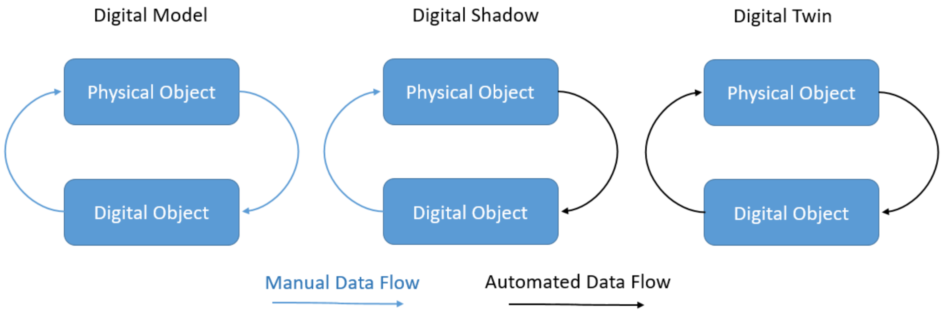


Figure 2.2: Digital model, shadow and twin. The Figure visualizes the difference between a digital model, shadow and a twin. The difference is characterized by the flow of data between the physical object and its digital twin. The figure is adapted from [Ful+20].

Some of the domains that use digital twins are smart cities, manufacturing, automotive industry, construction industry and healthcare. In smart cities, digital twins help with planning, development and energy efficiency. Furthermore, they are used as test environments for AI algorithms and the data provided by the sensors aids in analytics and monitoring. In manufacturing, digital twins introduce Industry 4.0 principles - insights into machine performance, production feedback, predictive issue detection. In the Automotive Industry, digital twins are found useful for simulation, data analytics and testing. Digital Twins show great versatility in their applications.

Implementing a Digital Twins system comes with several challenges. To start with, a prerequisite to AI is often a high-performance infrastructure. Such infrastructures, both software and hardware, come with high costs and security concerns. The quality of the collected data is not a given and it is crucial to ensure data quality with the right cleaning and processing procedures into the AI algorithms. To continue with, privacy and trust are some of the other challenges. Measures to protect user data have to be taken into consideration. The challenge of trust requires the analysis of potential negative impacts of AI. Furthermore, unrealistic expectations have to be considered. AI cannot solve all problems instantly and the use of AI has to be examined properly.

A commonality in the research of Digital Twins for the different field applications is the lack of unified models or architectures on how to construct a Digital Twin system. The advancement of AI and machine learning enrich Industry 4.0 concepts. Such concepts advance Digital Twin technology, particularly in predictive maintenance and assessment[Ful+20].

2.4.1 Fault detection

In a review of digital twins concepts (2021), it is indicated that one of the goals that digital twin models have is to reduce inconsistencies between the expected and actual behavior. It is stated that digital twins can identify and eliminate unforeseen issues, whereas the existing methodologies focus on predicted problems and the validation of requirements.

Digital twins fault detection and diagnosis can combine data-based fault diagnosis and physics-based fault diagnosis. This seems to be an effective solution for anomalous problems in such systems. Some researchers leverage transfer of deep learning, dynamic Bayesian network and nonlinear dynamics to model the behavior of physical entities.

In another approach, digital twins interpret collected data from a different perspective by comparing the simulated data to the collected data to determine the failure mode[Liu+21].

2.5 Predictive maintenance with Digital Twins

A publication by Raymon, Bedir and Cagatay (2022) reviews objectives, domains and solutions in the context of predictive maintenance with Digital Twins. Predictive maintenance is said to be crucial for Industry 4.0.

The review covers maintenance methods with varying complexities. The methods can be reactive, preventive, condition-based, predictive or prescriptive. The highlighted purpose of predictive maintenance is to reduce costs and increase machine up-times. The recent maturity of the topic Digital Twins indicates a possibility for a causally connected and synchronized physical and digital objects.

The aims of predictive maintenance using Digital Twins are concluded to be estimation, prediction and detection of the condition of systems. Outputs of the maintenance could be classification task results like categories or machine states, and regression task results such as time until a machine fails. The main application domains are stated to be Manufacturing and Energy. The use of Digital Twins for maintenance is made possible by platforms like OpenModelica, MathWorks Simulink and SAP Leonardo. The prediction involves various models such as geometric models for shapes and positions, physical models for property and load simulation and decision-making models for evaluation and reasoning. A combination of two or more models is possible[vTC22].

There are machine learning and deep learning approaches recorded for predictive maintenance using Digital Twins. A machine learning approach example is the usage of Support Vector Machines (SVM) for binary and multi-class classification tasks, and Decision Trees and Random Forest for remaining useful life (RUL) predictions. RUL predictions are done with deep learning approaches like Autoencoder. This approach is used further for anomaly detection. Another deep learning approach is Long Short-Term Memory (LSTM) applied for RUL prediction and feature extraction. Common evaluation metrics include accuracy, Root Mean Squared Error (RMSE), precision, recall and F1-measure. The versatility of methods and evaluation metrics showcases the use of Digital Twins across the different domains.

Another approach that expands the concept of Digital Twins is discussed by N. Stojanovic and D. Milenovic (2018)[SM18]. The purpose of the approach is continuous improvement and quality control enhancements. The digital twins are described as self-aware. The explanation of the system begins with a description of the system behavior theory shown on Figure 2.3. Predicted desirable (PD) is the desired behaviour of the system/process and the system tries to keep this state. Predicted Undesirable (PU) is expected but undesired behavior, preventive maintenance aims to predict and stop the occurrence of this behavior. Unpredicted Desirable (UD) is behavior that is not planned but still proper behavior. This behavior indicates instability. The system aims to understand, predict and include such behaviors in the model. The last behavior is Unpredicted Undesirable (UU), covering anomalies which the system does not recognise. The system's goal is to avoid these anomalies before they can occur[SM18].

The understanding of the Digital Twin system behavior is used to derive the functionality of Root-Cause-Analysis (RCA). RCA aims to know the process/system parameters that cause problems. The summarized duties of the self-aware digital twin can be continuous monitoring, behavior understanding, reaction to detected behavior with the purpose of continuous improvement, avoiding undesired behavior and detecting improvement opportunities.

The paper written by N. Stojanovic and D. Milenovic (2018) provides technical specifics. A web portal is used when the system malfunctions to try and find the cause. A component called storage worker, keeps database specifics hidden behind an API and is responsible for database connections and sessions. For instance, storing and retrieving data instances. The database used is a scalable non-relational database for both parsed

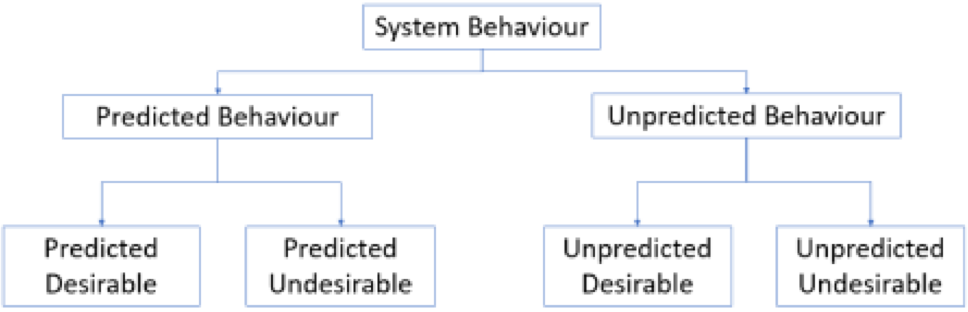


Figure 2.3: Example of behaviour categorization in a Digital Twins system[SM18].

input data and system generated data. The data is singular or multiple time series data. A complex procedure involves the storage of data in a single row. A clustering model groups data based on similarity and generates database entities with clustering properties like distances in clusters, means and standard deviations. The procedure is executed periodically and results in a new model and cluster sets each time. An anomaly detector receives suspicious entities and labels them if they are anomalies. The system uses timers to trigger components.

The authors mention the importance of partitioning the time space into windows for proper data analysis. In their analysis, 5-minute windows are deemed appropriate. The clustering algorithm used is K-Means. Silhouettes coefficients are used to confirm cluster numbers. The authors report an average score of 0.88 for 2 clusters and 0.72 for 3 clusters. As part of the root cause analysis, the clusters are shown via a scatter plot. The clusters are classified as behaviors according to Figure 2.3. The focus is on investigating parameters which deviate the most[SM18].

3 Methodology

Experiments play a key role in machine learning research[BV07]. The approach to this research consists of two experiments with the second experiment building upon the first experiment. The experiments begin with the creation of a causal model represented with a graph. The causal relationships are used to generate a dataset. The dataset created in the extended experiment follows a more complex causal model and the generation process includes adding noise and outliers. This introduces the need for outlier preprocessing in the second experiment. The creation of the data is followed by generating descriptive statistics, histograms and scatter plots for the datasets. The experiments continue with splitting their first dataset into train and test sets. The train set is used for fitting regression algorithms. The algorithms are used to make predictions on the test set. The predictions are used to calculate mean squared errors and to generate scatter plots with the predicted values and the original values. The models are used to predict in a simulated situation requiring data-driven decision-making. These predictions are to be compared to prediction results from a dowhy causal model at the end of the experiments. The experiments continue with alterations to the causal relationships and variable distribution which is followed with the creation of two new dataset. From each new dataset a test set is split and the previously fitted models are used to predict. The predictions generate new mean squared error results and scatter plots which are used to compare the performance of the algorithms on the different datasets and noting the differences between the datasets. The experiments conclude with the estimation of the causal effect of the treatment, refuting the robustness of the causal effect, calculating arrow strengths for the causal graphs and drawing interventional samples. The process is visualized on Figure 3.1.

3.1 Predictions

The regression algorithms in both experiments predict sales. The data generation aims to create experimental store datasets incorporating various factors such as demand patterns and marketing records. Other variables such as external marketing, temperature and promotion are added to datasets in the extended experiment. Predicting sales for given marketing campaigns helps understand the impact of such promotional strategies and possibly save on marketing in situation where marketing might not be helpful to increase the sales of a store.

The variables in the dataset are considered features where sales is considered the target. This is the case for each of the generated datasets. The predictions are made based on the features.

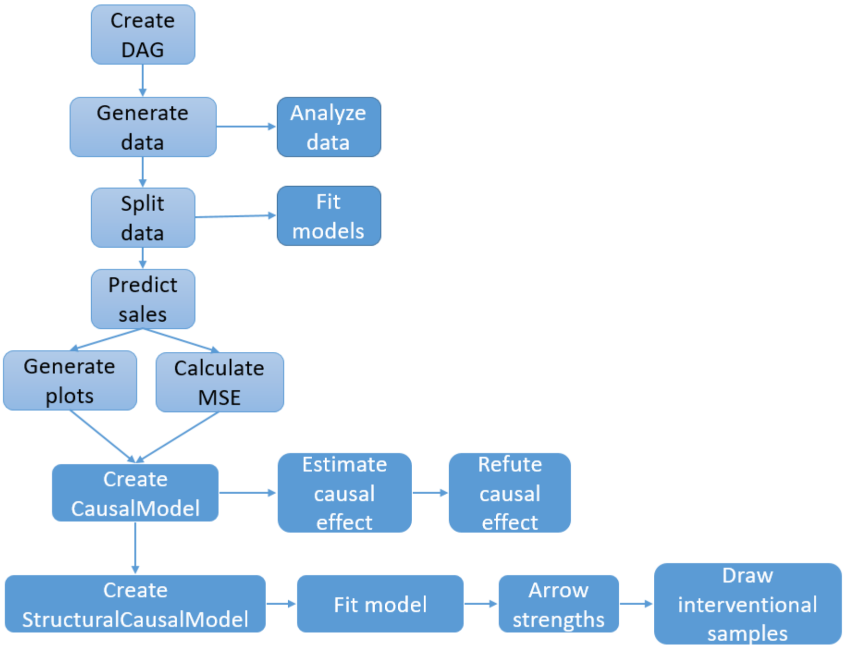


Figure 3.1: This chart illustrates the flow of the experiment process. The cells with black text are executed for each dataset and the cells with white text are executed only once for the first dataset in each experiment.

The motivation behind the variable naming is to point out potential situation where the research can be of value. For instance, predicting sales in the context of digital twins can be used to optimize the management of a store, integrating sales predictions with inventory management.

3.2 Literature Review

The research design of this thesis is experimental. The strategy includes literature reviews of causality, directed acyclic graphs, causal models, domain knowledge and digital twins. The literature then guides the formulation of the experiment and as a final step the experiment results are analysed.

The starting point of the literature survey is the concept of causality and causal discovery. The review begins with papers that investigate or survey how to learn causality with data. The search engine used is Google Scholar. Search terms are **causality**, **learning causality**, **causal inference** and **causal discovery**. Papers with more citations are preferred. When a paper is found to relate well with the purpose of this research, papers that have cited the work are further browsed for more related literature material. The literature review continues with topics related to domain knowledge and digital twins with search terms **domain knowledge** and **digital twins**.

3.3 Implementation

The implementation of the two experiments is done in Python. Python provides extensive support for data science through libraries like pandas, numpy, sklearn and matplotlib[Van16]. Furthermore, there are libraries that help with causal inference like dowhy[SK20]. Experiments can be done in a jupyter notebook where code can be segmented into cells and each cell can have an output. These capabilities of a jupyter notebook allow for live code execution, data analysis and visualization, and easier documentation and presentation[Pim+21].

The implementation involves training regression algorithms such as XGBoostRegressor, SVR, RandomForestRegressor and LinearRegression. The predictions of the fitted models are evaluated via scatter plots which show how close the predicted values are to the actual values and the popular mean squared error metric creating a base for comparison[Ple+22][Tat21]. The dowhy library is used to investigate the causal effect on the outcome - sales.

Coding practices like descriptive variable names and documentation comments are employed[SSH17][SHJ13]. The experiments follow the same clear and organized structure of data generation, data analysis and visualization, training a regression algorithm, simulating situational data for decision making and analysis of causal effect of a the chosen treatment with dowhy.

```
# data generation with numpy
demand = np.random.gamma(10, scale=1, size=10000)
```

Table 3.1: Main software tools and libraries used in the implementation.

Name	Description	Version
pandas	Easy to use data analysis and manipulation tool	1.5.0
numpy	Provides a multidimensional array object and an assortment of mathematical, logical, shape and other routines	1.23.3
dowhy	Causal inference tool that supports modeling and causal assumption testing	0.11.1
networkx	Graph creation and manipulation tool	3.0
matplotlib	Visualization tool for plots and figures	3.6.0
seaborn	Based on matplotlib and used for the visualization of statistical graphics	0.12.1
sklearn	Provides tools for predictive data analysis	1.1.3
xgboost	Provides machine learning algorithms under the Gradient Boosting framework	2.0.3

```
marketing = demand + np.random.gamma(8, scale=1, size=10000)
sales = demand + 0.6 * marketing + np.random.gamma(3, scale=1, size=10000)
data = pd.DataFrame(dict(demand=np.round(demand),
marketing=np.round(marketing), sales=np.round(sales)))
data.head() # data excerpt
data.describe() # descriptive data stastics

# visualizations
sns.histplot(data['demand']) # histograms for each column
plt.scatter(data['demand'], data['sales']) # scatter plots

# training
X = data[['demand', 'marketing']] # features
y = data['sales'] # target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=SEED) # splitting data into train and test set
regressor = xgb.XGBRegressor(random_state=SEED, verbosity=1)
regressor.fit(X_train, y_train) # training a XGBRegressor

# evaluation of the regressor via MSE and plotting predicted vs original
y_pred = regressor.predict(X_test)
mean_squared_error(y_test, y_pred)
fig, ax = plt.subplots(1, 1)
ax.scatter(X_test['marketing'], y_test, label='test data')
ax.set_title('Test set and predicted')
ax.scatter(X_test['marketing'], y_pred, color='r', label='predicted')
ax.legend()
```

```
plt.xlabel('marketing')
plt.ylabel('sales')

# situational predictions
situational_data = pd.DataFrame(dict(demand=[5, 5, 5], marketing=[5, 10, 15]))

# analysis of the causal effect
model = dw.CausalModel(data, treatment='marketing', outcome='sales',
common_causes='demand') # initialization of a causal model
# finding an estimand type (backdoor, frontdoor or instrument variable)
id_estimand = model.identify_effect(proceed_when_unidentifiable=True)
# estimating effect with a backdoor estimand method
target_estimand = model.estimate_effect(id_estimand,
method_name='backdoor.linear_regression')
# plotting a line for the effect together with a scatter plot for marketing and sales
dw.plotter.plot_causal_effect(target_estimand, data['marketing'],
data['sales'])
```

3.4 Validity and Reliability

The internal validity of the research methods is ensured through the design and implementation of the experiments. The use of Python and the libraries mentioned above provides robustness to data science tasks. The causal effect estimates of dowhy can be verified via the data generation formulas. The synthetic dataset cannot compare to insights gained from real data, however, the methodologies employed result in insights which can be generalized.

The research methods are stable and can be reproduced. This is ensured through coding practices and control over the random state of the data generation process and the algorithm training. This is all clearly segmented in jupyter notebooks.

4 Approach

4.1 Data

4.1.1 Simple synthethic data for a small store

The data used in the experiments is synthetic, generated with the help of the numpy Python package. A seed is used to make the results reproducible. In the first experiment, synthetic data is used to emulate the demand, marketing campaign and sales of a small store. The data is used to create pandas DataFrame object. The head,the first 5 entries of the DataFrame object, are shown in Table 4.1. DataFrames have an interface that makes preprocessing and visualizing data easier. For instance, in Table 4.2 the result of the describe method is shown. This method generates descriptive statistics for the numeric data. The table is used to make sure that there are no negative values and to get a basic understanding of the generated data.

Table 4.1: Excerpt of the first dataset. The first 5 rows containing values for demand, marketing and sales.

demand	**marketing**	**sales**
8.0	17.0	19.0
12.0	26.0	30.0
10.0	19.0	22.0
11.0	17.0	24.0
15.0	21.0	31.0

Table 4.2: Descriptive statistics for the synthetic data used in the first experiment.

	demand	**marketing**	**sales**
count	10000.000000	10000.000000	10000.000000
mean	10.094700	18.078600	23.939300
std	3.206674	4.268354	5.676005
min	2.000000	6.000000	9.000000
25%	8.000000	15.000000	20.000000
50%	10.000000	18.000000	23.000000
75%	12.000000	21.000000	27.000000
max	31.000000	38.000000	53.000000

Plots are used to analyze the data. The histograms on Figure 4.1 show the distribution of the generated data and the scatter plots on Figure 4.2 are used to visualize the relationships between the variables and sales.

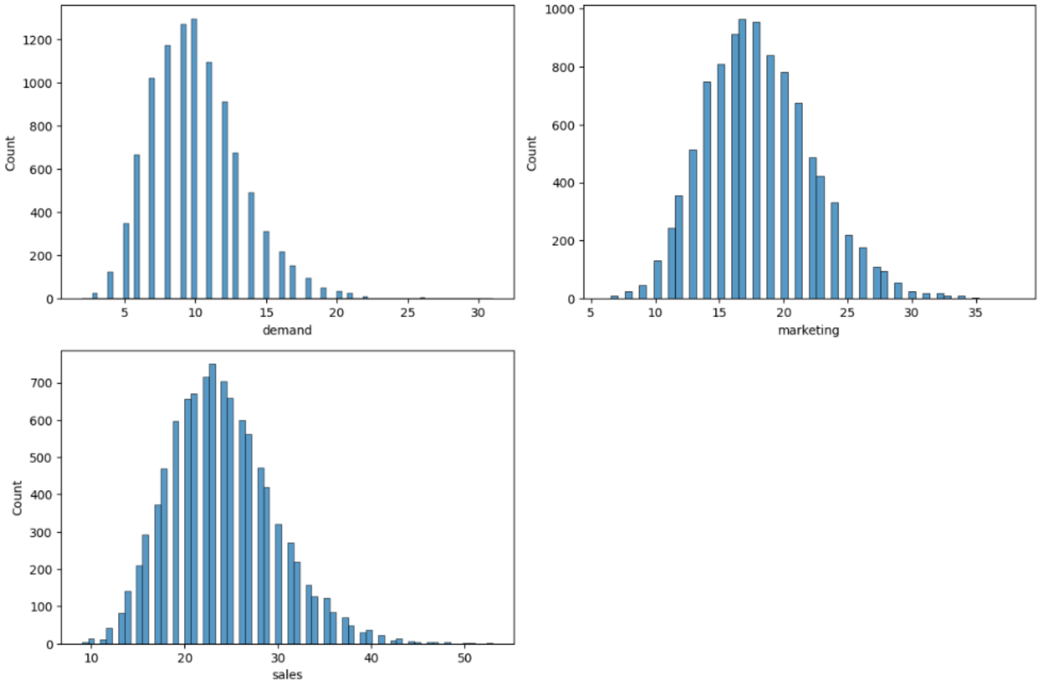


Figure 4.1: Histograms showing the gamma distributions of values for demand, marketing and sales used in the first experiment.

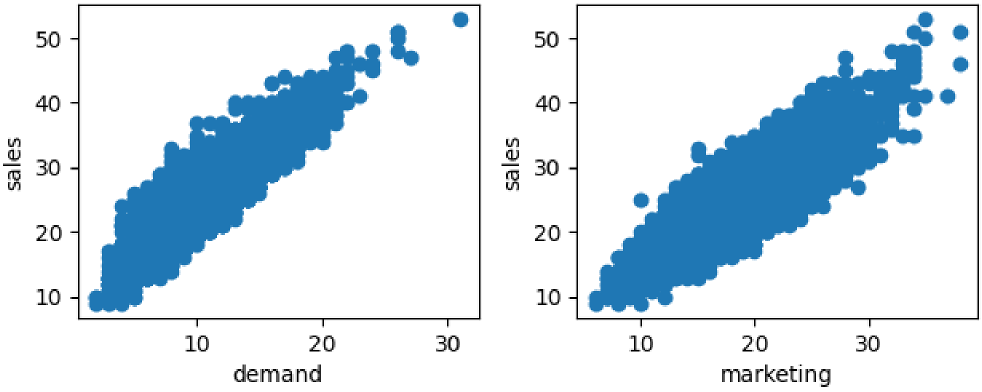


Figure 4.2: Scatter plots depicting the relationship of demand and marketing with the sales. The plots indicate similar positive linear relationships.

4.1.2 Synthetic data for a store. Adding noise, outliers and complexity to the causal model

Similar to the first store dataset, this one is generated with the help of numpy and a seed is used to ensure that the results can be reproduced. This set contains values for demand, external marketing campaigns, marketing campaigns, temperature, promotion and sales.

Unlike the first dataset, the second introduces more complexity by adding noise and outliers to the data. Furthermore, the relations introduced are more varied. This is done for the purpose of introducing different causal relationships between the variables, which are investigate further in the Implementation chapter.

An excerpt of the data is shown in Table 4.3 and the descriptive statistics of the dataset are shown in Table 4.4.

Table 4.3: Excerpt of the first dataset in the extended experiment. The table contains the first 5 rows of the created DataFrame object.

demand	**ext_marketing**	**temperature**	**marketing**	**promotion**	**sales**
10.0	8.0	12.0	26.0	19.0	41.0
16.0	6.0	17.0	21.0	16.0	41.0
10.0	5.0	20.0	19.0	13.0	35.0
13.0	6.0	22.0	23.0	18.0	40.0
17.0	5.0	20.0	29.0	30.0	58.0

Table 4.4: Descriptive statistics for the synthetic data used in the second experiment.

	demand	**ext_marketing**	**temperature**	**marketing**	**promotion**	**sales**
count	10000	10000	10000	10000	10000	10000
mean	12.15	7.11	14.05	22.66	19.12	42.94
std	3.75	2.08	8.33	4.73	4.08	18.62
min	3	1	-7	9	3	16
25%	10	6	8	20	16	37
50%	12	7	14	22	19	42
75%	14	9	21	25	22	47
max	78	15	36	92	49	599

The distribution of the values for the variables is shown on Figure 4.3. The scatter plots on Figure 4.4 visualize the generated relationships between the variables and sales.

4.2 Initial experiment

The implementation of the methodology includes two experiments with synthetic data which follow the same structure. In the first experiment a simple causal model is developed and represented as a directed acyclic graph, shown on Figure 4.5. The figure

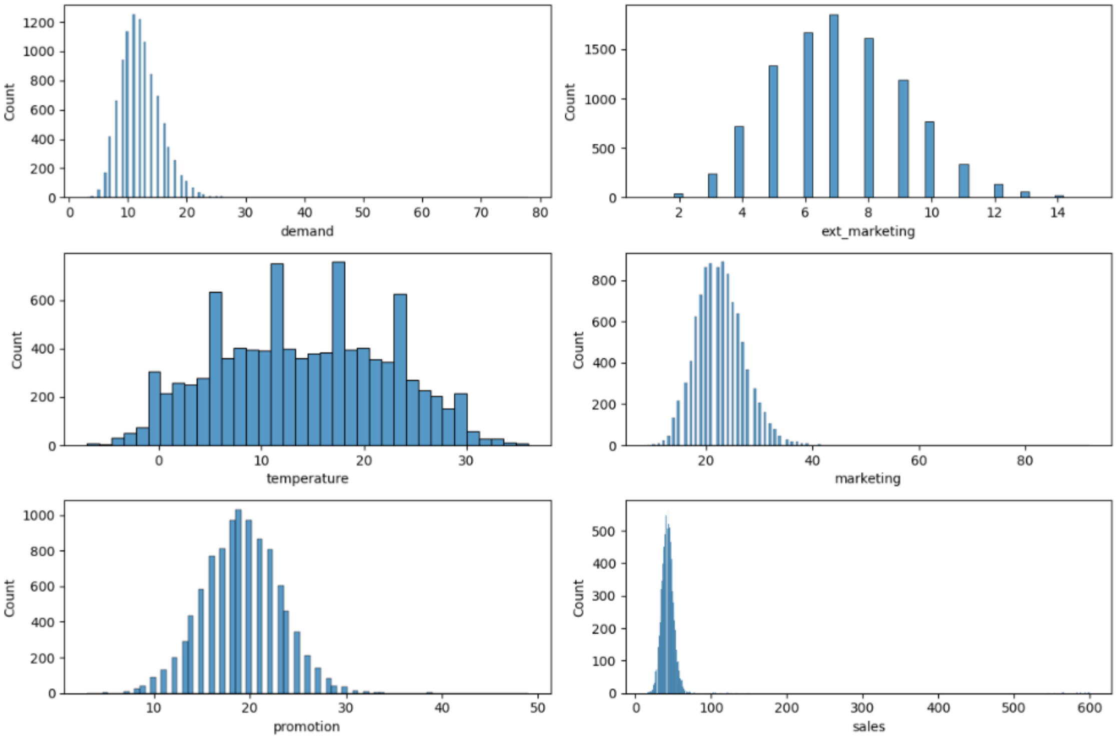


Figure 4.3: Histograms showing the distributions of variables for the generated data in the second dataset for demand, external marketing, temperature, marketing, promotion and sales. The temperature data has a base of 14 to which are added daily variation, seasonal variation and noise. The peaks are the result of the daily variation.

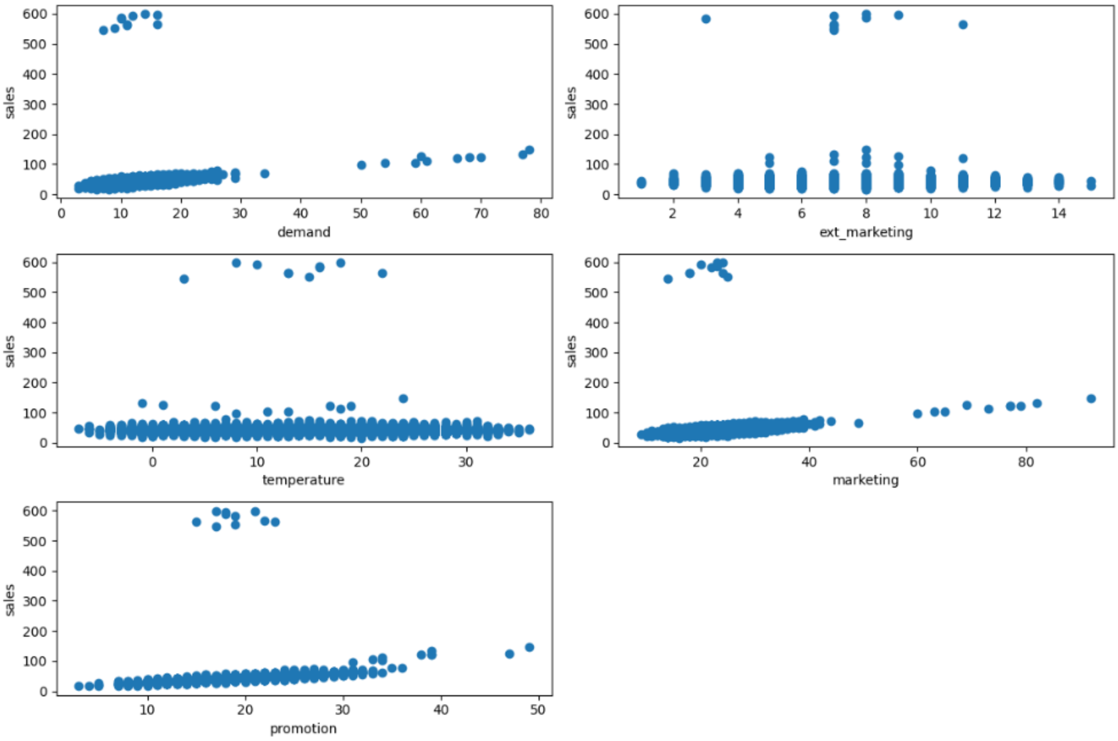


Figure 4.4: Scatter plots showing the relationship between sales and the other variables. The plots reveal outliers and a positive linear relationship between sales and demand, marketing and promotion.

represents the simulated causal relationship of three variables - demand, marketing and sales. In the experiment, demand is a confounder, marketing is the treatment and sales is the outcome. This can be seen as a simple generalization of what causes sales and how a treatment and the outcome can be affected by the environment.

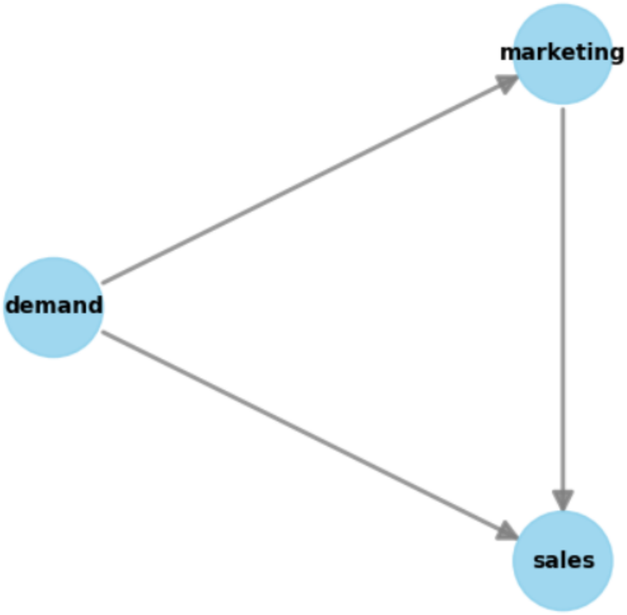


Figure 4.5: Directed acyclic graph representing the causal relationship between demand, marketing and sales. Demand plays the role of a confounder, marketing is the treatment and sales is the outcome.

The next step of the experiment is the generation of the data with the help of the numpy package. Ten thousand entries are generated with different averages and the data generation formulas state the relationships shown on Figure 4.5. Finally, the values are rounded with the assumption that there are no partial sales for the considered product, same for demand and marketing units.

```
demand = np.random.gamma(10, scale=1, size=10000)
marketing = demand + np.random.gamma(8, scale=1, size=10000)
sales = demand + 0.6 * marketing + np.random.gamma(3, scale=1, size=10000)
data = pd.DataFrame(dict(demand=np.round(demand),
marketing=np.round(marketing), sales=np.round(sales)))
```

In the next step of the experiment, the dataset is split into a training and test set with the help of the sklearn train_test_split method with ratio 80
20. This results in a training set of length 8000 entries and a test set of length 2000 entries. The train set is used to fit the regression algorithms - XGBRegressor, SVR, RandomForestRegressor and LinearRegression. Predictions are made on the test set

with each algorithm and the mean squared error is calculated, shown in Table 5.1. To further evaluate the performance of the regressors, the predicted values for sales are plotted together with the original values of sales from the test set, shown on Figure 5.1. This is done with the matplotlib Python package.

At this point, trained regression algorithms are available and a situation that requires a data-driven decision can be simulated. For instance, demand is estimated to have the value of 5 and management tries to decide what values of marketing would result in acceptable values for sales. Given the decided upon demand value and different marketing values, the regressors are used to make predictions for sales, so that an optimal decision for marketing can be made. The results are listed in Table 5.2.

In the real world, assumptions are made. That the distribution stays the same is one such assumption. Another assumption is that causality relationships persist. If the regressors were to be used for another shop, in another region, or the same shop but after the shop has been moved, the regressor might not provide trustworthy predictions. In this context, a new situation is simulated. The new causal relationships are shown in Figure 4.6.

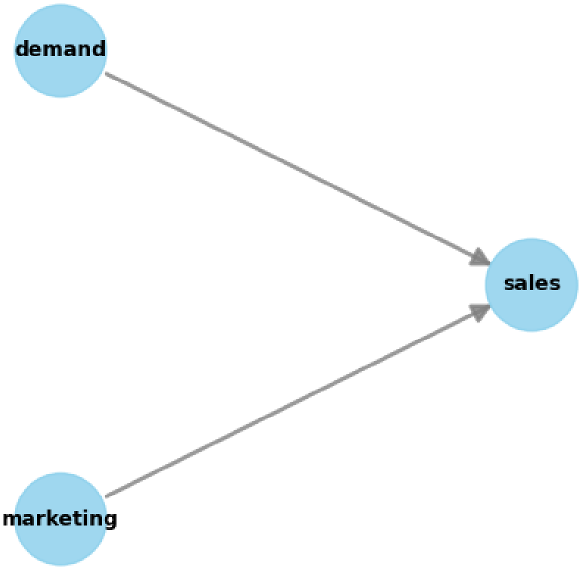


Figure 4.6: Directed acyclic graph representing the causal relationship between demand, marketing and sales. In this case, demand is assumed to affect just the sales, where marketing is the treatment and sales is the outcome.

According to the new causal model, another dataset is generated. Transitioning from the first dataset to this new dataset can be seen as intervention, since the algorithms used on the new dataset are trained on the first dataset.

```
demand_2 = np.random.normal(20, scale=1, size=10000)
marketing_2 = np.random.gamma(8, scale=1, size=10000)
```

```
sales_2 = demand_2 + 0.6 * marketing_2 + np.random.gamma(3, scale=1,
size=10000)
data_2 = pd.DataFrame(dict(demand=np.round(demand_2),
marketing=np.round(marketing_2), sales=np.round(sales_2)))
```

There are two differences in the data generation. These differences can be considered equivalent to deliberate changes. The first intervention or change is in the distribution of demand values and the second is that these values are not added to the marketing values.

The new dataset is split into a train and test set like the first dataset and the regression algorithms are used to predict on the test set. The predictions are used to calculate the mean squared error and the predicted values are plotted together with the test values on Figure 5.2.

In the transition from the first to the second dataset, two violations are introduced when using the fitted models on the second dataset - a violation of causal relationships and a violation of the assumption that the values are independent and identically distributed (IID). This is a simulation of non-IID situation. IID cannot be assumed in the first place due to the interventions. Another intervention takes place, a third dataset is generated and the causal relationship from the first dataset is applied, shown on Figure 4.5.

```
demand_3 = np.random.normal(20, scale=1, size=10000)
marketing_3 = demand_3 + np.random.gamma(8, scale=1, size=10000)
sales_3 = demand_3 + 0.6 * marketing_3 + np.random.gamma(3, scale=1,
size=10000)
data_3 = pd.DataFrame(dict(demand=np.round(demand_3),
marketing=np.round(marketing_3), sales=np.round(sales_3)))
```

The resulting third dataset is split into a training and test set. The regression algorithms trained on the first dataset make predictions on the new test set. Similar to before, the predicted values are plotted against the test set, shown on Figure 5.3 and errors are calculated, listed in Table 5.4.

The first experiment concludes with the estimation of the causal effect. This is done with the methodologies provided by the dowhy Python package. The first step is building a CausalModel object. The object uses the first dataset that was generated. Marketing is set to be the treatment, sales is the outcome and demand is marked as a common cause. Then, the causal effect is identified as a backdoor estimand. The estimand is used estimate the effect. This step requires the selection of a method - backdoor.linear_regression. This method estimates the mean value of the causal effect of the treatment, visualized on Figure 5.4.

The dowhy package supports different methods to check the robustness of the estimand. In this experiment, the random common cause method and the placebo treatment refuter method are employed. As the name of random common cause implies, the method introduces a random common cause between the treatment and the outcome. Then, the estimated causal effect is tracked for significant changes. The second refute method, placebo treatment, simulates the replacement of the treatment variable with a placebo treatment that is expected to have no effect on the outcome. After the replacement, the causal effect is estimated on the placebo modified data. Then, a check is done to indicate whether the new causal effect is statistically different from the original.

The causal directed graph, on Figure 4.5, can be used to create a StructuralCausalModel object. After automatically assigning causal mechanisms based on the data and data is fitted to the model, arrow strengths can be calculated for each non-root node. In this experiment, the arrow strengths are calculated for the marketing node and the sales node via the arrow_strength method.

Finally, interventional samples are drawn for sales given demand and marketing values. To compare the sales samples to the values predicted by the regression algorithms, the same values for demand and marketing are given. The predicted values are listed in Table 5.2 and the interventional sales samples are listed in Table 5.5.

4.3 Introducing complexity to the initial experiment

The purpose of the second experiment is to take the causal model from the first experiment and to make it more realistic. This is achieved by adding more complexity to the causal relationship and more variables. The graph from Figure 4.5 is extended with a new confounder called external marketing, an instrument variable temperature and promotion is added as a mediator between the marketing treatment and the sales outcome. The new graph is shown on Figure 4.7.

Data is generated according to variable relationships shown on Figure 4.7. Other additions are noise and outliers. Some noise is added to each variable and outliers are added to demand and sales. The scatter plots for the resulting data are shown on Figure 4.4. The number of entries for each variables is 10 000, same as in the first experiment.

```
demand = np.random.gamma(10, scale=1, size=SIZE) +
    np.abs(np.random.normal(2, 1, SIZE))
outlier_indices = np.random.choice(10000, 10, replace=False)
demand_outliers = np.random.choice(np.arange(50, 80), 10, replace=False)
demand[outlier_indices] = demand_outliers
ext_marketing = np.random.normal(4, 1, size=SIZE) +
    np.abs(np.random.normal(3, 2, SIZE))

time = np.arange(SIZE)
daily_variation = 10 * np.sin(2 * np.pi * time / 24)
seasonal_variation = 5 * np.sin(2 * np.pi * time / 365)
temp_noise = np.random.normal(0, 2.5, SIZE)
temperature = 14 + daily_variation + seasonal_variation + temp_noise

marketing = demand + 0.15 * ext_marketing + 0.1 * temperature +
    np.random.gamma(8, scale=1, size=SIZE)
promotion = 0.4 * marketing + np.random.normal(6, 2, SIZE) +
```

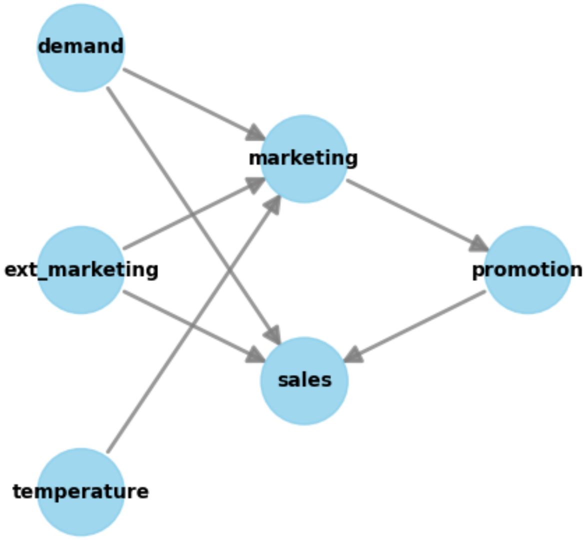


Figure 4.7: Extended directed acyclic graph representing the causal relationship between demand, external marketing, temperature, marketing, promotion and sales. Demand and external marketing are confounders and can be considered environmental variables. Temperature is an instrument variable unaffected by anything else in the causal model and affects only the treatment. Marketing is the treatment and has an indirect effect on sales through the mediator promotion.

```
    np.random.normal(4, 3, SIZE)

sales = demand + 0.05 * ext_marketing + 1.3 * promotion +
    np.random.gamma(3, scale=1, size=SIZE) + np.random.normal(2, 2, SIZE)
outlier_indices = np.random.choice(10000, 10, replace=False)
sales_outliers = np.random.choice(np.arange(540, 600), 10, replace=False)
sales[outlier_indices] = sales_outliers
```

The data is divided into features and a target. The target is sales and the rest of the variables are considered features. The data is further split into a training set and a test set, the split ratio is 80/20. The regression algorithms are fitted with the training set. The resulting models are used to make predictions on the test set and the mean squared errors are calculated.

Another set of results are generated by preprocessing for outliers before fitting the regression algorithms. The Interquartile Range method, a statistical technique, is used to filter out outliers. The method works by calculatating the first and third quartiles for a variable and finding the range, the difference between the third quartile and the first. A value is considered an outlier if it is smaller than the value of the first quantile subtracted by the 1.5 of the interquantile range value or bigger than the value of the third quartile added to 1.5 of the interquartile range value.

The preprocessed dataset is split into a training set and a test set. The ratio used for the split is 80/20 and the training set is used to fit the algorithms. The regressors make predictions for the test set and the predicted values are plotted against the test set. The mean squared errors are calculated.

The experiment continues with the model being used to make a data-driven decision in a simulated situation. A constant value is chosen for demand, external marketing and temperature. Three combinations of marketing and promotion are chosen and sales are predicted.

Following the structure of the first experiment, the causal model is changed, shown on Figure 4.8. In this intervention, the confounding effect that demand and external marketing have on marketing is removed.

The purpose of the interventions is to introduce two violations in the assumptions made when the fitted algorithms are used. The first violated assumption is that causal relationships don't change and the second violation is the introduction of simulated non-IID. A second dataset is generated according to the graph shown on Figure 4.8 as a result of the interventions. Furthermore, the distribution for demand is changed from gamma to normal distribution and the resulting dataset is preprocessed for outliers with the Interquartile Range technique.

```
# data + noise
demand_2 = np.random.normal(20, scale=1, size=SIZE) +
    np.abs(np.random.normal(2, 1, SIZE))
outlier_indices = np.random.choice(10000, 10, replace=False)
demand_outliers = np.random.choice(np.arange(50, 80), 10, replace=False)
```

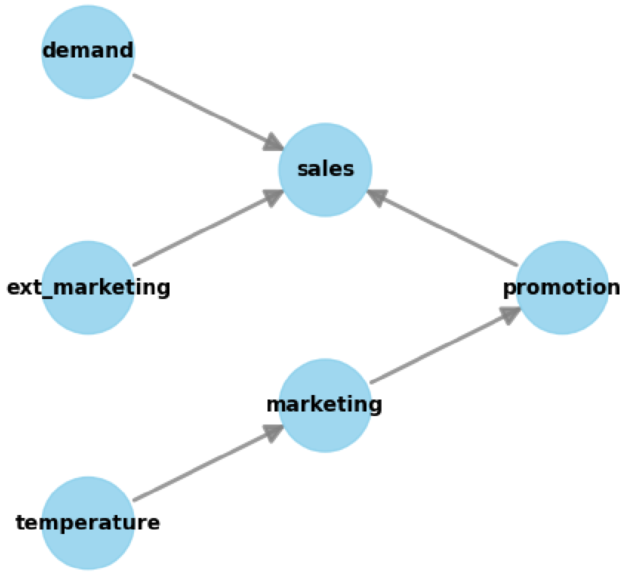


Figure 4.8: Extended directed acyclic graph representing the causal relationship between demand, external marketing, temperature, marketing, promotion and sales. The causal model is the same as on Figure 4.7 but without the confounding effect that demand and external marketing have on marketing.

```
demand_2[outlier_indices] = demand_outliers
ext_marketing_2 = np.random.normal(4, 1, size=SIZE) +
    np.abs(np.random.normal(3, 2, SIZE))

# same temperature

marketing_2 = 0.1 * temperature + np.random.gamma(8, scale=1, size=SIZE)
promotion_2 = 0.4 * marketing_2 + np.random.normal(6, 2, SIZE) +
    np.random.normal(4, 3, SIZE)

sales_2 = demand_2 + 0.05 * ext_marketing_2 + 1.3 * promotion_2 +
    np.random.gamma(3, scale=1, size=SIZE) + np.random.normal(2, 2, SIZE)
outlier_indices = np.random.choice(10000, 10, replace=False)
sales_outliers = np.random.choice(np.arange(540, 600), 10, replace=False)
sales_2[outlier_indices] = sales_outliers

data_2 = pd.DataFrame(dict(demand=np.round(demand_2),
    ext_marketing=np.round(ext_marketing_2), temperature=np.round(temperature),
    marketing=np.round(marketing_2), promotion=np.round(promotion_2),
    sales=np.round(sales_2)))
data_2.describe()
```

The preprocessed second dataset is split into a train set and a test set, with ratio 80/20. The regressors fitted with the train set from the first data set are used to make predictions with test set from the now altered second dataset. The predicted values are plotted against the values from the test set and the mean squared errors are calculated for the new predictions.

To investigate further the causal impact of such changes, a third dataset is generated through an intervention on the second dataset. The causal relationships between variables are changed to be the same as in the first dataset, shown on Figure 4.7.

```
# data + noise
demand_3 = np.random.normal(20, scale=1, size=SIZE) +
    np.abs(np.random.normal(2, 1, SIZE))
outlier_indices = np.random.choice(10000, 10, replace=False)
demand_outliers = np.random.choice(np.arange(50, 80), 10, replace=False)
demand_3[outlier_indices] = demand_outliers
ext_marketing_3 = np.random.normal(4, 1, size=SIZE) +
    np.abs(np.random.normal(3, 2, SIZE))

# same temperature
marketing_3  = demand_3 + 0.15 * ext_marketing_3 + 0.1 * temperature +
    np.random.gamma(8, scale=1, size=SIZE)
promotion_3  = 0.4 * marketing_3 + np.random.normal(6, 2, SIZE) +
    np.random.normal(4, 3, SIZE)

sales_3 = demand_3 + 0.05 * ext_marketing_3 + 1.3 * promotion_3 +
    np.random.gamma(3, scale=1, size=SIZE) + np.random.normal(2, 2, SIZE)
outlier_indices = np.random.choice(10000, 10, replace=False)
sales_outliers = np.random.choice(np.arange(540, 600), 10, replace=False)
sales_3[outlier_indices] = sales_outliers

data_3 = pd.DataFrame(dict(demand=np.round(demand_3),
    ext_marketing=np.round(ext_marketing_3), temperature=np.round(temperature),
    marketing=np.round(marketing_3), promotion=np.round(promotion_3),
    sales=np.round(sales_3)))
```

As before, the third dataset is preprocessed for outliers with the Interquartile Range method. Then, the dataset is split into a training set and a test set, with ratio 80/20. The regression algorithms trained on the train set from the first data set is used to predict for the new test set. The predictions are plotted against the values from the test set and the mean squared errors are calculated.

In the following part of the experiment, the causal effect of marketing on sales is estimated with the first dataset. The marketing data is labeled as treatment, temperature data as instrument, sales as outcome, demand and external marketing as common causes. A call to the indetify_effect method identifies an estimand and a call to the estimate_effect method calculates the effect. The estimated effect is plotted as a line on the scatter plot of the treatment and outcome.

The robustness of the estimated effect is evaluated with the random common cause method and the placebo refuter method.

The experiment continues with the initialization of a StructuralCausalModel with the causal graph from Figure 4.7. Causal mechanisms are automatically assigned to the model with the initial dataset and the model is fitted with the data. Arrow strengths are estimated for marketing, promotion and sales. The experiment concludes with getting interventional samples for sales via the StructuralCausalModel object. The variable values are the same as in the simulated situation which requires data-driven decision making, shown in Table 5.7.

5 Results

5.1 Initial experiment

The mean squared errors calculated for the predictions on the first dataset for each regression algorithm are shown in Table 5.1. On Figure 5.1, the predicted values are plotted against the test values. The scatter plots use marketing values for the x-axis and sales for the y-axis. The original values are colored blue and the predicted red.

Table 5.1: The table contains the mean squared error for each regression algorithm and the mean baseline. The values are similar, indicating that each algorithm was able to capture the linear relationship.

Algorithm	**MSE**
XGBRegressor	3.242
SVR	3.196
RandomForestRegressor	3.414
LinearRegression	3.144
Mean Baseline	32.12

The simulated data consists of constant value of 5 for demand and increasing values for marketing. This data includes the predictions of each algorithm, shown in Table 5.2.

Table 5.2: The table lists sales predictions for a given constant demand and increasing marketing values. Sales are predicted with XGBRegressor, SVR, RandomForestRegressor and LinearRegression.

demand	**marketing**	**xgb_sales**	**svr_sales**	**rfr_sales**	**lr_sales**
5	5	9.0	11.0	14.0	11.0
5	10	14.0	14.0	14.0	14.0
5	15	17.0	17.0	18.0	17.0

After the first intervention, predictions on test set from the second dataset, with causal relationships shown on Figure 4.6, results in errors shown in Table 5.3. Scatter plots of the predicted values with each algorithm are shown on Figure 5.2. Furthermore, the predictions of the models are plotted against the test set on Figure 5.2. The scatter plots use marketing as x-axis and sales as y-axis.

Table 5.4 contains the mean squared errors for predictions on a test set from the third dataset, where an intervention changed the causal relationships to the same as in the first dataset. The scatter plots of the predicted values are visible on Figure 5.3.

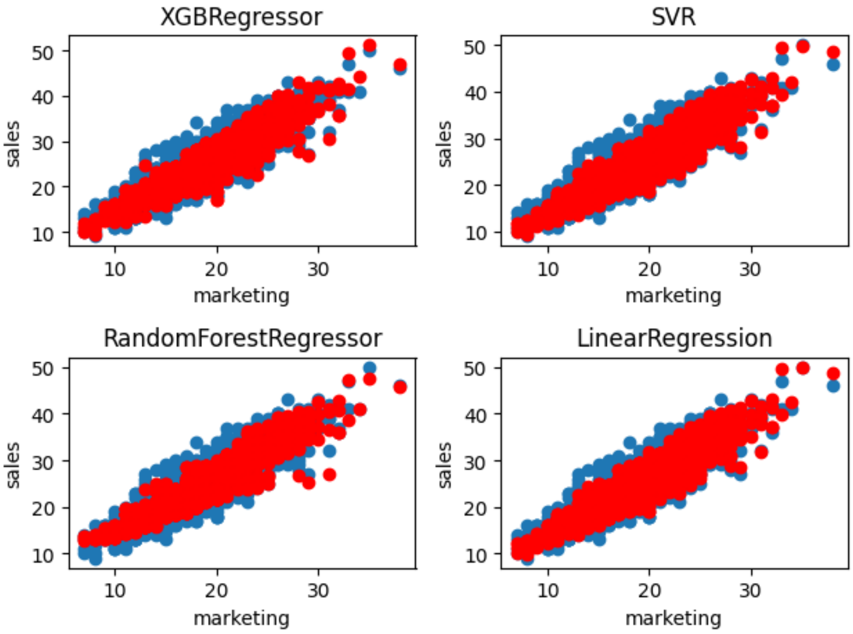


Figure 5.1: Scatter plots of the test data and the predicted values for that data. The plots indicate that each model was able to capture the relationships between the variables well. Original values are blue and predicted values are red.

Table 5.3: The table contains the mean squared error values calculated for the predictions on the test set from the second dataset. In this dataset the distribution for demand is changed to normal, the demand mean is doubled and the causal relationships are changed.

Algorithm	**MSE**
XGBRegressor	20.42
SVR	3.17
RandomForestRegressor	17.58
LinearRegression	3.18

Table 5.4: The table contains the mean squared error values calculated for the predictions on the test set from the third dataset. In this dataset the distribution for demand is normal and the mean value given in the generation function is doubled, however, the causal relationship is the one shown on Figure 4.5

Algorithm	**MSE**
XGBRegressor	4.43
SVR	3.13
RandomForestRegressor	3.99
LinearRegression	3.07

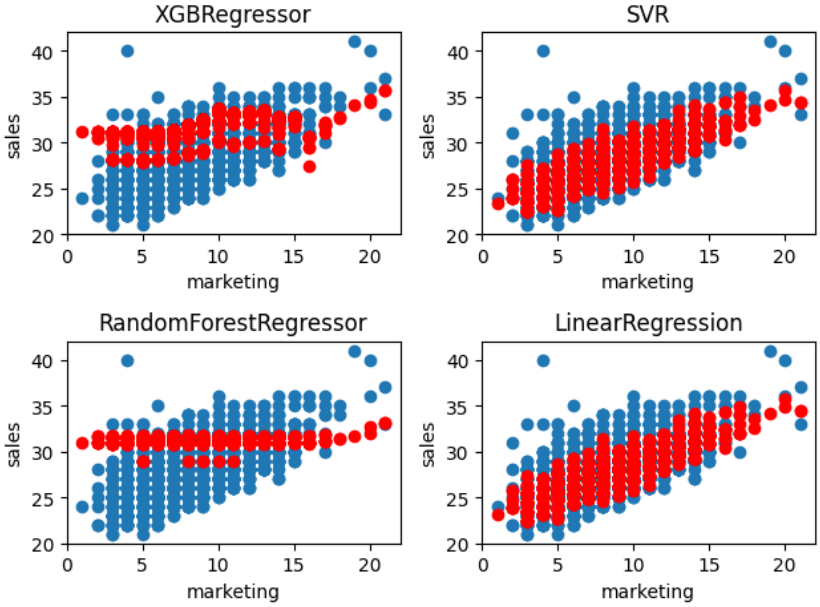


Figure 5.2: Scatter plot of the test set and the predicted values. The test set is split from the second dataset. This dataset has the causal relationships shown on Figure 4.6. The test set sales values are plotted in blue and the predicted values in red.

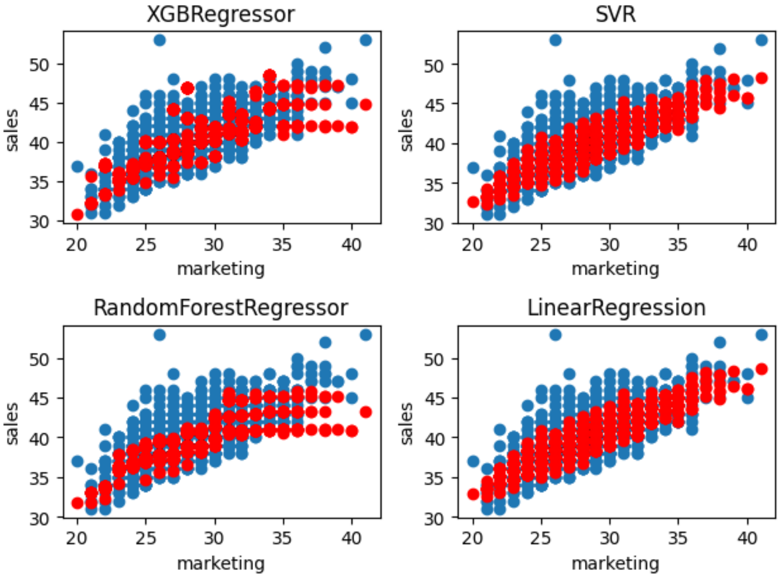


Figure 5.3: Scatter plots of the test data and the predicted values. The test set is split from the third dataset. This dataset has the same variable relationships as the first dataset and distribution of demand identical to the one in the second dataset. The test set sales values are in blue and the predicted values in red.

An effect identification function from the dowhy library for a causal model with marketing as a treatment, sales as an outcome and demand as a common cause detects a backdoor. The identification results in an expression that takes the derivative with respect to marketing representing the outcome change when marketing is different. The second part of the expression is a conditional expectation which controls the effect of demand.

$$\frac{d}{d[marketing]}(E[sales|demand])$$

An estimation of the causal effect of marketing with a backdoor linear regression method results in 0.611. The causal effect is plotted on Figure 5.4. This estimation is done for the first synthetic dataset.

The estimated causal effect is refuted to check its robustness. Refuting with the random common cause method estimates a new effect of 0.611 with a p-value of 0.96. The p-value indicates that the new effect is not significantly different then the already estimated one. The placebo treatment method estimates a new effect of 0.0004 and a p-value of 0.47. In this case, the effect appears different, however, the p-value provided by the method suggests otherwise.

A StructuralCausalModel object for the data provides the arrow strengths for the causal graph on Figure 4.5. The arrow strength from demand to marketing is estimated to be 10.37 indicating that one unit change in demand results in an average change of

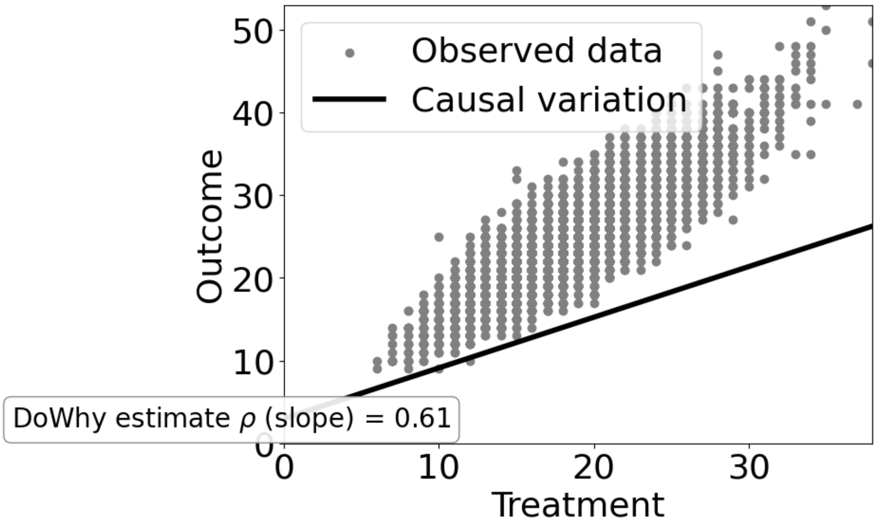


Figure 5.4: Scatter plot of sales as outcome and marketing as treatment. The plot contains the causal effect estimated for marketing via the dowhy package.

10.37 units of marketing. Other arrow strengths reveal that one-unit change in demand is 10.1 units of change in sales. Finally, one-unit change in the treatment marketing is associated with an average change of 6.78 units in sales.

The results from the first experiment conclude with the interventional samples for sales drawn via the StructuralCausalModel object. The results are listed in Table 5.5.

Table 5.5: The table contains interventional samples for sales based on the same values for demand and marketing used in the simulated situation, shown for predictions in Table 5.2.

demand	**marketing**	**sales**
5	5	10
5	10	16
5	15	17

5.2 Extended experiment with additional complexity

Following the structure of the initial experiment, the results of the extended experiment consist of mean squared errors, plots and causal effect estimates. Table 5.6 contains the mean squared error values for predictions on a test set split from the first extended dataset. The results are generated without preprocessing for outliers and once after preprocessing. The rest of the predictions are generated by using the models fitted with the preprocessed dataset.

Removing the outliers with the Interquartile Range method applied to demand and

Table 5.6: The table contains the mean squared error values calculated for the predictions on the test set from the first dataset in the extended experiment. One set of errors is calculated with models fitted with the preprocessed data and another set after preprocessing.

Algorithm	**MSE with outliers**	**MSE w/o outliers**
XGBRegressor	375.25	8.9
SVR	272.24	7.73
RandomForestRegressor	286	8.41
LinearRegression	271.84	7.74
Mean Baseline	319.65	48.89

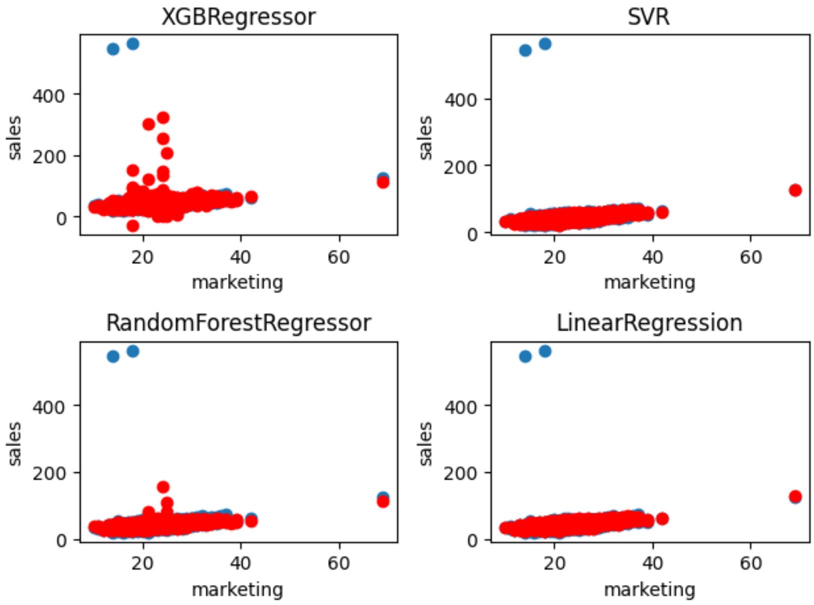


Figure 5.5: Scatter plots of the predicted values against the test values. The predicted values are in red and the blue values are the values from the test set. The data is not preprocessed and outliers are visible on the plots. The XGBRegressor and the RandomForestRegressor seem to be affected more by the outliers.

sales, reduces the number of entries in the dataset to 9779. The predicted values are plotted against the test values on Figure 5.6.

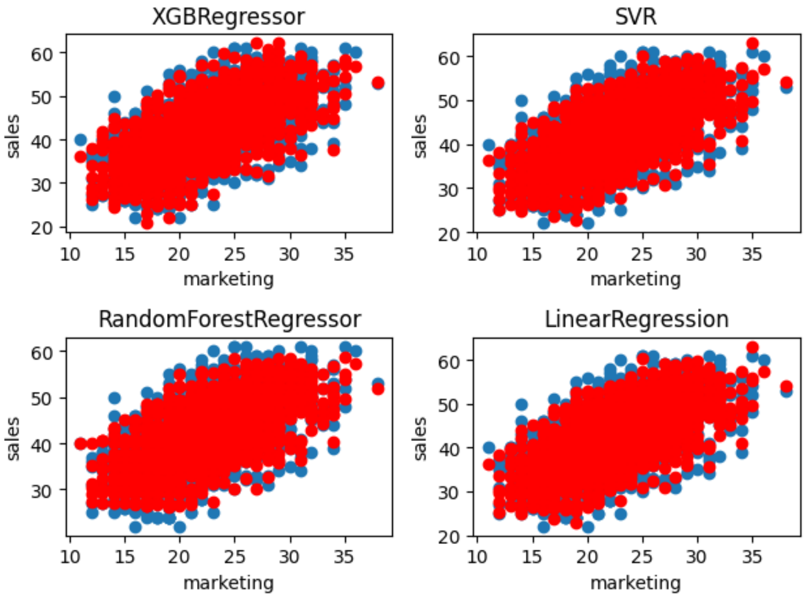


Figure 5.6: Scatter plot of the predicted values against the test values. The predictions are made after preprocessing for outliers with the Interquartile Range method.

The situational data of constant demand of 10, external marketing of 4, temperature of 20, increasing marketing treatment and increasing promotion mediator results in prediction shown in Table 5.7. Promotion despite not being the treatment changes with marketing to keep the causal relationship since it is a mediator, shown on Figure 4.7.

Table 5.7: The table contains sales predictions for given constant demand of 10, external marketing of 4 and temperature of 20. The values for the treatment marketing and the mediator promotion increase.

marketing	**promotion**	**xgbr_sales**	**svr_sales**	**rfr_sales**	**lr_sales**
20	18	38.0	39.0	40.0	39.0
40	36	57.0	62.0	54.0	62.0
70	50	57.0	80.0	54.0	80.0

After the first intervention, predicting on the second dataset results in the mean squared errors listed in Table 5.8. The predictions are further shown on Figure 5.7 together with the test values.

The prediction results conclude with mean squared errors and plots on a test set split from the third dataset, where an intervention changes back the causal relationships to

Table 5.8: The table contains the mean squared error values calculated for the predictions on the test set from the second dataset in the extended experiment. The third dataset has the same causal relationship as the first and demand distribution as the second dataset.

Algorithm	**MSE**
XGBRegressor	29.88
SVR	7.81
RandomForestRegressor	32.44
LinearRegression	7.42

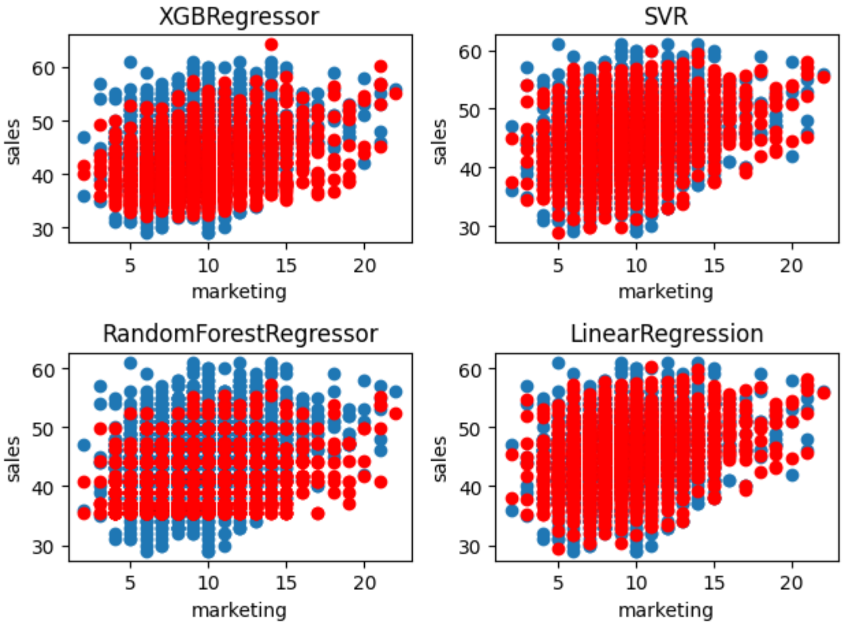


Figure 5.7: Scatter plots of the predicted values against the test values. The test set is from the second synthetic dataset and the data follows the causal variable relationships from Figure 4.8. The distribution for demand is normal and the mean value is doubled.

the same as in the first dataset. The demand distribution is the same as in the second dataset. The errors are listed in Table 5.9 and the predictions are shown on Figure 5.8.

Table 5.9: The table contains the mean squared error values calculated for the predictions on the test set split from the third dataset in the extended experiment. The third dataset has causal relationships like the first dataset and normal distribution for demand as in the second dataset.

Algorithm	**MSE**
XGBRegressor	20.13
SVR	7.02
RandomForestRegressor	30.16
LinearRegression	6.97

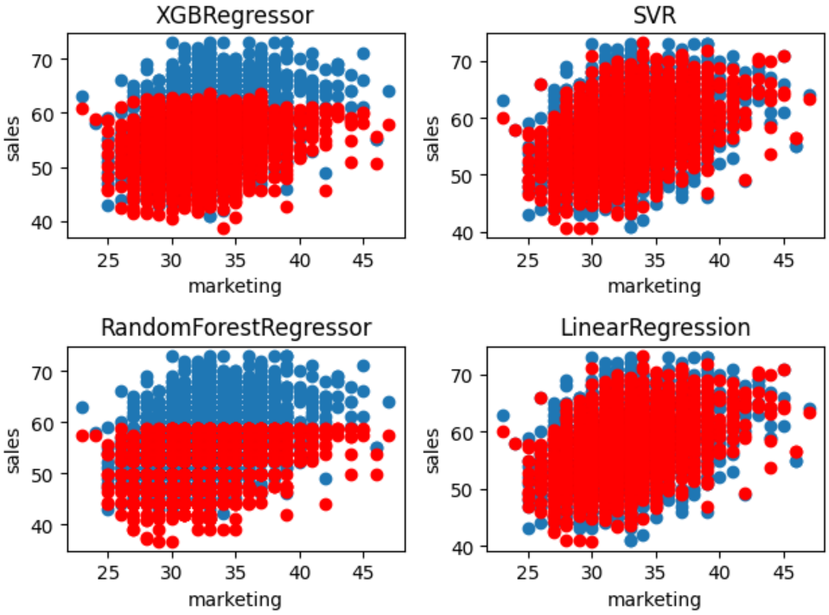


Figure 5.8: Scatter plots of the predicted values against a test set split from the third dataset whose variable relationships are depicted on Figure 4.7. The distribution for demand is normal and the mean doubled compared to the first dataset. The regression algorithms are fitted with a train set from the first synthetic dataset.

Further results are provided by investigating the causal effect of marketing with dowhy. Identifying the causal effect for a CausalModel object with marketing as a treatment, temperature as an instrument, sales as an outcome and common causes demand and external marketing. The method detects a backdoor estimate with expression:

$$\frac{d}{d[marketing]}(E[sales|demand, marketing_{external}])$$

This expression derives the expected value of sales with respect to marketing, while controlling for demand and an external marketing. Estimates and expressions are found for the temperature instrument variable and a frontdoor for the promotion mediator. Investigating the causal effect further with a backdoor linear regression method estimates an effect for marketing of 0.455. The effect is plotted on Figure 5.9.

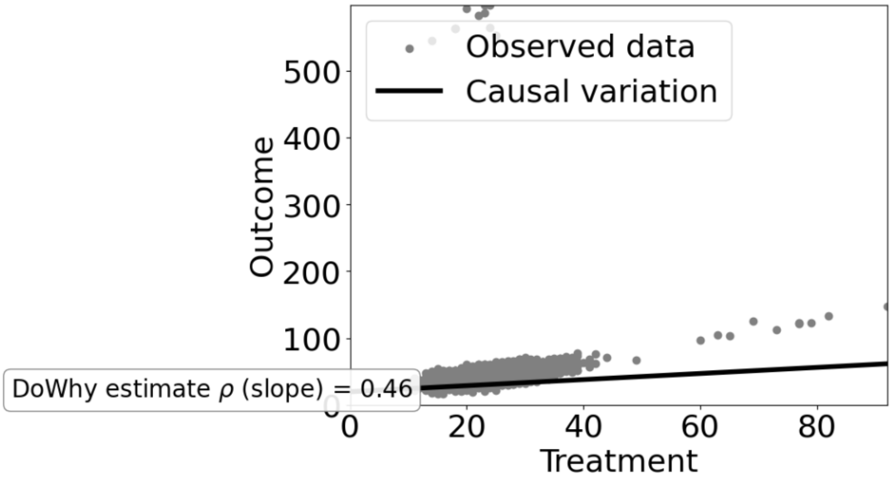


Figure 5.9: Scatter plot of sales as an outcome and marketing as a treatment for the first dataset. The plot contains the causal effect estimated for marketing via the dowhy package.

Refuting the the causal effect of marketing with the common cause method results into the same effect of 0.455 and a p-value of 0.82, indicating no significant difference between the estimated causal effect and the new effect. The new effect is calculated to be -0.003 when refuted with the placebo treatment method. However, a p-value of 0.82 indicates no significant difference between the estimated effect of 0.455 and the new effect -0.003.

Arrow strengths are calculated with a StructuralCausalModel object for a the graph on Figure 4.7. One-unit change in demand is having an average 15.07 unit change effect on marketing. Changing a unit in external marketing has a unit change of 0.16 on marketing. A one-unit change in temperature has an average unit change of 0.69 on marketing. A single unit of marketing leads to an average change of 3.47 units of promotion. The final arrow strengths to sales indicate that a unit of demand can have a change of 33.23 units of sale, external marketing unit can change sales by 5.37 and one-unit change in promotion can lead to a 15.62 unit change in sales.

The results of the second experiment conclude with drawing interventional samples for sales via a StructuralCausalModel object. The values are listed in Table 5.10. The sales samples are drawn for a constant value of 10 for demand, 4 for external marketing, 20 for temperature and similarly changing values for marketing and promotion.

Table 5.10: The table contains drawn interventional samples (via a StructuralCausalModel) for sales, given constant demand, external marketing and temperature. The values for marketing and promotion change similarly. The table shows data that can be used to make a data-driven decision.

demand	**ext_marketing**	**temperature**	**marketing**	**promotion**	**sales**
10	4	20	20	18	41
10	4	20	40	36	62
10	4	20	70	50	81

6 Discussion

To evaluate the importance of causal learning for a digital twins system, the predictions of machine learning algorithms have to be analysed and evaluated. The XGBRegressor, SVR, RandomForestRegressor and LinearRegression seem to perform decently on the first regression task in the initial experiment. The mean squared errors are close to 3.2 for each of the algorithms and their errors are much smaller compared to the error of the mean baseline which is 32.12. The scatter plots on Figure 5.1 indicate that there is no significant difference in the way the algorithms make their predictions. The limitations in this part of the experiment are mainly in the simplicity of the data - the data is clean and the relationship between marketing and sales is linear. This is rectified to some extend in the extended experiment where noise and outliers are introduced together with more variables and more complex variable relationships.

In the real world, regression algorithms are useful when predicting future sales depending on a planned marketing campaign and estimated demand. Predictions can suggest how management should proceed with their investments in marketing. This is illustrated with the the results in Table 5.2. When demand and marketing are 5, the most diversity in predictions is observed. In the case of marketing being 10, the models predict the same value of 14. In the third scenario, all models but the RandomForestRegressor predicted a value of 17.

Using the models on the second dataset, which results from intervening on causal relationships and distribution of demand, indicates that the XGBRegressor and the RandomForestRegressor models are strongly affected by the introduced changes. Their mean squared errors are more than 5 times larger. The SVR and LinearRegression models on the other hand seem to be only slightly affected. This is further illustrated on Figure 5.2. The predictions made by the SVR and LinearRegression models overlap better with the values from the test set than the predictions made by the other two models.

In the real world, a model may be fitted with data from a shop and this shop may have data similar to the first dataset. Then, the shop might be moved to a new location or the fitted model may be used for another shop. The data for the new location or shop may differ from the first shop or location the way the first synthetic dataset differs from the second, hence the interventions. The results of this experiment indicate that algorithms such as XGBRegressor and RandomForestRegressor may not be able to perform well on the new data. Different causality relationships and variable distributions might result in predictions that cannot be trusted due to a higher error.

Intervening on the second dataset changes the causal relationships to the same as in the first dataset and results in the third dataset. The mean squared errors for predictions from a split from this set are in Table 5.4. The mean squared errors are close to the errors for predictions on the test set from the first dataset. This indicates that causal relationships between variables may have a bigger impact on predictions than a change in distribution highlighting the importance of causality. For digital twins systems, the results underline the importance of causal inference and learning if said system is to include machine learning components.

In the experiment, the causal relationships are well-known due to using synthetic data and knowing how the data was generated. For instance, it is known that:

```
sales = demand + 0.6 * marketing + np.random.gamma(3, scale=1, size=10000)
```

The effect of marketing is defined as 0.6. The following results aim to reveal how the causal effect can be estimated. The results reveal that having sufficient domain knowledge allows for the estimation of results very close to the target. Domain knowledge seems to play a crucial role indicated by the need to select the right role for the variables in a causal model. For instance, in the creation of a CausalModel object, it is indicated which variable is the treatment, outcome, instrument and common cause. Selecting the right variables for treatment and outcome can be simple, however, naming all important common causes requires knowledge that in many cases is not available or requires human expertise [Guo+20]. In the case that variables are labeled sufficiently well, the results reveal that dowhy is able to find the type of estimate from the CausalModel object, namely backdoor given the demand confounder. Using the identified type of estimate, which correctly evaluated in this case, allows for the estimation of the causal effect from the data, calculated to be 0.611. The estimate is close to the original value of 0.6. A line chart indicates the estimated causal effect on the observed data, shown on Figure 5.4.

The estimated value of 0.611 is evaluated for robustness with two methods. The random common cause refute method returns the same value of 0.611 as a causal effect for marketing. This means that adding a new common confounder to the equation does not make the estimated effect different. This is further supported by the interpretation of the 0.96 p-value. P-values higher than 0.05 indicate that the any observed difference between the estimated effect and the new effect after the additional of a random confounder are not significantly different.

The results of the arrow strengths summarise the effect of variables on other variables in terms of units. The arrow strengths reveal that one-unit change in demand has a bigger effect on sales than one-unit change in marketing.

The interventional samples drawn for sales can be compared to the predictions made by the fitted models, shown in Table 5.2. The first sales sample of 10 is only different by a single unit compared to the predictions made by the XGBRegressor, SVR and LinearRegression. The sample for sales is 16 when demand is 5 and marketing is 10. This is 2 units more than the prediction made by the regressors. The final sample of 17 is the same as the predictions. The results indicate that the regressors evaluate the impact of marketing on sales to be a weaker than the StructuralCausalModel does. Further research may reveal why this is the case.

The extended experiment with additional complexity underlines the importance of preprocessing. In the given example, preprocessing with the IQR method for outliers shows how the mean squared error metric for evaluation of machine learning algorithms is strongly affected by outliers even if the predictions are close to the test values. The metric may be affected so much as to indicate that an algorithm is performing less well than baselines such as mean. Algorithms like the XGBRegressor and the RandomForestRegressor seem to be affected more by outliers than SVR and LinearRegression. Comparing the plots on Figure 5.5 and Figure 5.6 reveals that preprocessing makes the pattern in the data easier to see.

The prediction for sales with the fitted models, shown in Table 5.7, reveal that the XGBRegressor model and the RandomForestRegressor end up with a greater weight for the confounders than simpler algorithms such as SVR and LinearRegression.

The findings of the initial experiment with regards to using using the models on data with different causal relationships and distributions are supported in the extended experiment. Predicting on the second dataset results in predictions with a bigger error for XGBRegressor and RandomForestRegressor. Predicting on the third dataset sees these errors slightly reduced for RandomForestRegressor and reduced by almost 33% for XGBRegressor. Comparing the plots in Figures 5.7 and 5.8 for the two algorithms may be interpreted as the opposite.

Identifying the causal effect for the new data further points the relevance of domain knowledge, detecting three estimate types - backdoor, instrument and frontdoor. The additional estimate types are detected due to the instrument temperature and the mediator promotion.

The causal inference with a backdoor linear regression method, which takes into consideration the confounders, estimates a causal effect for marketing of 0.455. The data generation formulas for promotion and sales are:

```
promotion = 0.4 * marketing + np.random.normal(6, 2, SIZE) +
np.random.normal(4, 3, SIZE)
sales = demand + 0.05 * ext_marketing + 1.3 * promotion +
np.random.gamma(3, scale=1, size=SIZE) + np.random.normal(2, 2, SIZE)
```

Marketing affects promotion with a strength of 0.4 and promotion affects sales with a strength of 1.3. This results in marketing affecting sales with a strength of 0.52 which is not very different from 0.455. The placebo treatment method and the random common cause method indicate that the estimated effect is quite robust. The arrow strengths calculated via the StructuralCausalModel indicate that marketing is strongly affected by demand and not so much by external marketing or temperature. A one-unit change in marketing means an average change of 3.47 units in promotion and a unit change in promotion results in an average change of 15.62 units in sales. The arrows strengths further reveal that demand has the most impact on sales which is further supported by the predictions made with XGBRegressor and RandomForestRegressor, visible in Table 5.7.

The extended experiment concludes with interventional samples for sales drawn via the StructuralCausalModel. The results are visible in Table 5.10, the same values are used in Table 5.7 to compare the causal model to the predictions made by the fitted models. The first interventional sample differs with 1 to 3 units from the predicted values. The second sample of 62 is the same as the predictions made by SVR and LinearRegression which differs from the predictions made by the other two models. The final interventional sample is close to the predictions made by SVR and LinearRegression. The results indicate that the StructuralCausalModel provides results similar to the results from the SVR and the LinearRegression models.

7 Conclusion

The development of digital twins system is often costly and complex[PIV23]. However, such systems are expected to improve the industry and provide more options for monitoring, maintenance and data-driven control.

The utilization of synthetic data for two distinct store scenarios underscores a deliberate effort to simulate real-world complexities and to highlight the importance of causal learning in parallel to machine learning. The limitations in data simplicity are noted. Furthermore, the data was generated with a given trend. Real-world data might not reveal any relationships between variables. The implementation and results sections present an exploration of experimental methodology and findings.

The initial experiment explores a simple causal model, examining the impact of marketing on sales. The research achieves low mean squared errors and coherent predictions on the test sets of the synthetic data. The comparison of the mean squared errors calculated after each internvetion on the data reveals the relevance of causal relationships in the causal model. Establishing the importance of causal effects, the experiment continues with effect estimation. The estimated effects are close to the expected values showcasing the worth of the tools and methodologies used in the experiment.

Building upon the first experiment, the extended experiment introduces additional complexity to the causal model, incorporating an additional confounder, a mediator and an instrument variable to better reflect real-world scenarios. The generation of the data includes adding noise and outliers. Performance evaluation of the fitted regressors shows the importance of preprocessing. Overall, the extended experiment results confirm the findings from the initial simpler experiment. Comparisons between machine learning algorithms and causal modeling approaches underscore the nuanced influence of different factors on predicting outcomes.

To summarize, the study shows the role of causal learning in enhancing the accuracy and robustness of predictive models.

Bibliography

[BV07] Hendrik Blockeel and Joaquin Vanschoren. "Experiment Databases: Towards an Improved Experimental Methodology in Machine Learning". In: *Knowledge Discovery in Databases: PKDD 2007*. Ed. by Joost N. Kok et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 6–17. ISBN: 978-3-540-74976-9.

[EBA20] Itxaro Errandonea, Sergio Beltrán, and Saioa Arrizabalaga. "Digital Twin for maintenance: A literature review". In: *Computers in Industry* 123 (2020), p. 103316. ISSN: 0166-3615. DOI: `https://doi.org/10.1016/j.compind.2020.103316`. URL: `https://www.sciencedirect.com/science/article/pii/S0166361520305509`.

[ES07] Frederick Eberhardt and Richard Scheines. "Interventions and Causal Inference". In: *Philos. Sci.* 74 (Dec. 2007). DOI: `10.1086/525638`.

[Ful+20] Aidan Fuller et al. "Digital Twin: Enabling Technologies, Challenges and Open Research". In: *IEEE Access* 8 (2020), pp. 108952–108971. DOI: `10.1109/ACCESS.2020.2998358`.

[Gru98] Janis Grundspenkis. "Causal domain model driven knowledge acquisition for expert diagnosis system development". In: *Journal of Intelligent Manufacturing* 9 (Jan. 1998), pp. 547–558. DOI: `10.1023/A:1008840303610`.

[Guo+20] Ruocheng Guo et al. "A Survey of Learning Causality with Data: Problems and Methods". In: *ACM Comput. Surv.* 53.4 (2020). ISSN: 0360-0300. DOI: `10.1145/3397269`. URL: `https://doi.org/10.1145/3397269`.

[Lin+21] Jinjiao Lin et al. "Domain Knowledge Graph-Based Research Progress of Knowledge Representation". In: *Neural Comput. Appl.* 33.2 (2021), 681–690. ISSN: 0941-0643. DOI: `10.1007/s00521-020-05057-5`. URL: `https://doi.org/10.1007/s00521-020-05057-5`.

[Liu+21] Mengnan Liu et al. "Review of digital twin about concepts, technologies, and industrial applications". In: *Journal of Manufacturing Systems* 58 (2021). Digital Twin towards Smart Manufacturing and Industry 4.0, pp. 346–361. ISSN: 0278-6125. DOI: `https://doi.org/10.1016/j.jmsy.2020.06.017`. URL: `https://www.sciencedirect.com/science/article/pii/S0278612520301072`.

[Liu+23] Zheng Liu et al. "Digital twin for predictive maintenance". In: Apr. 2023, p. 6. DOI: `10.1117/12.2660270`.

[Pim+21] João Felipe Pimentel et al. "Understanding and improving the quality and reproducibility of Jupyter notebooks". In: *Empirical Softw. Engg.* 26.4 (2021). ISSN: 1382-3256. DOI: 10.1007/s10664-021-09961-9. URL: https://doi.org/10.1007/s10664-021-09961-9.

[PIV23] Massimo Panarotto, Ola Isaksson, and Vanessa Vial. "Cost-efficient digital twins for design space exploration: A modular platform approach". In: *Computers in Industry* 145 (2023), p. 103813. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2022.103813. URL: https://www.sciencedirect.com/science/article/pii/S0166361522002093.

[Ple+22] Vagelis Plevris et al. "Investigation of performance metrics in regression analysis and machine learning-based prediction models". In: June 2022. DOI: 10.23967/eccomas.2022.155.

[SHJ13] Daniela Steidl, Benjamin Hummel, and Elmar Juergens. "Quality analysis of source code comments". In: *2013 21st International Conference on Program Comprehension (ICPC)*. 2013, pp. 83–92. DOI: 10.1109/ICPC.2013.6613836.

[SK20] Amit Sharma and Emre Kiciman. *DoWhy: An End-to-End Library for Causal Inference.* 2020. arXiv: 2011.04216 [stat.ME].

[SM18] Nenad Stojanovic and Dejan Milenovic. "Data-driven Digital Twin approach for process optimization: an industry use case". In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 4202–4211. DOI: 10.1109/BigData.2018.8622412.

[SSH17] Alaaeddin Swidan, Alexander Serebrenik, and Felienne Hermans. "How do Scratch Programmers Name Variables and Procedures?" In: *2017 IEEE 17th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. 2017, pp. 51–60. DOI: 10.1109/SCAM.2017.12.

[Tat21] Abhishek V Tatachar. "Comparative Assessment of Regression Models Based On Model Evaluation Metrics". In: *International Research Journal of Engineering and Technology (IRJET)* 08.09 (2021).

[Van16] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data.* 1st. O'Reilly Media, Inc., 2016. ISBN: 1491912057.

[VCB22] Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. "D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery". In: *ACM Comput. Surv.* 55.4 (2022). ISSN: 0360-0300. DOI: 10.1145/3527154. URL: https://doi.org/10.1145/3527154.

[vTC22] Raymon van Dinter, Bedir Tekinerdogan, and Cagatay Catal. "Predictive maintenance using digital twins: A systematic literature review". In: *Information and Software Technology* 151 (2022), p. 107008. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2022.107008. URL: https://www.sciencedirect.com/science/article/pii/S0950584922001331.