TRUSTAI

TRANSPARENT, RELIABLE
& UNBIASED SMART TOOL

D3.1: Saliency measures for identifying causally relevant variables of human-like explanations

Lead participant: University of Tartu

September 30, 2022

DOCUMENT CONTROL PAGE

DOCUMENT	**D3.1 – Saliency measures for identifying causally relevant variables of human-like explanations**
TYPE	**Report**
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	30/09/2022
DATE OF DELIVERY	30/09/2022
VERSION	0.5
DELIVERABLE RESPONSIBLE	UT
AUTHOR (S)	Marharyta Domnich (UT), Raul Vicente Zafra (UT), Eduard Barbu (UT)
OFFICIAL REVIEWER/s	Gonçalo Reis Figueira (INESC TEC), Fábio Neves-Moreira (INESC TEC)

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	University of Tartu	08/06/2022	Introduction to counterfactual formalisation, biases and evaluation strategy.
0.2	University Of Tartu	17/07/2022	Integrated the information about the causal graph and feature constraints received from Apintech.
0.3	University of Tartu	08/08/2022	Feature constraints received from LTPlabs and integrated into the deliverable.
0.4	University of Tartu	12/08/2022	Feature constraints received from CWI and integrated into the deliverable.

0.5	University of Tartu	29/09/2022	Reviewer (INESC TEC) corrections and suggestions integrated

ACKNOWLEDGEMENTS

NAME	PARTNER
Nikos Sakkas	AIT
Francisco Amorim	LTPlabs
Gonçalo Reis Figueira	INESC TEC
Fábio Neves Moreira	INESC TEC
Peter Bosman	CWI
Tanja Alderliesten	CWI
Evi Sijben	CWI

Executive Summary

The present deliverable *D3.1, "Saliency measures for identifying causally relevant variables of human-like explanations,"* formalises the human heuristics described in behavioural and cognitive studies of human causal reasoning as a series of saliency measures. It presents the causal approach for identifying causally relevant features, discusses feature saliency and model uncertainty. Our purpose is to simplify and prioritise explanations for the TRUST-AI platform. Moreover, the deliverable introduces the notion of the causal graph and causal inference. The causal graphs and the constraints specific to each use case are produced by experts or extracted from specialised databases. This knowledge constraints the counterfactual search and helps answer specific causal questions.

Table of Contents

Abbreviations and Acronyms

AI	Artificial Intelligence
EC	European Commission
EU	European Union
GDA	Gene-disease associations
HCXAI	Human-centred Explainable AI
HVAC	Heating, Venting and Air Conditioning
KPI	Key Performance Indicators
MAD	Median Absolute Deviation
SDH	Succinate Dehydrogenase
SHAP	Shapley Additive Explanations
UC	Use-case
VAE	Variational autoencoder
WP	Work Package
XAI	Explainable Artificial Intelligence

Introduction

Explanations are needed for informing and supporting human decision-making. They are creating a shared understanding between an algorithm and a human by increasing transparency. Behavioural studies have extensively investigated how humans produce and perceive explanations for diverse mechanistic and social phenomena [1]. Explanatory understanding includes not only the processes of creating and discovering explanations but also the processes of providing and receiving them [2].

In deliverable D2.1, we have explored through questionnaires and interviews the types of explanations the three use cases are looking for. We have found that they prefer causal, contrastive, counterfactual, and prototype-style explanations. These explanations should be provided through tables, charts, interactive graphics, and plain text, the counterfactual textual explanations having the highest interest. As an expert commented in a questionnaire, "A counterfactual explanation is the best way to model the impact of some decisions."

The counterfactual explanation answers how the input settings should have been different to get the expected outcome. It assumes that humans have some output in mind and answer the question "Why P occurred instead of hypothetical expected Q" [3]. Identifying causally relevant actionable variables for counterfactual explanation is the biggest challenge. We approach the problem by introducing saliency measures that are related to human cognitive biases as part of an objective function.

After discussing with the partners, we defined specific causal questions for each use case. In addition to counterfactual explanation, all use-cases would like a graphical representation of feature saliency.

The deliverable is structured in the following way: first, we formalise the counterfactual explanation search. Next, the constraints and biases that help to rank the counterfactual explanations according to a human-centric heuristic are explored and formalized as saliency measures. Evaluation strategy for counterfactual explanation is introduced. An essential part of an explanation is the identification of causal relations. The difference between predictive machine learning and causal inference is explained in section 2. The causal graph primitives are introduced, and the main steps of causal calculus are presented. The section 3 incorporates expert domain knowledge about the data and presents tentative causal graphs and the human biases and constraints used in ranking the counterfactual search. Finally, section 4 reviews other methods that improve user trust in the system, such as feature intervention and feature saliency. Additionally, incorporating model uncertainty was suggested as it improves building counterfactual explanations and helps to visualise the search space.

1. Formalisation of counterfactual search

This section will provide a counterfactual definition, introduce saliency measures that can be viewed as human cognitive biases which affect the preferences for selecting cause of explanation, and formalise counterfactual search space according to human biases and constraints. Additionally, counterfactual evaluation strategy and metrics will be discussed.

1.1 Counterfactual explanation definition

Counterfactual explanation can be thought of as the possible smallest change in input settings in order to get the desired model output that changes the model prediction [4]. A typical example is a person who applied for a loan and was rejected by the model. The company is willing to provide an explanation about its decision. Giving a simple feature importance overview does not guarantee that changing the most important feature would flip the prediction to desired outcome. It is also possible that most important causes are not actionable and are undesirable in explanations overall, such as age or gender. Therefore, we are looking for actionable alternatives that help to understand what can change the model decision.

Tim Miller in [3] suggest that counterfactual explanation always answer the following question:

> *Given two events P and Q, in some situation the fact P occurred and the explainee is asking why foil Q did not occur in that situation instead. Given the situation* $(M, \overline{u})$
>
> $$Why\ (M, \overline{u}) \rightarrow \phi\ rather\ than\ \psi?$$
>
> *In which* ϕ *is the fact and* ψ *is the foil. This assumes that* ϕ *is true in the situation* $(M, \overline{u})$*, while* ψ *is not.*

In this notation for a bank loan task counterfactual explanation should answer the question “Why was the loan denied for this user rather than given?”. The explanation can be given in this way: “The loan would be granted if income increases by $5000.”

Another type of explanation which can be powerful for use cases is contrastive explanation (since almost all explanations are contrastive by nature it creates confusion in literature, so in Tim Miller notation this is called bi-factual). It compares two situations, where first is the current instance with prediction and another some hypothetical or historical observation. The question is why for this situation the prediction was one, while in some other situation it was different.

$$Why\ (M, \overline{u}) \rightarrow \phi\ but\ (M', \overline{u'}) \rightarrow \psi?$$

> *In which the* $(M, \overline{u})$ *and* $(M', \overline{u'})$ *are two different situations that might include two different models,* ϕ *is the fact and* ψ *is the surrogate. This assumes that* ϕ *is true in the situation* $(M, \overline{u})$*, while* ψ *is not.*

In a bank loan example it might be “Why was the loan given to mister X, while for mister Y the loan was denied?”. A suitable explanation for such a situation might be “The loan was given to mister X because his income is $5000 bigger than mister Y’s”.

Those explanations can be further categorised into property-contrast (Why does an object have property P, rather than property Q?), object-contrast (Why does object a have property P, while object b has property Q?) and time-contrast (T-contrast: Why does object a have property P at time t, but property Q at time t'?) explanations which might have different relevance for each use-case.

Many methods were designed to find constructive intervention to the input that have a meaningful impact on the model decision and lead to counterfactual change. Literature divides them into three main categories: independence-based, dependence-based and causality-based approaches [5].

Independence-based methods assume feature independence and may use evolutionary algorithms or combinatorial solvers in counterfactual search [6]. Such methods allow to add constraints to a search space and generate numbers of counterfactuals, however, it does not take into account feature correlations. To get reasonable explanations from such methods the number of data instances should be big enough to cover all directions in a search space.

Causality-based approaches take into account knowledge of the system and require either causal structural equations or causal graph. Generated counterfactuals are aligned with expert knowledge and can lead to desirable minimum-cost change [7]. However, practically it is difficult to obtain a true causal graph which is the main limiting factor.

Dependency-based approaches can be an intermediate solution between strong independence assumption and knowing true causal graph. The main idea is to encode various search and correlation constraints with variational autoencoders. An example of such a method can be CLUE that in addition takes into account the classifier's uncertainty [8].

1.2 Saliency measures related to human biases and constraints formalisation

People rarely expect an explanation that has an actual and complete chain of causes for the event. To identify causally relevant variables for the explanation we need to amplify some features with saliency measures. Saliency measures in counterfactual search meant to find causally relevant features, measure their importance and contributions of respective values to outcome. Selecting the cause of explanation hides various cognitive biases and hidden chains. Social science, philosophy and psychology suggest that there are multiple human biases and since they are domain specific [9], [2], in this section we address only those biases and constraints that might be relevant for TRUST-AI use-cases.

At this stage we can divide constraints into *user-specified* and *inherited cognitive constraints*.

Into user-specified constraints with input feature space $\{X_1, X_2, \dots, X_n\}$, where n - number of input features used in a model:

- List of actionable features: $\{X_1, X_2, \dots, X_k\}$, where $k \leqslant n$
- Feature values ranges (by default limited by min and max of feature values seen in dataset)
- Feature causal graph or set of rules
- Feature correlation
- Feature saliency (promotes more relevant features as cause)

Apart from user-specified constraints there might be inherited cognitive biases that the user is not consciously aware about. For revealing such biases we want to understand deeper why

users expect a certain prediction. There can be different reasons why some prediction is expected over other:

- It is normal behaviour for some other class, but the predicted class is different.
- It is its own historical experience (the user just saw other predictions for such settings).
- It is prior expert knowledge (known rule in the field).
- This outcome requires more responsible action and should be better justified.
- There is high uncertainty about this prediction.

Depending on the reason, the expected explanation follows a different structure and should appeal for different causes. Overall explanations for predictions that contradict users' expectations are mostly looking for exceptionable events (unusual time, unusual event, feature is far from data distribution etc.). The explanation is more appealing when it looks for controllable actionable features with the highest recency. The explanation is less satisfactory if it looks for changing physical properties, changing probabilities of events that happened or slight change of continuous variables (instead of possible categorical change). For instance, the explanation “The car crash happened because a tree fell on the road” is more appealing than “The car crash happened because car speed was 50 km/hour instead of 45”. However both might be valid counterfactual changes that lead to changing prediction output: car crash didn’t happen. We do not want to change the probability of a car being there, even though if the car was not on the road the car crash did not happen as well. As well as physical property, if a car just flies over a tree.

Therefore, explanations are selected in a biassed manner. The most likely explanation is not always the best explanation, while humans select one or two causes from an infinite number of causes. A good explanation should be relevant. It should provide high quality information of the right quantity that is relevant to conversation and in a good manner. When we explain something to someone, we assume a mental model of the explainee and explain only unknown information that will be relevant to such a model.

Let’s discuss some of the biases that we think are the closest for our use-cases:

- One of the key roles of explanations is generalisation. The user that receives an explanation unifies patterns which promotes the discovery of generalisations [10]. While from one side generalisation increases user understanding and trust in the system, on the other side it can lead to **overgeneralization**. When users generalise from previously seen explanations, they adjust their expectations to that experience. This can affect their assumptions of the system even when the object is different.
- With every explanation the new contextual information is presented which pushes previously important causes to background, such bias called **backgrounding**.
- People tend to explain events with abnormal causes even if other causes have bigger impact - **abnormality**.
- More recent events are more important than more distal events - **recency**.
- **Controllability** is related to responsibility, participants select intentional events as a cause, because it is possible to undo controllable events over uncontrollable.
- **Robustness** is a criterion for explanation selection. A cause is considered robust if the effect would still occur if conditions become somewhat different.

1.3 Counterfactual search space formalisation

Let D be the data set consisting of N input data points, $D = \{(x_1, y_1), \cdots, (x_N, y_N)\}$. We denote f as a trained predictor function that maps input space to output space.
With the desired output y the search for one counterfactual explanation can be formulated in the following way:

$$c = \arg\min_{x'} yloss(f(x'), y) + \lambda dist(x', x)$$

Where $yloss(f(x'), y)$ is the loss measure between counterfactual output $f(x')$ and desired output y. $dist(x', x)$ is the distance measure between counterfactual input point x' and the original input x. The hyperparameter $\lambda \in [0, 1]$ is a weight parameter. A high λ means it is more important to have counterfactual input close to the original input. A low λ means it is more important to find counterfactual point x' that gives output $f(x')$ closer to desired output y.
In this simple formulation we assumed feature independence.
For ***dependence-based*** methods we assume that factual input x is generated by a generative model g such that: $x = g(z)$, where z are latent codes. Thus, in counterfactual search we would be looking for alternative input x' that depend on latent space z that limits the search space [5].

The initial formalisation searches only for one counterfactual and does not include cognitive biases. It is important to search for multiple counterfactuals, since it is likely that the closest counterfactual does not comply with user constraints. Additionally, more counterfactuals give a bigger overview of a model behaviour. Thus, we aim to output several counterfactuals, which form a better overview of how the system works. Some counterfactuals can be later filtered out as an extra step if they do not comply with user constraints. We added saliency measures to initial objective function to promote causally relevant human-like explanations. The final formalisation of counterfactual search is the following:

$$C = \arg\min_{c_1, \cdots, c_k} \frac{1}{k} \sum_{i=1}^{k} yloss(f(c_i), y) + \lambda_1 \sum_{i=1}^{k} dist(c_i, \mathbf{x}) -$$

$$-\lambda_2 \cdot diversity(c_1, \cdots, c_k) - \lambda_3 \sum_{i=1}^{k} L(\mathbf{x}|c_i) + \lambda_4 \sum_{i=1}^{k} \frac{|\delta f(c_i)|}{\delta c_i}$$

All terms are weighted with hyperparameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and can be set to 0 if constraint is not applicable. We explain the meaning and implementation of each of these terms and how they contribute to selecting the most human-like cause separately.

<u>First term: loss function</u>

$$min_{c_1, \cdots, c_k} \frac{1}{k} \sum_{i=1}^{k} yloss(f(c_i), y)$$

The choice of loss function depends on the use-case, for classification problem hinge-loss pr cross-entropy can be applied.
Hinge loss:

$$hinge_{yloss} = \max(0, 1 - z \cdot logit(f(c)))$$

Where z is -1 when $y = 0$ and 1 when $y = 1$ and $logit(f(c))$ is the unscaled output from the model.
Cross-entropy:

$$crossentropy_{yloss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log f(c)_i$$

Where N is output size, y_i is desired output for a class with 0 and 1 values and $f(c)_i$ respective counterfactual output.

For regression mean squared error can be applied:

$$MSE_{yloss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i - f(c)_i)^2$$

Where N is output size, y_i is desired output for a class of a real value and $f(c)_i$ respective counterfactual output.

<u>Second term: distance function between input and counterfactual</u>

$$\min_{c_1,\cdots,c_k} \lambda_1 \sum_{i=1}^{k} dist(c_i, \mathbf{x})$$

Choice of distance function is crucial. Since not all features are equally changeable, we might want to apply different distance measures for some features. This term is able to incorporate ***recency*** bias to weight more recent samples and features more over others. On the other hand, feature weight can be increased for intentional features to promote **controllability**. There is also a natural division between continuous and categorical features which would have different properties.
Continuous features have different ranges. Therefore, it is important to normalise feature scale for instance with feature-wise distance dividing by the median absolute deviation (MAD) as suggested in [6]. Thus, the final distance measure would be feature-wise l_1 distance between the counterfactual example and the original input.

$$dist_{cont}(c, x) = \frac{1}{d_{cont}}\sum_{p=1}^{d_{cont}} \frac{|c^p - x^p|}{MAD_p}$$

Where d_{cont} is the number of continuous variables and MAD_p is the median absolute deviation for the p-continuous variable. This distance can be further weighted by feature importance weights to promote recency in causes. Alternatively, Mahalanobis distance can be used.
Categorical features
For categorical features, however, it is unclear how to define a notion of distance. It must be relative for each feature, possibly relative with respect to the difficulty of changing and maybe to the feature importance as well. We can apply a distance measure that corresponds to a difficulty of changing a feature which assigns 1 if the counterfactual has this value different from original input and 0 if original input and counterfactual are equal. However, it can be weighted further by the number of categories and difficulty scales:

$$dist_{cat}(c, x) = \frac{1}{d_{cat}} \sum_{p=1}^{d_{cat}} I(c^p \neq x^p)$$

Where d_{cat} is the number of categorical variables, the sum might be weighted by the cost of changing each feature separately.

Third term: Diversity within counterfactuals

$$\min_{c_1,\cdots,c_k} -\lambda_2 diversity(c_1, \cdots, c_k)$$

Diversity term promotes a variety of output counterfactuals that helps to convince the user with different arguments and explore the search space in more than one direction. The function was introduced in [6] and use the determinant of the kernel matrix that is based on a distance between two counterfactuals:

$$diversity(c_1, \cdots, c_k) = \det(K)$$

Where $K_{i,j} = \frac{1}{1+dist(c_i, c_j)}$ and $dist(c_i, c_j)$ denotes a distance metric between the two counterfactual examples.

Fourth term: Penalty for deviating from sample distribution

The distance between desired class probability distribution and counterfactual outcome should be minimised to ensure that counterfactual input belongs to that class distribution.

$$\min_{c_1,\cdots,c_k} -\lambda_3 \sum_{i=1}^{k} L(\mathbf{x}|c_i)$$

where k is the number of output counterfactuals, $L(\mathbf{x}|c_i)$ - is a likelihood that a counterfactual c_i belongs to desired class distribution.

Fifth term: Robustness of counterfactual

Counterfactual explanation should be **robust**, meaning that a small change in input should still lead to the same class prediction. Mathematically, $f(c + \epsilon) = f(c)$, where f is a model.

$$\min_{c_1,\cdots,c_k} \lambda_4 \sum_{i=1}^{k} \frac{|\delta f(c_i)|}{\delta c_i}$$

Where k is the number of output counterfactuals.

The communication with a user requires further filtering of causes, where we follow **abnormality** to select what was abnormal in the input situation and which cause flip the situation to be normal from a human perspective. All terms are weighted with hyperparameters. Their importance can vary for each use-case.

User can perform multiple actions in counterfactual search:

- Select distance metric for different feature types (continuous, categorical)
- Change the number of output counterfactuals
- Weight feature list with feature importance list
- Select actionable feature list and values ranges

- Put other hyperparameters weight for each term

1.4 Evaluation of counterfactual explanation

Counterfactual explanations at the end will be evaluated by the end users, however, we try to measure metrics that can measure how meaningful counterfactuals are during the intermediate phase.

Validity: The counterfactual should be valid, meaning it leads to a desired counterfactual outcome.

$$Validity = \frac{\sum_{i=0}^{k} f(c_i) = y}{k} \cdot 100\%$$

Where k is number of counterfactuals, y desired output class. Validity of 100% means all counterfactuals lead to desired outcome change and 0% means none of counterfactual outputs desired class.

Cost: The distance between initial output and counterfactual defines the cost of counterfactual.

$$Cost = dist(c, x)$$

Sparsity: While cost refers to the change of output values, sparsity quantifies how many features should be changed within counterfactual.

$$Sparsity = \sum_{l=1}^{d} 1_{c^l \neq x^l}$$

Where d is number of features and x^d denotes d-th feature value.

Diversity: The distance between counterfactuals can be measured as counterfactual diversity.

$$Diversity = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} dist(c_i, c_j)$$

Where k is the number of counterfactuals and $dist(c_i, c_j)$ is calculated according to categorical or continuous distances types.

Constraint violation: This metric counts how many times counterfactual violates user-defined constraints.

Data support: We want our counterfactual to be close to the data that support the desired class. This metric measures how neighbourhood points around counterfactual instances are classified.

$$Support = 1 - \frac{1}{N} \sum_{i \in kNN(c)} |f(c) - f(x_i)|$$

Where kNN denotes the k-nearest neighbours and n is the number of neighbourhood points around the counterfactual instance.

Average time: Finally, the time of counterfactual search to converge for desired output should be measured.

After a user receives an explanation we can measure *human-based heuristics* with a 10-point Likert Explanation Satisfaction Scale [11]. We aim to measure such criteria:

- **Simplicity:** represents the number of causal mechanisms.
- **Recency:** how counterfactual relies on the most recent events.
- **Effectiveness:** if counterfactual helps to make a better decision (informative).
- **Trust:** how counterfactual increases confidence in a system and not contradict with domain knowledge.
- **Unbiased:** counterfactual does not rely on incorrect features.
- **Coherency:** each element in explanation positively constraints other ones.

2. Causal inference

2.1 Introduction

Judea Pearl brought the causal inference to the scientific community's attention in a series of papers [12], [13] and books [14], [15].
The causal inference is fundamentally different from the problem of prediction or regression the machine learning specialist is familiar with. If we want to change a system by acting on the parts of a system, we need to understand how the parts work together; that is, we need to understand the causal structure of the domain.
We can start with the observation that a feature can be a good predictor in a dataset, but this does not necessarily imply that it has a causal effect on the outcome. For example, a man's shoe size is an excellent height predictor. However, the answer to the question "What should we do to increase the height of a man?" could not possibly be buying a larger shoe. Estimating a causal effect requires knowledge of the domain and the relevant mechanisms operating. In other words, it requires expert knowledge that is not encoded in data.
According to [16], data science has three tasks demanding different methods and philosophies.

1. The first task belongs to the descriptive statistics and answers questions like
    a. What happened?
    b. Who is affected?
    c. Do the instances have the X [property] also have Y [property]?
2. The second task is predictive and aims to answer questions like the following:
    a. What will happen?,
    b. Who will be affected?
    c. Do people with X are more likely to have Y?
3. The third task, called causal inference, answers questions like
    a. What will happen if?
    b. Why are the instances affected?
    c. How does Y change when I change X?

To better understand how causality relates to cognitive abilities, it is helpful to revisit Pearl's ladder of causation [Table 1].

Level	Activity	Typical Questions	Examples
Association	Seeing	What is? How would seeing X change my belief in Y?	What a symptom tells me about a disease? What does a survey tell us about the election results?
Intervention	Intervening	What if? What if I do X?	What if I take aspirin? Will my headache be cured? What if we ban cigarettes?

Counterfactuals	Imagining	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking in the past two years?

Table 1. Judea Pear's ladder of causation, with typical questions at each level

At the first level of the hierarchy (**Association**), simple questions can be answered by computing a conditional probability measure.
If our data is sampled from a joint probability distribution, $p(x, y, z, \ldots)$ then the observational behaviour of the variable y conditioned on x is given by the following formula:

1. $p(y|x) = \frac{p(x,y)}{p(y)}$

For example, the first question in the "Examples" column is answered by computing the degree of association between a certain symptom and the disease. All machine learning techniques, including the very successful deep learning methods and the genetic programming algorithms used in this project, answer questions at the first rug of the ladder of causation.
The second level of the ladder encompasses questions about the interventions. We are interested in estimating the outcome of the user actions. The interventional behavior $p(y|do(x))$ describes the distribution Y when the user intervenes in the data generation process by forcing the variable X to take the value x.
On the third and last level are the hardest queries, dealing with counterfactual scenarios. If we have a good causal model, we could answer, in principle, counterfactual questions of the type: Given that my customer bought a software package at the price t, would s(he) buy it if I would double the price?

2.2 Randomization control experiment

The randomization control experiment is a standard paradigm for finding the causal effects. From a population of individuals, we randomly select two groups of individuals alike in all controllable respects (as seen in figure 2.1).
To one of the two groups, we perform an intervention, in literature called a treatment (T=1), and then estimate its causal effect. To find the causal effect of treatment T on the outcome Y, we have to consider two worlds, called the real world and the counterfactual world.
In the real world, the treatment T is performed (T=1) and the effect of the treatment on outcome Y is observed. In the counterfactual world, the treatment is not performed (T=0).
The magnitude of the causal effect is the difference between Y values attained in the real world versus the counterfactual world.

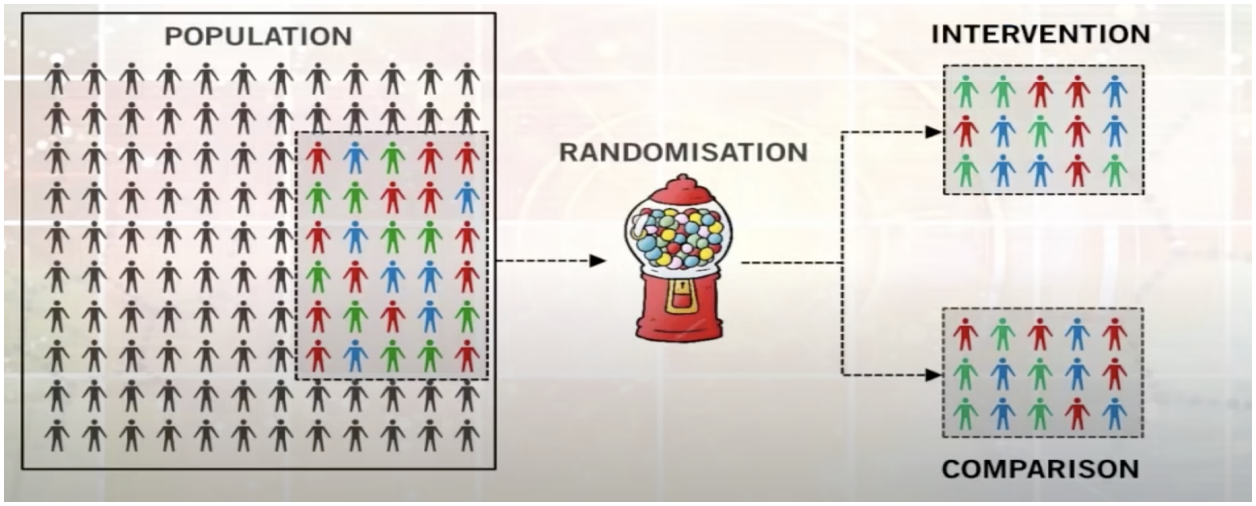

Figure 2.1. A randomization control experiment. The treatment T is performed for the upper group. The causal effect is gauged by comparing with the lower group.

Unfortunately, a randomization control experiment cannot always be performed for ethical reasons. Imagine that we want to determine the effect of smoking (treatment) on health (outcome). We cannot force the non-smoking individuals in the intervention group to start smoking.

Fortunately, there are ways to estimate a causal effect without a randomization control experiment.

For example, if we can observe every single variable that affects the outcome, we can compare the individuals with the same values for the controlled variables but different values for the treatment variable. Though this method can be applied in some instances, it is doubtful that many individuals having the above properties can be found.

The second way to estimate the causal effects is with the causal calculus that uses the domain knowledge represented as a causal graph.

2.3 Causal graphs

The causal models can be thought of as mechanisms by which the data is generated. The causal graphs allow specification of the domain knowledge. In particular, the causal graphs encode the expert's prior knowledge of the data generation mechanism and assumptions about plausible causal mechanisms. In this section, we present the main modelling elements of a causal graph that allows the specification of the knowledge for the three use cases.

A causal graph is a directed acyclic graph made up of two kinds of elements.

- **Nodes** represent variables or features in the world or system we are modelling. Each node represents something that is potentially observable, measurable, or knowable about a system.
- **Edges** connect nodes. Each edge represents a mechanism or causal relationship related to the values of the connected nodes. Edges are directed to indicate the flow of causal influence.

The causal graphs encode the following intuitions:

1. The assumptions are encoded by the missing edges and the direction of the edges. If there is no edge between two nodes, this means that the variables represented at nodes do not influence each other; they are statistically independent.
2. The relationships represent independent causal mechanisms.
3. The causal graphs cannot be learned from the data alone. There are multiple causal mechanisms that can be fitted to single data distribution. We need an expert or a collection of assumptions validated by an expert to specify the potential mechanism that can be causing the observations.

The following building blocks are crucial in the specification of causal graphs because any causal graph can be built by a combination of them.

1. A **chain (**A → B → C).
    a. A is causing B, which in turn is causing C. B can be thought of as a mediator that transmits the effect of A to C. conditioning on B. A and C are independent.

A familiar example is (Fire → Smoke → Alarm)

1. **A fork** (A ← B → C).
    a. B is a common cause of A and C, also called a confounder for A and C. A and C are statistically correlated, even though there is no causal link between them. For example, "ice cream consumption" and "shark attacks" are correlated, though none is the cause of the other. The common cause is "summer," where the heat increases both ice-cream consumption and shark attacks.
2. A **collider** (A→B←C). B is caused by both A and C. For example, the "musical talent" and the "child of a veteran" are marginally independent variables, but both contribute to "winning a scholarship."

Chains, forks, and colliders are causal structures that explain the observed associations. The observed data associations are either causal associations or noncausal associations. Noncausal associations can arise through forks and conditioning on a collision node.

The following causal graph [figure 2.2] is the result of a study [17] that aims to assess the association between the risk of T2DM [type 2 diabetes mellitus] and educational level across eight Western European countries. The bold arrows denote consistent causal relations. According to it, the causal factors for diabetes are Age, BMI (body mass index), Sex, and Diet. Please also note that the correlation between BMI and Smoking status can be explained by the educational level, the common cause of both variables (see the fork building block above).

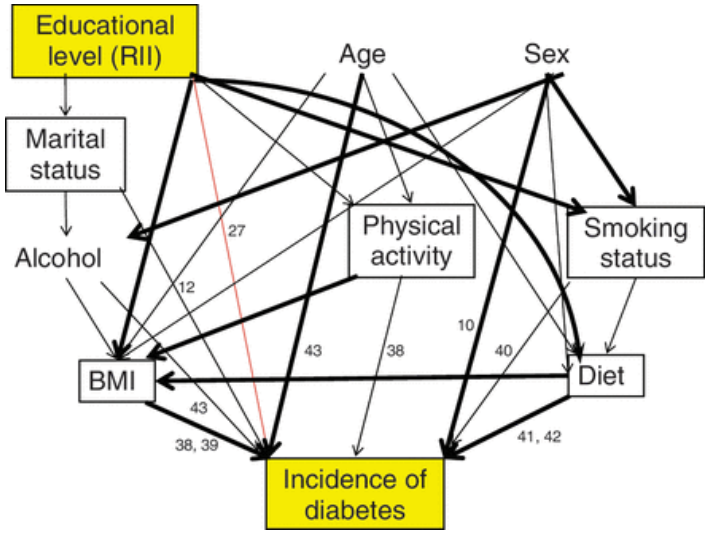

Figure 2.2. A causal graph of social factors that influence T2DM [type 2 diabetes mellitus]

2.4 Causal inference

The causal inference process starts with a research question. For example, if we take the example in the causal graph in figure 2, we can pose the following research questions:

1. What is the effect of BMI on the incidence of diabetes
2. What is the impact of diet on the incidence of diabetes

Please notice that the BMI and the diet are the only features we can perform interventions on. The other causally related features with the outcome, sex, and age are not actionable.
The causal calculus allows estimating the effect of interventions on the outcome. Judea Pearl's do-calculus for causal inference has been implemented in the software library DoWhy [18]. DoWhy is designed to define the critical assumptions for performing causal inference using a four-step procedure to model and validate the causal assumptions.

1. **Modelling**. The causal graph encompassing the causal assumptions is built. In the causal graph, the edges incident to the treatment T are removed. The probability of Y (outcome) given do(T) (the intervention) should be computed using the causal calculus.
2. **Identification**. In this stage, the causal effects are identified using the properties of the causal graph. The problem is how to represent the quantities in the distribution P(T|do(Y)) using the observational data. The solution is to adjust for the other feature influence and simulate a randomised experiment.
3. **Estimation**. In this stage, the causal effect is computed with the observed data to calculate the impact of the intervention. A conditional probability is estimated by keeping the confounders constant.
4. **Robustness**. In the last stage, the robustness of the estimate is validated. A series of conditional independent testing and integration tests are implemented to evaluate if the modelling is correct.

3. Use case specific causal graphs and constraints

This section collects expert domain knowledge about each use-case. There are three methods by which the domain knowledge for the three use cases studied in this project and for other datasets can be elicited:

1. The experts in the domain manually build the causal graphs. In the modelling process, the feature correlation calculated for observational data is the information used by the experts.
2. The relevant literature is mined using NLP techniques, and an expert validates the domain knowledge.
3. A non-expert consults the literature to find relevant causal graphs, as in the case with the incidence of diabetes above.

3.1 Energy case

This section presents the current data collected for the energy case. After a general description of the data attributes, the pertinent biases and constraints for the counterfactual search are shown.

Currently, the energy data has 2294 instances and 12 features. The data is a time series comprising the electricity consumption history, the current day, the indoor and outdoor temperature, and the wind velocity.

The data [figure 3.1] is split by default in training, validation, and test data with the following proportions:

1. 40 percent of the data is used for training
2. 10 percent of the data is used for validation
3. 50 percent of the data is used for testing

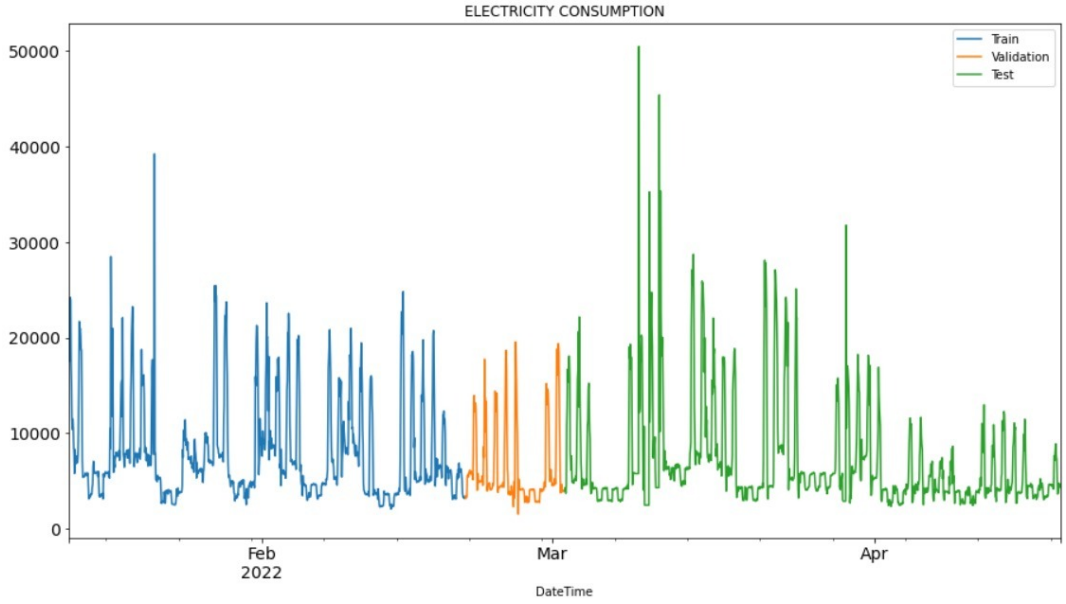

Figure 3.1. Electricity consumption data with the training/validation/test set

3.1.1 Counterfactual search biases and constraints

In table 3.1, some attributes of the features are described. We are interested in the feature's actionability, that is, if it can be used in the counterfactual search.

Feature_name	Type	Actionable*	Range	Mean, std	Constraints
Electricity, hourly, KWH)	continuous	NO	[1.54, 50.4]	[7.33, 5.28]	
Indoor temp [C]	continuous	YES	[11.75, 26.19]	[18.73, 2.75]	*HVAC system constraints
outdoor temp [C]	continuous	NO	[2.12, 27.91]	[12,29, 3.94]	
Wind [km/h]	continuous	NO	[0, 51.83]	[17.38, 12.34]	

**HVAC stays for Heating, Venting, and Air Conditioning*

Table 3.1. Some attributes of the features for the energy case.

Other (actionable/ behavioural) features not included in the current setup, but that will be added in the future when more data will be collected are:

- HVAC Operation during nonwork hours: number
- Lighting Operation during nonwork hours: number
- Devices Operation during nonwork hours: number
- High solar gain: (0: no instance, 1: few instances, 2: many instances)
- Open Window: (0: no instance, 1: few instances, 2: many instances)
- Lighting during daylight conditions (0: no instance, 1: few instances, 2: many instances)

In figure 3.1, the correlation of the features is presented.

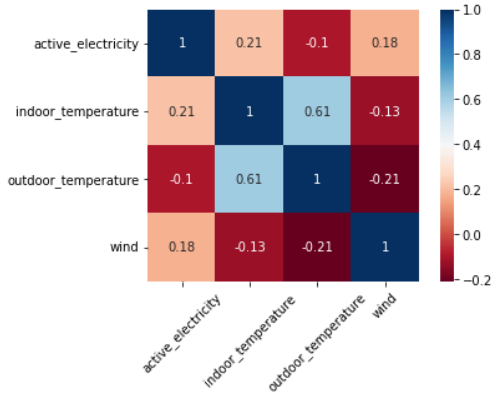

Figure 3.1. The feature correlation for the electricity case

From the human biases discussed in a previous section, recency is true by design in the energy case because only the recent historical data is included in the time series modelling. The other human biases do not apply in this case.

3.1.2 Causal graph

In the energy case, we are interested in the following research questions:

1. How is the **indoor temperature** influencing **energy consumption**?
2. How is the **wind** influencing **energy consumption**?
3. How does the **pricing of energy slots** influence **user behaviour**?

The Apintech energy experts have drawn the causal graph in figure 3.3. The data for answering the third question has not been collected yet, but it will be available next year. The third question is the critical question to be answered for the energy case.
The causal graph for the energy case is based on the following assumptions:

1. In summer, the energy consumption decreases because the indoor temperature increases. In winter, the energy consumption increases because the indoor temperature decreases. According to [19] the energy use decreases with rising temperatures due to reduced demand for energy for heating purposes, and the speed of that decrease declines with increasing temperature levels.
2. The relationship of wind with energy consumption is more complicated. According to [20], four mechanisms build heat exchange with the surrounding environment influencing the overall energy consumption: air infiltration and exfiltration, surface heat transmission; air flows affect the effectiveness of air-conditioning, and wind affects human thermal comfort.

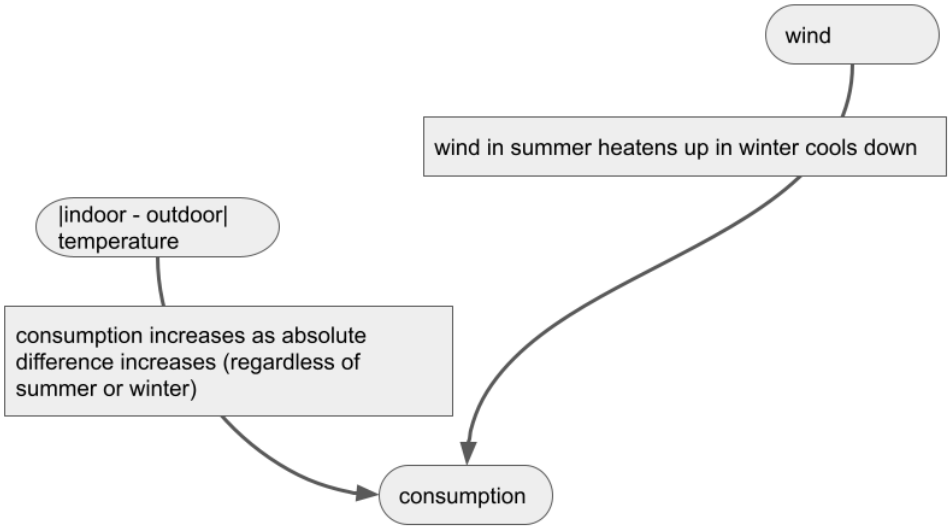

Figure 3.2. A preliminary causal graph for the energy case showing the influence of the indoor|outdoor temperature and wind on the building level energy consumption.

3.2 Healthcare case

3.2.1 Extracting knowledge for healthcare

In the healthcare case, we use the DisGeNET [21] database that integrates human gene-disease association information from various online repositories with data mined from scientific literature corpora. In particular, the database is regularly updated through data mining techniques with different relations between genes and diseases. The DisGeNET association type ontology gives the type of permissible association relations.
Currently, the DisGeNET database comprises 1,134,942 gene-disease associations (GDA), between 21,671 genes and 30,170 diseases, disorders, traits, and clinical or abnormal human phenotypes.
The strength between a gene and a disease is quantified by the GDA score that indicates the popularity of gene-disease association across all data sources, giving higher weight to curated gene-disease repositories.
Furthermore, the GDA score is supplemented with another measure called Evidence Level. It measures the strength of evidence of a gene-disease relationship that has a corresponding qualitative classification given by the following labels: Definitive, Strong, Moderate, Limited, and Disputed [22]
We have filtered the records in the database for the paraganglioma case based on the GDA score and the evidence level. Here are the results with a short explanation:

- The total number of genes associated with paraganglioma is 166. Genes with a solid evidence level (SDHD, VHL, EPAS1, TIMEM127, EGLN1, TP53)
- If we select the genes with a GDA score greater than 0.5, we obtain the following candidate genes
    - SDHD (Succinate Dehydrogenase Complex Subunit D).
    - SDHB (one of four subunits of the succinate dehydrogenase)
    - SDHC (one of four subunits of the succinate dehydrogenase). According to MedlinePlus, the SDHD, SDHB, and SDHC genes provide instructions for making one of four succinate dehydrogenase (SDH) enzyme subunits. The SDH enzyme plays a critical role in mitochondria, which are structures inside cells that convert the energy from food into a form that cells can use.
    - VHL is a gene that makes a protein that helps control cell growth, cell division, and other essential cell functions. Mutated forms of the VHL gene may increase the growth of cells, including abnormal cells. This function can play a role in paraganglioma tumor growth.

We have extracted a set of sentences from PubMed where the paraganglioma and a gene are co-occurring. Two examples of sentences supporting a gene - paraganglioma association are given below:

1. Genetic studies have shown that familial paragangliomas are associated with germline mutation of succinate dehydrogenase subunits SDHD on 11q23.

2. PCC/PGL are associated with a variety of hereditary syndromes, comprising genetic alterations in RET, NF1, VHL, and SDHx genes, the last 2 being involved in regulating the hypoxia pathway.

A medical doctor should validate the information extracted from the DisGeNET database. We discussed the information extracted to date with the medical doctor Jeroen Jansen (CWI), who confirmed that most of it is accurate. Doctor Jansen also said that he was aware of this information.

3.2.2 Feature overview

Current data consist of patient age, tumour volume, medicine use, BMI, MR images, biomedical screenings and treatment information. However, out of all features the model uses age and tumour volume data at this stage.

Feature name	Type	Number of instances	Actionable	Mean, std	Constraints
Age 1 (years)	double	77	No	44.79,11.83	Age must be bigger than 0
Age 2 (years)	double	77	No	48.44,11.75	Age must be bigger than 0
Age 3 (years)	double	77	No	51.67,11.95	Age must be bigger than 0
Volume 1 (ml)	double	77	No	10.57,15.50	Volume must be bigger than 0 and smaller than 1500
Volume 2 (ml)	double	77	No	13.74,20.23	Volume must be bigger than 0 and smaller than 1500
Volume 3 (ml)	double	77	No	18.58,30.92	Volume must be bigger than 0 and smaller than 1500

Table 3.2. Some attributes of the features for the health case.

It is important to notice that all current features are not actionable. It is possible to generate counterfactual explanations for informational purposes, but we can't suggest them as an action to the user.

Figure 3.3 shows the correlation for features that are used in the model.

	age_1	age_2	age_3	volume_1	volume_2	volume_3
age_1	1.000000	0.992102	0.986132	0.144282	0.099925	0.104221
age_2	0.992102	1.000000	0.996573	0.135389	0.089609	0.090891
age_3	0.986132	0.996573	1.000000	0.132961	0.086921	0.094935
volume_1	0.144282	0.135389	0.132961	1.000000	0.971805	0.891749
volume_2	0.099925	0.089609	0.086921	0.971805	1.000000	0.955892
volume_3	0.104221	0.090891	0.094935	0.891749	0.955892	1.000000

Figure 3.3. Feature correlation for age and tumour volume features.

3.2.3 Causal graph

The initial version of the causal graph consists of features that are used in the current model. The age of one patient causally related to the next measurements. Similarly, previous measurements of patients affect the next tumour volume measurements. However, there is no causal link between the patient and tumour volume known to experts. Figure 3.4 illustrates these relations.

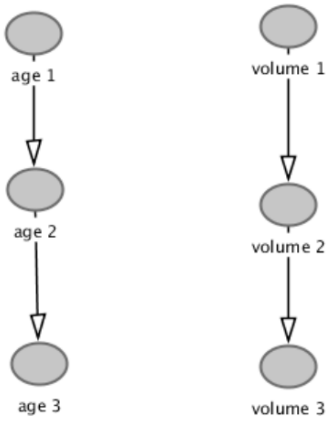

Figure 3.4. Causal graph for features that are used in the current model in the health case.

3.3 Online retail

This section reviews the expert knowledge for the current data collected for the retail case. After a general description of the data attributes, the pertinent biases and constraints for the counterfactual search are shown.

3.3.1 Feature overview

Currently, the training data has 3529 instances and 13 features. The output is binary with 0 if the time slot is chosen by the customer and 0 otherwise. In table 3.3, attributes of the features are described. We are interested in the feature's actionability, if it can be used in the counterfactual search.

Feature name	Meaning	Type	Actionable *	Range	Mean, std	Constraints
slotcost	The displayed time slot price	cont.	yes	[4.59; 10.18]	(6.21, 0.59)	Decision-maker could establish that min_price <= slotcost <= max_price
slot_start	Number of minutes since the order instant until the opening time of the slot	cont.	yes	[840; 22800]	(9873.17, 6149.87)	slot_start >= 0
exact_selection_customer_perc	The historical percentage of times that the customer selected a given time slot	cont.	yes	[0; 1]	(0.065, 0.142)	0 <= exact_selection_customer_perc <= 1
rank_cost	Considering all time slots presented to the customer, gives the percentile in which the time slot lies with respect to price, e.g., rank_cost = 12% means that the time slot is in the 12% cheapest slots offered	cont.	no (consequence of slotcost and remaining time slot offers)	[0; 1]	(0.529, 0.248)	0 <= rank_cost <= 1
median_cost	The median price of all time slots shown to the customer	cont.	no (consequence of slotcost and remaining time slot offers)	[5.75; 7.38]	(6.51, 0.28)	
partial_selection_customer_perc	The historical percentage of times that the customer selected a time slot that intersects the time slot under analysis with respect to time, e.g., If the customer had selected time slot ranging from 1pm to 3pm and the time slot under analysis was from 2pm to 4pm, the previous case would be taken into account for the percentage.	cont.	yes	[0; 1]	(0.116, 0.176)	0 <= partial_selection_customer_perc <= 1
expanding_avg_days_to_delivery	The average number of days between the moment of the order and the start of the time slot for previous time slot selections	cont.	yes	[0; 6]	(1.46, 0.64)	Expanding_avg_days_to_delivery >= 0 (and expanding_avg_days_to_delivery <= 6 for our model, since we discarded time slots 7 days ahead of the moment of the order)
days_since_first_purchase	The number of days since the customer last made a purchase using the retailer attended home delivery service	cont.	yes	[1; 363]	(162.99, 100.31)	Days_since_first_purchase >= 0
q1_cost	Considering the price distribution of the time slots displayed to the customer, this feature provides the first quartile.	cont.	no	[5.41; 6.90]	(5.99, 0.31)	

max_cost	The highest price among the prices displayed to the customer	cont.	yes	[6.31; 13.46]	(7.37, 0.94)	When varying max_cost, the price distribution of the time slot price panel presented to the customer will change. To test variations in max_cost, price features need to be updated
min_cost	The lowest price among the prices displayed to the customer	cont.	yes	[4.42; 6.32]	(5.28, 0.30)	When varying min_cost, the price distribution of the time slot price panel presented to the customer will change. To test variations in min_cost, price features need to be updated
slot_width	The range of a time slot in minutes, e.g., a slot that starts at 1pm and ends at 3pm as a slot_width of 2h or 120 min.	cont	yes	[120; 450]	(147.22, 33.57)	Decision-maker could establish that min_width <= slot_width <= max_width

Table 3.3. Feature overview of online retail data.

In figure 3.5, the correlation of the features is presented.

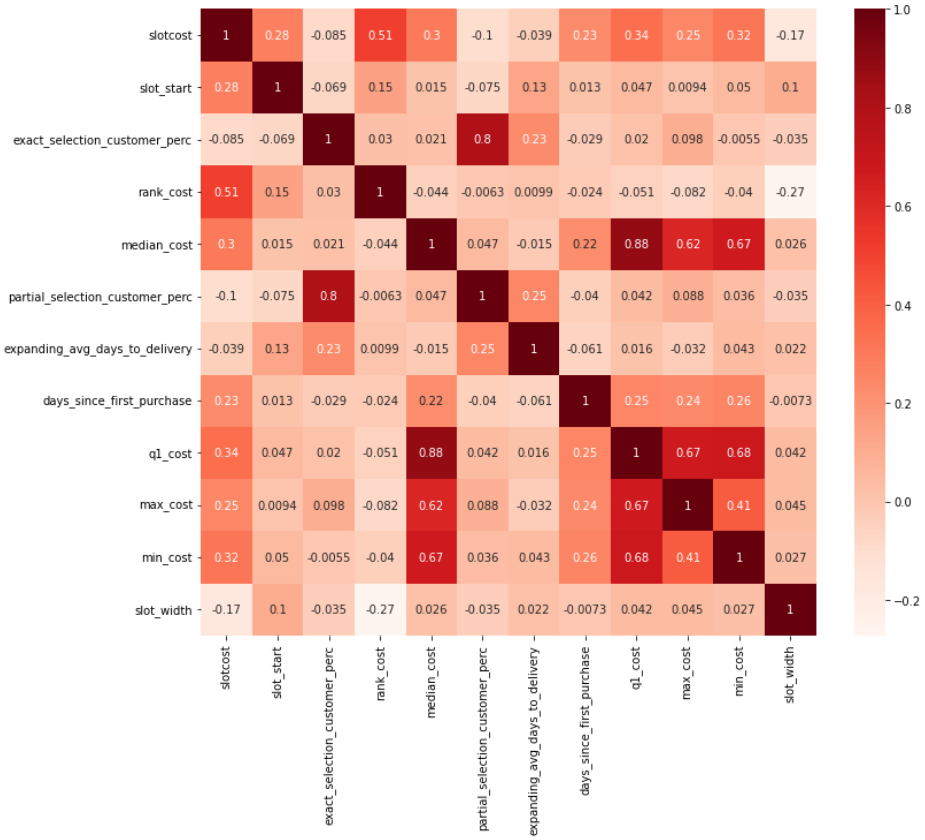

Figure 3.5. The feature correlation for the electricity case

3.3.2 Expert knowledge

In the literature [23] on customer preferences for attended home delivery services, the following three features are commonly known to be determining:

- Lead Time: customers tend to prefer slots that are closer in time (where the difference between slot start and the order time is lower).
- Availability: customers are willing to pay more if there is a high variety of available options to choose from.
- Range: Customers prefer narrower slots as these allow them to know more accurately the moment of delivery. For instance, choosing a time slot ranging from 1pm to 2pm provides a more accurate estimate of when I need to be home to receive the order, while a time slot lasting for the whole afternoon does not.

Regarding price, generally, when the price of a time slot is increased its selection probability decreases. However, there could be an encapsulated effect due to the retailer's current pricing policies. For those slots that are more popular, the retailer introduces mark-ups to collect higher revenues. Therefore, in the data, we could find the pattern that time slots with a higher price actually tend to be more preferred.

We treat the problem as a classification problem where we want to answer the following question:

- "Given a combination of customer and time slot, will the customer select the slot? And with which probability?

However, in the context of solving the retail use case, we want a model capable of determining such probability while taking into account the characteristics of competing time slots. Therefore, we introduce several statistical measures to characterise the price distribution of the offer presented to the customer. Therefore, we know that features rank_cost, median_cost, q1_cost, max_cost and min_cost all depend on slotcost.

Experts assessed several rules of how features relations should work. The notion of rules in the following:

↑/↓ feature name 1 → ↑/↓ feature name 2,

Where ↑ means that the increase of feature name 1 (or ↓ the decrease of feature name 1) should lead to ↑ increase of feature name 2 (or ↓ decrease of feature name 2).

For retail case the following set of rules were identified:

↑ slotcost → ↓ selection probability

↑ slot_start → ↓ selection probability

↑ exact_selection_customer_perc → ↑ selection probability

↑ slotcost → ↑ rank_cost (potentially)

↑ rank_cost → ↓ selection probability

↑ slotcost → ↑ median_cost (potentially)

↑ median_cost → ↑ selection probability (assuming the slot under analysis remains with the same price)

↑ partial_selection_customer_perc → ↑ selection probability

↑ expanding_avg_days_to_delivery (could indicate that customer does not value service speed and is more price sensitive)

↑ days_since_first_purchase (could indicate customer generally presents higher time slot choice probabilities as he/she is loyal to the retailer and its service)

↑ slotcost → ↑ q1_cost (potentially)

↑ q1_cost → ↑ selection probability (assuming the slot under analysis remains with the same price)

↑ slotcost → ↑ max_cost (potentially)

↑ max_cost → ↑ selection probability (assuming the slot under analysis remains with the same price)

↑ slotcost → ↑ min_cost (potentially)

↑ min_cost → ↑ selection probability (assuming the slot under analysis remains with the same price)

↓ slot_width → ↑ selection probability

The LTPlabs experts have drawn the causal graph in figure 3.6.

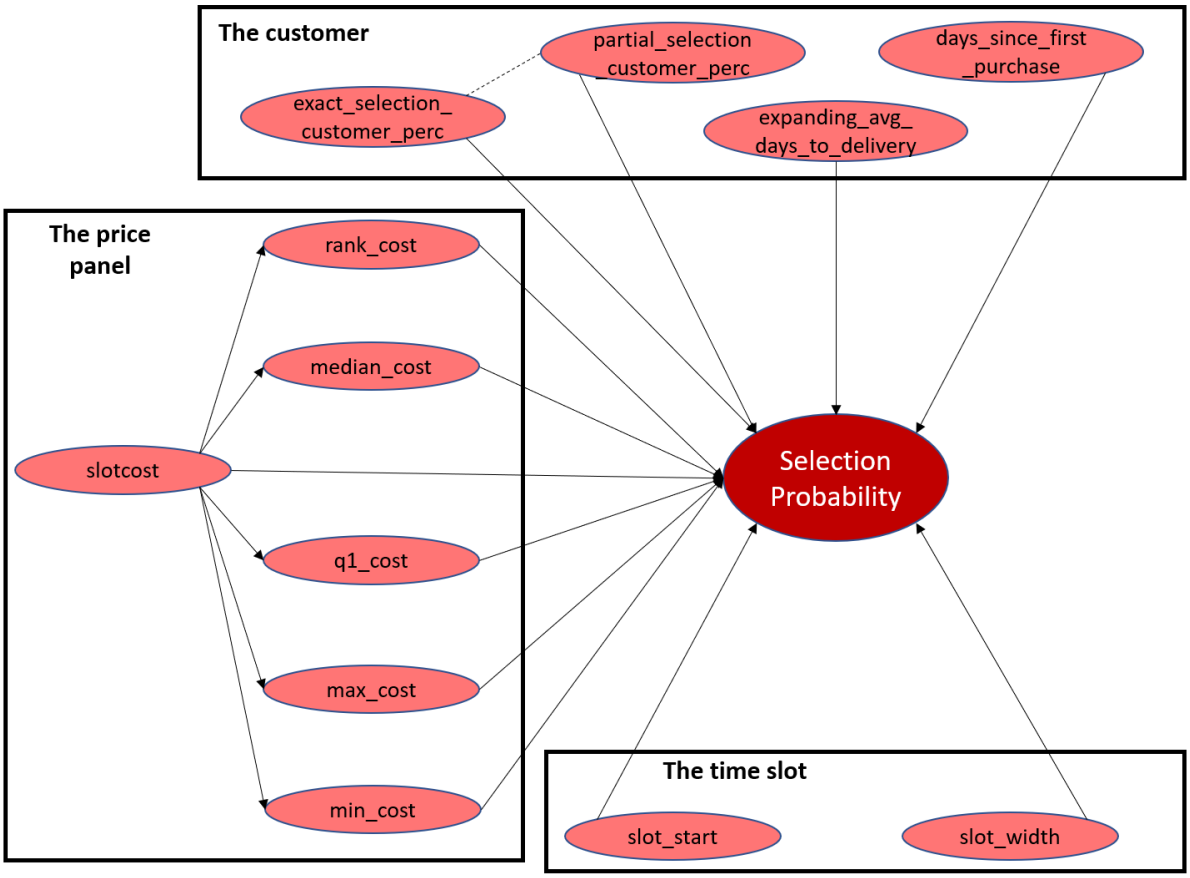

Figure 3.6. A preliminary causal graph for the retail case showing the influence on selection probability of the time slot.

4. Other methods that increase trust in the platform

This section lists other methods that increase user trust in the system, such as model interventions and feature importance. It also discusses how model uncertainty can help to create better counterfactual explanations and visualise the search space.

4.1 Promoting user dialogue with model interventions

Model interventions allow users to play around input feature values to see what would be the output in this scenario. The user should be able to query several scenarios of what happens if input would be different. Such an interactive way increases trust in the system and helps users to estimate model behaviour. The best learning and understanding form is an interactive dialogue. In addition to “What happen if” questions, we can suggest some counterfactuals that actually change the output. For example, if a user plays around one specific feature multiple times and receives the same model output, we can suggest running a counterfactual search for this feature so the output flips.

4.2 Visualising global and local feature importance for the model

Feature saliency or feature importance provides information about how important certain features are for the model. Feature importance is a common way to express how single features influence the model prediction. SHAP feature importance graph shows not only features sorted by importance, but data points distribution of each feature. During the questionnaire of preferred types and forms of explanations for the end users, all respondents (doctors, decision-makers, operational managers) marked feature importance graphs with contributions to outcome with higher score for understandability and effectiveness than symbolic expression graphs. In order to verify that the graph was correctly interpreted, we asked follow-up questions about how some concrete features affect model decision. In general, most questions were answered correctly, however, there was one question that was answered wrongly by 2 out of 3 respondents. Therefore, it might be that graph presentation should be simplified. An additional suggestion from one doctor was to explain the meaning of SHAP values.

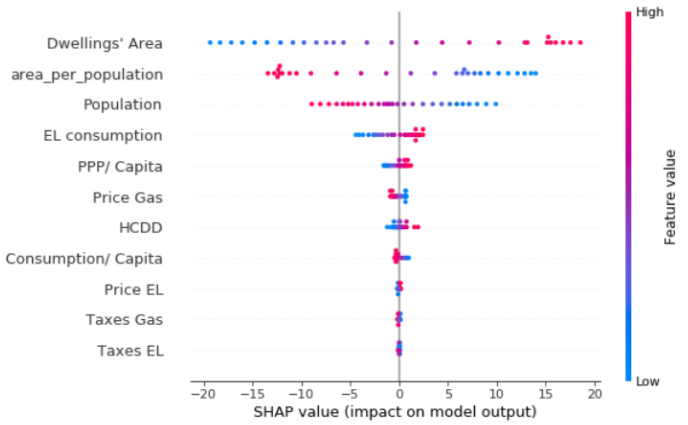

Figure 4.1. Example of global feature importance graph with contributions to outcome estimated on energy country sub-case.

One instance explanation can provide local feature importance. In the questionnaire the combination of textual explanation with local feature importance received the highest efficiency scores for retail customers.

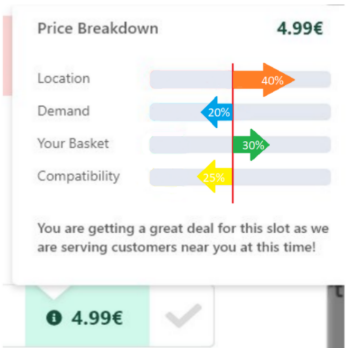

Figure 4.2. Example of local feature importance graph in combination with textual explanation.

4.3 Generating explanations for different uncertainty levels

Explanations are an important part of how we obtain the new information. They are needed for informing and supporting human decision making. When we give explanations to each other, we share many biases and adjust the explanation according to the informational goal. When we are making high risk decisions, we would like to be certain that the decision that is generated is reliable and convinces the user in decision reliability. Uncertainty estimation can increase trustworthiness of the machine learning system [8] and measure the reliability of the decision. Even if the model prediction is not certain in the outcome we sometimes still want to show that prediction in an informative way, highlighting abnormality of the feature values for such outcome. The algorithm of forming an explanation based on explanation uncertainty is shown in Figure 4.3.

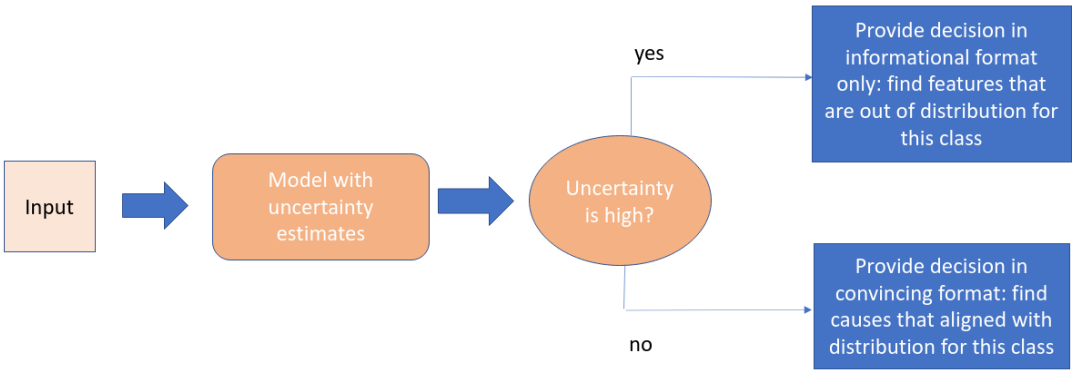

Figure 4.3. Explanation generation pipeline that includes decision of model uncertainty. The diagram shows how uncertainty of the model affects the goal of explanation.

For instance, if there is a diabetes classifier with two outcomes: "high-risk" diabetes class and "low-risk" diabetes class. In case of a big confidence decision we can explain the model

outcome using the most important features that lie within this class distribution: "The model is very certain that the patient belongs to a high-risk diabetes class, since he is over 50 years old and his glucose is 148.". Alternatively, if the classifier is not certain, we need to highlight which features contribute to uncertainty informing decision-maker for such abnormalities: "The model is not certain to predict this patient to a high-risk diabetes class, since the patient is 26 years old with 78 glucose but 31 BMI still point to high diabetes risk.". Note how words like "very certain", "not certain"-"since"-"but" construction make model prediction more human-friendly and increase trust in perception.

Furthermore, recent user studies confirm that proving counterfactual explanations for different confidence levels increase user trust in the system significantly [4]. In addition to that counterfactual space can be visualised using showing how confidence scores can change with respect to proposed counterfactual (Figure 4.4).

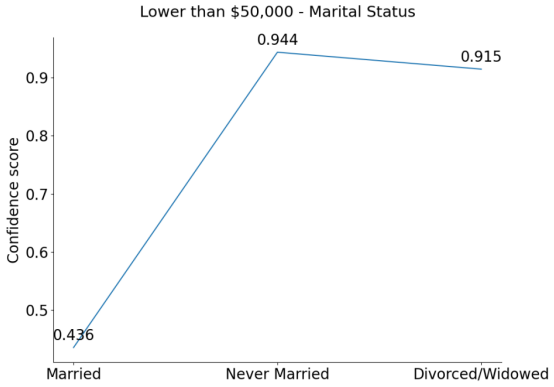

Figure 4.4. Example of counterfactual visualisation on income prediction dataset [4]. The prediction “lower than $50 000” reaches maximum confidence score for “Never Married” value. Knowing that actual value is “Divorced/Widowed” an example of counterfactual that can be generated in this scenario is “One way you could have got a confidence score of less than 0.5 (0.44) instead is if Marital Status had taken value Married rather than Divorced/Widowed.”

Conclusion

This deliverable provided ways to identify causally relevant features for human-like explanations. We argued that counterfactual explanations are the most human-like. Thus, we formalized counterfactual search based on the learned model that includes saliency measures according to human biases. Cognitive biases, such as recency, controllability, abnormality, and robustness were integrated to the objective function as mathematical terms. We have explained the difference between predictive learning and causal inference and argued that causal graphs should enhance the counterfactual search. With our partners' help, we have built initial causal diagrams and estimated feature correlations for each use case. Moreover, two other methods meant to increase user trust in the system, namely feature importance and feature intervention, have been discussed. The model uncertainty power to improve counterfactual explanations and visualize the counterfactual search space has also been examined.

The future work involved estimating the causal effects with the observed data for each use case based on the initial causal graphs. The estimation is an iterative process that will result in the improvement of the causal diagrams.

The formalized counterfactual search based on the learned model will be implemented and tested in a loop that requires user validation. By comparing the counterfactual solutions obtained using the formalized counterfactual search and those computed using the causal graphs, we hope to contribute to causal machine learning research.

Moreover, as feature saliency is one of the ways all use cases want to receive explanations, we will integrate this feature importance calculation into the TRUST-AI system.

References

[1] Google, “AI Explainability Whitepaper,” Jul. 2020. Accessed: Oct. 12, 2021. [Online]. Available: https://cerre.eu/wp-content/uploads/2020/07/ai_explainability_whitepaper_google.pdf

[2] F. C. Keil, “Explanation and Understanding,” *Annu. Rev. Psychol.*, vol. 57, no. 1, pp. 227–254, Jan. 2006, doi: 10.1146/annurev.psych.57.102904.190100.

[3] T. Miller, “Contrastive explanation: a structural-model approach,” *Knowl. Eng. Rev.*, vol. 36, p. e14, 2021, doi: 10.1017/S0269888921000102.

[4] T. Le, T. Miller, R. Singh, and L. Sonenberg, “Improving Model Understanding and Trust with Counterfactual Explanations of Model Confidence,” *ArXiv Prepr. ArXiv220602790*, 2022.

[5] M. Pawelczyk, S. Bielawski, J. van den Heuvel, T. Richter, and G. Kasneci, “CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms.” arXiv, 2021. doi: 10.48550/ARXIV.2108.00783.

[6] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, New York, NY, USA, 2020, pp. 607–617. doi: 10.1145/3351095.3372850.

[7] A.-H. Karimi, B. Schölkopf, and I. Valera, “Algorithmic Recourse: From Counterfactual Explanations to Interventions,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York, NY, USA, 2021, pp. 353–362. doi: 10.1145/3442188.3445899.

[8] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato, “Getting a CLUE: A Method for Explaining Uncertainty Estimates.” arXiv, 2020. doi: 10.48550/ARXIV.2006.06848.

[9] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019, doi: 10.1016/j.artint.2018.07.007.

[10] J. J. Williams and T. Lombrozo, “The role of explanation in discovery and generalization: evidence from category learning.,” *Cogn. Sci.*, vol. 34, no. 5, pp. 776–806, Jul. 2010, doi: 10.1111/j.1551-6709.2010.01113.x.

[11] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, “Metrics for Explainable AI: Challenges and Prospects.” arXiv, 2018. doi: 10.48550/ARXIV.1812.04608.

[12] J. Pearl, “The seven tools of causal inference, with reflections on machine learning,” *Commun ACM*, vol. 62, no. 3, pp. 54–60, 2019, doi: 10.1145/3241036.

[13] E. Bareinboim and J. Pearl, “Causal Inference by Surrogate Experiments: z-Identifiability,” *CoRR*, vol. abs/1210.4842, 2012, [Online]. Available: http://arxiv.org/abs/1210.4842

[14] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. Cambridge University Press, 2009.

[15] J. Pearl and D. Mackenzie, *The Book of Why*. New York: Basic Books, 2018.

[16] M. A. H. n, J. Hsu, and B. Healy, “A Second Chance to Get Causal Inference Right: A Classification of Data Science Tasks,” *CHANCE*, vol. 32, no. 1, pp. 42–49, Jan. 2019, doi: 10.1080/09332480.2019.1579578.

[17] C. Sacerdote *et al.*, “Lower educational level is a predictor of incident type 2 diabetes in European countries: The EPIC-InterAct study.,” *Int. J. Epidemiol.*, vol. 41, no. 4, pp. 1162–1173, 2012, doi: 10.1093/ije/dys091.

[18] A. Sharma and E. Kiciman, “DoWhy: An End-to-End Library for Causal Inference.” 2020.

[19] S. Petrick, K. Rehdanz, and R. S. J. Tol, “The impact of temperature changes on residential energy consumption,” Kiel Institute for the World Economy (IfW Kiel), Kiel Working Papers 1618, 2010. [Online]. Available: https://ideas.repec.org/p/zbw/ifwkwp/1618.html

[20] E. A. Arens and P. B. Williams, “The effect of wind on energy consumption in buildings,” *Energy Build.*, vol. 1, no. 1, pp. 77–84, 1977, doi: https://doi.org/10.1016/0378-7788(77)90014-7.
[21] J. Piñero *et al.*, “The DisGeNET knowledge platform for disease genomics: 2019 update,” *Nucleic Acids Res.*, vol. 48, no. D1, pp. D845–D855, Nov. 2019, doi: 10.1093/nar/gkz1021.
[22] N. T. Strande *et al.*, “Evaluating the Clinical Validity of Gene-Disease Associations: An Evidence-Based Framework Developed by the Clinical Genome Resource,” *Am. J. Hum. Genet.*, vol. 100, no. 6, pp. 895–906, 2017, doi: https://doi.org/10.1016/j.ajhg.2017.04.015.
[23] P. and D. Amorim, E.-L. Nicole, M. Fredrik, and Sara, “Customer Preferences for Delivery Service Attributes in Attended Home Delivery,” *Chic. Booth Res. Pap. No 20-07*, Jul. 2022, doi: http://dx.doi.org/10.2139/ssrn.3592597.