

# Explaining Predictions from Machine Learning Models: Algorithms, Users, and Pedagogy

Ana Lucic



# Explaining Predictions from Machine Learning Models: Algorithms, Users, and Pedagogy

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Universiteit van Amsterdam  
op gezag van de Rector Magnificus  
prof. dr. G.T.M. ten Dam  
ten overstaan van een door het College voor Promoties ingestelde  
commissie, in het openbaar te verdedigen in  
de Aula der Universiteit  
op vrijdag 23 september 2022, te 14:00 uur

door

Ana Lucic

geboren te Paracin

## Promotiecommissie

Promotor:

Prof. dr. M. de Rijke                      Universiteit van Amsterdam

Co-promotor:

Prof. dr. H. Haned                      Universiteit van Amsterdam

Overige leden:

Prof. dr. F. Silvestri                      Sapienza University of Rome

Prof. dr. M. Lovrić                      McMaster University

Prof. dr. M. Welling                      Universiteit van Amsterdam

Prof. dr. C. Sánchez Gutiérrez      Universiteit van Amsterdam

Dr. F. P. Santos                      Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This research was supported by the Netherlands Organisation for Scientific Research under project number 652.001.003, the Partnership on AI, and Ahold Delhaize.

Copyright © 2022 Ana Lucic, Amsterdam, The Netherlands

Cover by Off Page, Amsterdam

Printed by Off Page, Amsterdam

ISBN: 978-94-93278-21-9

*The more I see, the less I know.*

– Anthony Kiedis

## Acknowledgements

Moving to a foreign country to do a PhD on a foreign topic was not something I had originally planned, but it has nonetheless turned out to be a really rewarding experience. This thesis is the culmination of that experience, and there are many people without whom it would not have been possible.

First and foremost, I want to thank my supervisors who guided me through this whole process. Maarten – thank you for always listening to me and advocating for me. Hinda – thank you for your unwavering support and for always being on my team.

Next, I'd like to thank the committee members who reviewed my thesis. Miroslav – thank you for motivating me to pursue a math major during my undergrad. This decision has fueled everything I've done since then and I'm not sure I would have been confident enough to do it without your encouragement. Fernando – thank you for taking over the FACT-AI course we created for the MSc AI program. I'm really looking forward to seeing how it develops over the next few years. Clarisa – thank you for giving me advice about the challenges I encountered during my fellowship project. Your guidance helped me prioritize and carve out the next steps. Fabrizio – thank you for letting me do my internship project unofficially after I was unable to do a formal internship due to the pandemic. This project is my favorite chapter in this thesis and it has inspired me to shift my research into a new direction. Max – thank you for allowing me to pursue this new direction under your guidance.

I want to thank my paranymphs for standing with me as I defend my thesis. Maartje and Maurits – thank you for being my main sources of support within our group and for sharing all of the ups and downs of doing a PhD with me. I also want to thank the many great people I've met while working at the University of Amsterdam: Alexey, Ali, Ali, Amir, Antonis, Arezoo, Artem, Boris, Chang, Christof, Chuan, Dan, David, Gabriel, Hamid, Ilya, Jiahuan, Jie, Jin, Julia, Julien, Katya, Ke, Maria, Marlies, Mohammad, Mostafa, Mozdheh, Olivier, Pengjie, Petra, Sam, Sami, Shaojie, Spyretta, Svitlana, Thomas, Trond, Vera, Yangjun, and Yifan – thank you for making Science Park (and especially our mint green container) a fun place to work. In particular, I want to thank Harrie, Hosein, Nikos, Rolf, Christophe, Bob, and Tom for your leadership in the group, and of course for all the borrels, biertjes, and bitterballen. Marzieh, Mariya, and Mahsa – thank you for your honesty, friendship, and support over these last few years. Ziming – thank you for letting me babysit your cat for a summer, it was truly one of the highlights of my PhD. I also want to thank the many bright students I had: Michael, Stefan, Kim, Puja, Fije – thank you for teaching me way more than I taught you.

During my PhD, I had the opportunity to do a fellowship at the Partnership on AI. Rebecca – thank you for leading me through my time at the Partnership and for your trust in me amid all the twists and turns involved in doing real-world research.

I've had three lovely roommates while living in Amsterdam, each of which were living with me at very different phases of my PhD. Elisabet – thank you for helping me through the challenges and uncertainties of early PhD life and for showing me every single specialty coffee shop in Amsterdam. Maria – thank you for quarantining with me and supporting me through hairdresser fiascos and global pandemics (not sure what's worse!). Luckily we can always fall back on falafel salads and eggless desserts. Camila – thank you for supporting my leopard print addiction and for always being up for something fun. Special shoutout to my Kelowna roommates Danielle and Bree, who only lived with me briefly during my PhD, but have nonetheless contributed to many “good vibes” and sometimes questionable memories.

I've learned that having a support system outside the research world is incredibly important while pursuing a PhD. Sophie, Tamara, Juliette, Jago, and Thomas – thank you for adopting me at Mystic Garden all those years ago and for the many parties, picnics, pizzas, and pals we've shared since then. Duffey, Allie, Kristine, Craig, Devra, Jeff, and Aly – thank you for not forgetting about me even though I only come home once a year. Thomas – thank you for showing me how to be a kinder, softer person and for always listening to the many things I have to say (ook in het Nederlands). You are what I look forward to. Bedankt aan de hele familie Jak om me welkom te heten met open armen en open harten.

Naravno se moram zahvaliti našom rajom. Hvala mojim Paraćincima: Deda Ujče, Baka Ujna, Ana, Sale, Sanja, Mišo, Margarita, Marija i Nikola, i mojim drugim roditeljima iz Kanade: Ksenija, Mišo, Srdjan, Ksenija, Stanko, Žeki I Gogo. Vi ste mi uvijek u srcu. Najvažnije, hoću da se zahvalim svojim roditeljima, Vesna i Tiho, koji su meni sve moguće učinili. Hvala mojoj Baki koja me je “dva puta rodila” i cuvala od malenog. Ja vas obozavam, i ovo je za vas.

Ana Lucic  
Amsterdam  
July 2022





---



---

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Outline and Questions . . . . .	3
1.1.1	Algorithms . . . . .	3
1.1.2	Users . . . . .	4
1.1.3	Pedagogy . . . . .	5
1.2	Main Contributions . . . . .	5
1.3	Thesis Overview . . . . .	6
1.4	Origins . . . . .	6
<b>I</b>	<b>Algorithms</b>	<b>9</b>
<b>2</b>	<b>Counterfactual Explanations for Tree Ensembles</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Related Work . . . . .	13
2.2.1	Counterfactual Explanations . . . . .	13
2.2.2	Algorithmic Recourse . . . . .	13
2.2.3	Adversarial Examples . . . . .	13
2.2.4	Differentiable Tree-based Models . . . . .	14
2.3	Problem Formulation . . . . .	14
2.4	Method: FOCUS . . . . .	15
2.4.1	Loss Function Definitions . . . . .	15
2.4.2	Tree-based Models . . . . .	16
2.4.3	Approximations of Tree-based Models . . . . .	17
2.4.4	Our Method: FOCUS . . . . .	18
2.4.5	Effects of Hyperparameters . . . . .	18
2.5	Experimental Setup . . . . .	18
2.5.1	Datasets . . . . .	18
2.5.2	Models . . . . .	19
2.5.3	Distance Functions . . . . .	19
2.5.4	Evaluation Metrics . . . . .	20
2.6	Experiment 1: FOCUS vs. FT . . . . .	20
2.6.1	Baseline: Feature Tweaking . . . . .	20
2.6.2	Results . . . . .	21
2.7	Experiment 2: FOCUS vs. DACE . . . . .	23
2.7.1	Baseline: DACE . . . . .	23

2.7.2	Results . . . . .	24
2.8	Discussion and Analysis . . . . .	24
2.8.1	Case Study: Credit Risk . . . . .	25
2.9	Conclusion . . . . .	28
<b>3</b>	<b>Counterfactual Explanations for Graph Neural Networks</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Related Work . . . . .	31
3.2.1	GNN Explainability . . . . .	31
3.2.2	Counterfactual Explanations . . . . .	32
3.2.3	Adversarial Attacks . . . . .	32
3.3	Background . . . . .	32
3.3.1	Graph Neural Networks . . . . .	32
3.3.2	Matrix Sparsification . . . . .	33
3.4	Problem Formulation . . . . .	33
3.5	Method: CF-GNNExplainer . . . . .	34
3.5.1	Adjacency Matrix Perturbation . . . . .	34
3.5.2	Counterfactual Generating Model . . . . .	35
3.5.3	Loss Function Optimization . . . . .	35
3.5.4	CF-GNNExplainer . . . . .	36
3.5.5	Complexity . . . . .	37
3.6	Experimental Setup . . . . .	37
3.6.1	Datasets and Models . . . . .	37
3.6.2	Baselines . . . . .	38
3.6.3	Metrics . . . . .	39
3.6.4	Hyperparameter Search . . . . .	39
3.7	Results . . . . .	40
3.7.1	Main Findings . . . . .	40
3.7.2	Comparison to GNNEXPLAINER . . . . .	41
3.7.3	Summary of Results . . . . .	41
3.8	Conclusion . . . . .	43
<b>II</b>	<b>Users</b>	<b>45</b>
<b>4</b>	<b>Contrastive Explanations for Forecasting Errors</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Related Work . . . . .	50
4.2.1	Local Explanations for Tree Ensembles . . . . .	50
4.2.2	Feature Importance Explanations . . . . .	50
4.2.3	Contrastive Explanations . . . . .	51
4.2.4	Outlier Detection . . . . .	51
4.3	Method: MC-BRP . . . . .	51
4.4	Experimental Setup . . . . .	54
4.4.1	Dataset and Model . . . . .	54
4.4.2	Comparison to LIME . . . . .	55

---

4.4.3	User Study Design . . . . .	55
4.5	Experimental Results . . . . .	58
4.5.1	Objective Questions . . . . .	58
4.5.2	Subjective Questions . . . . .	59
4.6	Discussion . . . . .	60
4.6.1	Comparing Attitudes Conditioned on Background . . . . .	60
4.6.2	Comparing Attitudes in the Treatment Group . . . . .	61
4.6.3	User Study Limitations . . . . .	62
4.7	Conclusion . . . . .	62
<b>III</b>	<b>Pedagogy</b>	<b>65</b>
<b>5</b>	<b>Teaching Responsible AI through Reproducibility</b>	<b>67</b>
5.1	Introduction . . . . .	67
5.2	Related Work . . . . .	69
5.2.1	Characterizing Responsible AI Courses . . . . .	69
5.2.2	Similar Responsible AI Courses . . . . .	70
5.3	Reproducibility in ML Research . . . . .	71
5.4	Learning Objectives . . . . .	71
5.5	Course Setup . . . . .	72
5.5.1	Lectures . . . . .	73
5.5.2	Paper Discussions . . . . .	73
5.5.3	Group Project . . . . .	74
5.5.4	Online Course Format . . . . .	75
5.6	Feedback . . . . .	76
5.6.1	Feedback from Students . . . . .	76
5.6.2	Feedback from the ML Reproducibility Challenge . . . . .	78
5.7	Factors Contributing to a Successful Course . . . . .	78
5.7.1	General . . . . .	78
5.7.2	Concerning FACT-AI . . . . .	79
5.7.3	Concerning Reproducibility . . . . .	80
5.8	Areas of Improvement . . . . .	80
5.8.1	General . . . . .	80
5.8.2	Concerning FACT-AI . . . . .	80
5.8.3	Concerning Reproducibility . . . . .	81
5.9	Conclusion . . . . .	81
<b>6</b>	<b>Conclusions</b>	<b>83</b>
6.1	Main Findings . . . . .	83
6.1.1	Algorithms . . . . .	83
6.1.2	Users . . . . .	84
6.1.3	Pedagogy . . . . .	84
6.2	Future Directions . . . . .	84

---



# Introduction

---

Machine learning (ML) is the study of algorithms that learn models directly from data [94]. Such algorithms are typically self-improving – their parameters are updated iteratively based on the data they receive, thereby *learning* a model that is representative of the data. Once an ML model is trained, it is usually evaluated on unseen data in order to test its generalization capabilities. The ability to generalize to new situations is one of the most important aspects of ML models, and is perhaps the reason such models are often referred to as “intelligent” [131]. In other words, ML models use information from the past (i.e., historical data) to make predictions about the future (i.e., unseen data).

The field of ML has enjoyed great success in the last decade primarily due to the increased availability of data and computational resources [4]. As ML models have become more prominent in decision-making scenarios [22], there has been an increased demand for ensuring such models are (i) fair, (ii) accountable, (iii) confidential, and (iv) transparent [99].<sup>1</sup> However, ML models can be difficult to interpret due to their complex architectures and the large numbers of parameters involved, effectively deeming them “black-boxes” [20]. In this thesis, we primarily focus on developing methods to increase *transparency*, which we define as mechanisms that provide insight into an ML model. This knowledge is typically presented to a user in the form of an *explanation*.

Recently, the artificial intelligence (AI) research community has embarked on the development of explainable artificial intelligence (XAI): a relatively new subfield of AI where the aim is to explain predictions from complex ML models [47]. Explanations can be used to make ML models more accountable to various stakeholders involved in the pipeline by providing insight into not only how the model arrived at its decision, but also how to change or contest the decision, if necessary [141]. We distinguish between two main types of explanations:

- **Behavior-based explanations:** provide insight into how an ML model makes predictions from an algorithmic or mathematical perspective. For example, ranking the most important features [86, 107], identifying influential [69, 116] or prototypical [73, 129] training samples, or generating counterfactual perturbations [122, 139, 141]. Behavior-based explanations are important for understanding the internal processes of ML models.

---

<sup>1</sup>Although confidentiality and transparency may seem like contradictory objectives, confidentiality typically refers to preserving the privacy of individuals within a training dataset, while transparency refers to the ML model and the process that went into deploying it.

- **Process-based explanations:** provide insight into the ML modeling pipeline. For example, detailing how the data were collected and preprocessed [39], or reporting on how the model was trained and evaluated [91]. Process-based explanations are important for ensuring that ML research is conducted in a responsible and reproducible manner.

This thesis has three parts: the first focuses on algorithms, the second focuses on users, and the third focuses on pedagogy. In the first two parts of this thesis, we develop methods for generating behavior-based explanations, which is what the majority of existing XAI methods produce. Guidotti et al. [47] develop a taxonomy for classifying XAI methods using four main criteria, we slightly adapt their taxonomy as follows:<sup>2</sup>

- **Problem:** the type of explanation we want to generate.
  - (i) *Global explanations:* interpret ML model behavior in general, i.e., how it makes predictions across data points.
  - (ii) *Local explanations:* interpret individual ML model predictions.
  - (iii) *Inspection:* interpret model behavior through visual representations (globally or locally).
  - (iv) *Transparent design:* model is inherently interpretable (globally or locally).
- **Model:** the dependence on model class.
  - (i) *Model-specific:* requires full access to the model’s inner workings, where the model can be a neural network, tree ensemble, support vector machine, etc.
  - (ii) *Model-agnostic:* treats model as a “black-box” and is therefore not dependent on its inner workings.
- **Explainer:** the mechanism used to generate explanations, e.g., decision rules, feature attributions, sensitivity analysis, prototype selection, etc.
- **Data:** the type of data being explained, e.g., tabular, image, text, graph, etc.

In Chapters 2 and 3, we develop and investigate local, model-specific explanation methods. We focus on local explanations methods because they are a natural precursor to global explanations [137]. We focus on model-specific explanations because in many practical scenarios, we have full access to the ML model and can therefore make use of its inner workings when generating explanations. The algorithm proposed in Chapter 2 is specific to tree ensembles and operates on tabular data, while the algorithm proposed in Chapter 3 is specific to graph neural networks (GNNs) and operates on graph data.

---

<sup>2</sup>We refer to explanations as “global” or “local” since this aligns more closely with standard terms recently in the literature. We also introduce the distinction between “model-specific” and “model-agnostic”.

In general, we take the position that treating the model as a “black-box” in order to interpret its predictions is a somewhat contradictory statement; how can we understand what a model is doing if do not have access to its internal processes? However, there exist use cases where we need explanations but we do not have full access to the model, such as auditing [105]. To accommodate such scenarios, we propose a method in Chapter 4 that is model-agnostic in principle, but we evaluate it in a model-specific manner.

In the third and final part of the thesis, we shift from investigating behavior-based explanations for explaining individual predictions to investigating process-based explanations for explaining the ML modeling pipeline. We do so from a pedagogical point of view. We use reproducibility as a mechanism for teaching about responsible AI concepts: fairness, accountability, confidentiality, and transparency in a graduate-level course at the University of Amsterdam. In Chapter 5, we report on our experiences and lessons learned after teaching the course over two academic years.

## 1.1 Research Outline and Questions

---

This thesis focuses on explaining for ML models in three different contexts: algorithms (Chapters 2 and 3), users (Chapter 4), and pedagogy (Chapter 5). Below, we describe the main research questions for each chapter.

### 1.1.1 Algorithms

In the first part of the thesis, we focus on the algorithmic component of generating explanations for ML predictions. Based on existing work from the philosophy, cognitive science and social psychology disciplines, Miller [90] identifies various aspects of explainability that AI researchers should pay attention to when generating explanations for ML models. One of the main recommendations is that explanations should be *contrastive*: they should allow the user to compare and contrast between the original instance (i.e., data point) and a counterfactual case.<sup>3</sup> *Counterfactual explanations* are defined as the minimal perturbations to the input data that such that the prediction changes [141]. Counterfactual explanations have been identified as explanations that can “*provide information to the data subject that is both easily digestible and practically useful for understanding the reasons for a decision, challenging them, and altering future behavior for a better result*” [141].

In the first chapter in this part of the thesis, we develop and evaluate a method for generating counterfactual explanations specific to tree ensembles. We focus on tree ensembles because the majority of existing research on counterfactual explanations is either (i) model-agnostic or (ii) specific to deep learning [61], even though tree ensembles perform well on tabular data and are widely used in

---

<sup>3</sup>There has recently been discussion on the differences between *contrastive* and *counterfactual* explanations [61, 122]. For the purposes of this thesis, we use these terms interchangeably.

industrial settings [118]. In this chapter, we aim to answer the following research question:

**RQ1** Can we generate counterfactual explanations for tree-based models using gradient-based optimization?

To answer **RQ1**, we propose a method that introduces differentiable approximations into the optimization framework, which allows us to use standard gradient-based optimization techniques to generate counterfactual explanations. We find that, unlike existing approaches, our method (i) produces counterfactual examples for all instances in a dataset, (ii) produces counterfactual examples that are closer to the original instances compared to existing approaches, and (iii) can handle larger model sizes.

Although there exist many methods for generating counterfactual explanations for tabular, text, and image data (see surveys by Verma et al. [139], Karimi et al. [63], and Stepin et al. [122]), there are relatively few for graph data.<sup>4</sup> In the second chapter in this part of the thesis, we investigate the following research question:

**RQ2** Can we extend our counterfactual explanation method for tree-based models to graph-based models?

To answer **RQ2**, we formalize the problem of counterfactual explanations for graph neural networks (GNNs) and propose a method for generating them by iteratively removing entries in the adjacency matrix, which corresponds to removing edges in the graph. We also propose an experimental setup for evaluating counterfactual explanations for GNNs and find that our explanation method is able to generate counterfactual explanations that are more minimal and more accurate in comparison to the baselines.

### 1.1.2 Users

Given that explanations are created in order to be consumed by users, it is a natural next step to think about incorporating a more user-centric perspective into the development of XAI methods [31]. In this part of the thesis, we investigate generating explanations that are situated and evaluated within a particular deployment context. We ask the following research question:

**RQ3** Given a real-world use case, can we create an explanation method based on this use case and evaluate it in a context-specific manner?

To answer **RQ3**, we first identify a use case where users want to see explanations for ML models: explaining errors in forecasting predictions. We propose a method for generating explanations based on the specifics of this use case and design a user study to evaluate how interpretable and actionable our explanation are. We also test users' subjective attitudes towards the explanations generated by our method.

---

<sup>4</sup>At the time of writing [85], there were no existing methods for generating counterfactual explanations for graph data.



### 1.1.3 Pedagogy

In the final part of this thesis, we transition from explaining ML predictions to explaining best practices for conducting ML research. We ask the following research question:

**RQ4** How can we teach about responsible AI topics to a technical, research-oriented audience?

To answer **RQ4**, we design a graduate-level course about fairness, accountability, confidentiality, and transparency in AI. We describe how we structured the course around a reproducibility project and report in detail on the insights gained from teaching the course over two academic years. We emphasize that conducting research in a responsible and reproducible manner is not only important for individual ML researchers, but is also essential for scientific progress in general.

## 1.2 Main Contributions

---

In this section, we summarize the main contributions of this thesis.

### Theoretical contributions

1. A formalization of the counterfactual explanation problem for GNNs (Chapter 3).
2. An experimental setup for evaluating counterfactual explanations for GNNs (Chapter 3).
3. A user study framework for evaluating the effectiveness of contrastive explanations (Chapter 4).
4. An analysis on the difference in attitudes towards explanations between different types of stakeholders (Chapter 4).

### Algorithmic contributions

5. Flexible Optimizable CoUnTercfactual Explanations for Tree EnsembleS (FOCUS): an algorithm for generating counterfactual explanations for tree ensembles (Chapter 2).
6. CF-GNNExplainer: an algorithm for generating counterfactual explanations for GNNs (Chapter 3).
7. Monte Carlo Bounds for Reasonable Predictions (MC-BRP): an algorithm for generating explanations about errors in forecasting predictions (Chapter 4).

## Pedagogical contributions

8. A teaching setup for a course about responsible AI with a focus on reproducibility (Chapter 5), including a set of guidelines for implementing similar courses in the future based on our insights.

## 1.3 Thesis Overview

---

This thesis is organized into three parts, each part can be read independently.

The first part focuses on proposing new algorithms for explaining predictions from ML models. Specifically, we propose methods for generating counterfactual explanations for tree-based models (Chapter 2), and for graph-based models (Chapter 3). These methods can be applied on any tree- or graph-based model, respectively.

The second part focuses on the interaction between ML explanations and the users who consume them. We propose a method for explaining errors in forecasting predictions (Chapter 4). To evaluate our method, we propose a user study with both objective and subjective components, where we contrast and compare the results between two types of users: researchers and practitioners.

In the third part of the thesis, we shift our focus from translating knowledge about individual predictions to transferring knowledge to the next generation of researchers. We propose a course setup for teaching about responsible AI topics to a graduate-level audience and reflect on our learnings from past implementations of the course at the University of Amsterdam (Chapter 5).

## 1.4 Origins

---

Below we list the publications that are the origins of each chapter.

**Chapter 1** is based in part on the following paper:

- Ana Lucic, Madhulika Srikumar, Umang Bhatt, Alice Xiang, Ankur Taly, Q. Vera Liao, and Maarten de Rijke. A Multistakeholder Approach Towards Evaluating AI Transparency Mechanisms. *CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI*, 2021.

AL designed the framework during a research fellowship at the Partnership on AI, based on discussions with all authors. All authors contributed to providing feedback, AL did most of the writing.

**Chapter 2** is based on the following paper:

- Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles. *AAAI Conference on Artificial Intelligence*, 2022.

AL and HO designed the method. AL ran the experiments. All authors contributed to the writing, AL did most of the writing.

**Chapter 3** is based on the following paper:

- Ana Lucic, Maartje ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. *International Conference on Artificial Intelligence and Statistics*, 2022.

AL designed the method and ran the experiments. All authors contributed to the writing, AL did most of the writing.

**Chapter 4** is based on the following paper:

- Ana Lucic, Hinda Haned, and Maarten de Rijke. Why Does My Model Fail? Contrastive Local Explanations for Retail Forecasting. *ACM Conference on Fairness, Accountability, and Transparency*, 2020.

AL designed the method and ran the experiments. All authors contributed to the writing, AL did most of the writing.

**Chapter 5** is based on the following paper:

- Ana Lucic, Maurits Bleeker, Sami Jullien, Samarth Bhargav, and Maarten de Rijke. Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence. *AAAI Symposium on Educational Advances in Artificial Intelligence*, 2022.

AL and MB designed the course and implemented it together with Mdr. All authors contributed to the writing, AL did most of the writing.

The writing of this thesis also benefited from work on the following publications:

- Ana Lucic, Hinda Haned, and Maarten de Rijke. Explaining Predictions from Tree-based Boosting Ensembles. *SIGIR Workshop on Fairness, Accountability, Confidentiality, and Transparency in Information Retrieval*, 2019.
- Ana Lucic, Maurits Bleeker, Maarten de Rijke, Koustuv Sinha, Sami Jullien, and Robert Stojnic. Towards Reproducible Machine Learning Research in Information Retrieval. *SIGIR Tutorial on Reproducibility*, 2022.
- Ana Lucic, Maurits Bleeker, Samarth Bhargav, Jessica Zosa Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, and Robert Stojnic. Towards Reproducible Machine Learning Research in Natural Language Processing. *ACL Tutorial on Reproducibility*, 2022.

- Ana Lucic, Sheeraz Ahmad, Amanda Furtado Brinhosa, Q. Vera Liao, Himani Agrawal, Umang Bhatt, Krishnaram Kenthapadi, Alice Xiang, Maarten de Rijke, and Nicholas Drabowski. Towards the Use of Saliency Maps for Explaining Low-Quality Electrocardiograms to End Users. *IJCAI Workshop on Explainable AI*, 2022.
- Kim de Bie, Ana Lucic, and Hinda Haned. To Trust or Not to Trust a Regressor: Estimating and Explaining Trustworthiness of Regression Predictions. *ICML Workshop on Human in the Loop Learning*, 2021.
- Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. A Song of (Dis)agreement: Evaluating the Evaluation of Explainable Artificial Intelligence in Natural Language Processing. *International Conference on Hybrid Human-Artificial Intelligence*, 2022.
- Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. Order in the Court: Explainable AI Methods Prone to Disagreement. *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021.
- Surya Karunagaran, Ana Lucic, and Christine Custis. XAI Toolsheets: An Evaluation Framework for XAI Tools. *IJCAI Workshop on Explainable AI*, 2022.

# Part I

# Algorithms



# Counterfactual Explanations for Tree Ensembles

---

In the first part of this thesis, we explore creating algorithms for explaining predictions from various types of machine learning (ML) models. In this chapter, we address the following research question:

**RQ1:** Can we generate counterfactual explanations for tree-based models using gradient-based optimization?

Existing methods for generating counterfactual explanations for tree-based models are either based on heuristics [132] or on integer linear programming techniques [58]. The former do not necessarily converge to an optimal solution, while the latter can be extremely computationally intensive.

The answer to **RQ1** is yes: we can achieve this by generating probabilistic approximations of tree-based models, which are differentiable and can therefore be used within a standard gradient-based optimization framework. Our experimental results show that our proposed algorithm can generate minimal counterfactual explanations in a more efficient and reliable manner in comparison to the baselines.

## 2.1 Introduction

---

As ML models are prominently applied and their outcomes have a substantial effect on the general population, there is an increased demand for understanding what contributes to their predictions [29]. For an individual who is affected by the predictions of these models, it would be useful to have an *actionable* explanation – one that provides insight into how these decisions can be *changed*. The General Data Protection Regulation (GDPR) is an example of recently enforced regulation in Europe which gives an individual the right to an explanation for algorithmic decisions, making the interpretability problem a crucial one for organizations that wish to adopt more data-driven decision-making processes [32].

Counterfactual explanations are a natural solution to this problem since they frame the explanation in terms of what input (feature) changes are required to change the output (prediction). For instance, a user may be denied a loan

---

This chapter was published at the AAAI Conference on Artificial Intelligence (AAAI 2022) under the title “FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles” [84].

based on the prediction of an ML model used by their bank. A counterfactual explanation could be: “*Had your income been €1000 higher, you would have been approved for the loan.*” We focus on finding *optimal* counterfactual explanations: the *minimal* changes to the input required to change the outcome.

Counterfactual explanations are based on *counterfactual examples*: generated instances that are close to an existing instance but have an alternative prediction. The difference between the original instance and the counterfactual example is the counterfactual explanation. Wachter et al. [141] propose framing the problem as an optimization task, but their work assumes that the underlying machine learning models are differentiable, which excludes an important class of widely applied and highly effective non-differentiable models: tree ensembles. We propose a method that relaxes this assumption and builds upon the work of Wachter et al. by introducing differentiable approximations of tree ensembles that can be used in such an optimization framework. Alternative non-optimization approaches for generating counterfactual explanations for tree ensembles involve an extensive search over many possible paths in the ensemble that could lead to an alternative prediction [132].

Given a trained tree-based model  $f$ , we probabilistically approximate  $f$  by replacing each split in each tree with a sigmoid function centred at the splitting threshold. If  $f$  is an ensemble of trees, then we also replace the maximum operator with a softmax. This approximation allows us to generate a counterfactual example  $\bar{x}$  for an instance  $x$  based on the minimal perturbation of  $x$  such that the prediction changes:  $y_x \neq y_{\bar{x}}$ , where  $y_x$  and  $y_{\bar{x}}$  are the labels  $f$  assigns to  $x$  and  $\bar{x}$ , respectively. This leads us to our main research question in this chapter:

*Are counterfactual examples generated by our method closer to the original input instances than those generated by existing heuristic methods?*

Our main findings are that our proposed method is (i) a more *effective* counterfactual explanation method for tree ensembles than previous approaches since it manages to produce counterfactual examples that are closer to the original input instances than existing approaches; (ii) a more *efficient* counterfactual explanation method for tree ensembles since it is able to handle larger models than existing approaches; and (iii) a more *reliable* counterfactual explanation method for tree ensembles since it is able to generate counterfactual explanations for all instances in a dataset, unlike existing approaches specific to tree ensembles.

In the following sections, we examine existing work related to ours (Section 2.2) and formalize the counterfactual explanation problem (Section 2.3). We then describe the details of our method, Flexible Optimizable CoCounterfactual Explanations for Tree EnsembleS (FOCUS), in Section 2.4. In Section 2.5, we explain the experimental setup, followed by the experimental results in Sections 2.6 and 2.7. We analyze our findings in Section 2.8 and conclude in Section 2.9.



## 2.2 Related Work

Based on the taxonomy described in Chapter 1, our setting in this chapter is a *local explanation* problem for *tree ensembles*. We use *sensitivity analysis*, specifically counterfactual perturbations, on *tabular* data to generate our explanations. Our work is related to counterfactual explanations in general (Section 2.2.1), algorithmic recourse (Section 2.2.2), adversarial examples (Section 2.2.3), and differentiable tree-based models (Section 2.2.4).

### 2.2.1 Counterfactual Explanations

Counterfactual examples have been used in a variety of ML areas, such as reinforcement learning [89], deep learning [3], and XAI. Previous XAI methods for generating counterfactual examples are either model-agnostic [60, 70, 93, 104, 138] or model-specific [27, 42, 58, 109, 132, 141]. Model-agnostic approaches treat the original model as a “black-box” and only assume query access to the model, whereas model-specific approaches typically do not make this assumption and can therefore make use of its inner workings (see Chapter 1).

Our work is a model-specific approach for generating counterfactual examples through optimization. Previous model-specific work for generating counterfactual examples through optimization has solely been conducted on differentiable models [27, 42, 141].

### 2.2.2 Algorithmic Recourse

Algorithmic recourse is a line of research that is closely related to counterfactual explanations, except that methods for algorithmic recourse include the additional restriction that the resulting explanation must be *actionable* [57, 62, 63, 136]. This is done by selecting a subset of the features to which perturbations can be applied in order to avoid explanations that suggest impossible or unrealistic changes to the feature values (i.e., change *age* from 50  $\rightarrow$  25). Although this work has produced impressive theoretical results, it is unclear how realistic they are in practice, especially for complex ML models such as tree ensembles. Existing algorithmic recourse methods cannot solve our task because they (i) are either restricted to solely linear [136] or differentiable [57] models, or (ii) require access to causal information [62, 63], which is rarely available in real world settings.

### 2.2.3 Adversarial Examples

Adversarial examples are a type of counterfactual example with the additional constraint that the minimal perturbation results in an alternative prediction that is *incorrect*. There are a variety of methods for generating adversarial examples [16, 41, 125, 128]; a more complete overview can be found in the work of [13]. The main difference between adversarial examples and counterfactual examples is in the intent: adversarial examples are meant to *fool* the model, whereas counterfactual examples are meant to *explain* the model.

### 2.2.4 Differentiable Tree-based Models

Part of our contribution involves constructing differentiable versions of tree ensembles by replacing each splitting threshold with a sigmoid function. This can be seen as using a (small) neural network to obtain a smooth approximation of each tree. Neural decision trees [7, 144] are also differentiable versions of trees, which use a full neural network instead of a simple sigmoid. However, these do not optimize for approximating an already trained model. Therefore, unlike our method, they are not an obvious choice for finding counterfactual examples for an existing model. Soft decision trees [54] are another example of differentiable trees, which instead approximate a neural network with a decision tree. This can be seen as the inverse of our task.

## 2.3 Problem Formulation

---

A *counterfactual explanation* for an instance  $x$  and a model  $f$ ,  $\Delta_x$ , is a minimal perturbation of  $x$  that changes the prediction of  $f$ .  $f$  is a probabilistic classifier, where  $f(y | x)$  is the probability of  $x$  belonging to class  $y$  according to  $f$ . The prediction of  $f$  for  $x$  is the most probable class label  $y_x = \arg \max_y f(y | x)$ , and a perturbation  $\bar{x}$  is a counterfactual example for  $x$  if, and only if,  $y_x \neq y_{\bar{x}}$ , that is:

$$\arg \max_y f(y | x) \neq \arg \max_{y'} f(y' | \bar{x}). \quad (2.1)$$

In addition to changing the prediction, the distance between  $x$  and  $\bar{x}$  should also be minimized. We therefore define an *optimal counterfactual example*  $\bar{x}^*$  as:

$$\bar{x}^* := \arg \min_{\bar{x}} d(x, \bar{x}) \text{ such that } y_x \neq y_{\bar{x}}, \quad (2.2)$$

where  $d(x, \bar{x})$  is a differentiable distance function. The corresponding *optimal counterfactual explanation*  $\Delta_x^*$  is:

$$\Delta_x^* = \bar{x}^* - x. \quad (2.3)$$

This definition aligns with previous ML work on counterfactual explanations [60, 70, 132]. We note that this notion of *optimality* is purely from an algorithmic perspective and does not necessarily translate to optimal changes in the real world, since the latter are dependent on the context in which they are applied. It should be noted that if the loss space is non-convex, it is possible that more than one optimal counterfactual explanation exists.

Minimizing the distance between  $x$  and  $\bar{x}$  should ensure that  $\bar{x}$  is as close to the decision boundary as possible. This distance indicates the effort it takes to apply the perturbation in practice, and an optimal counterfactual explanation shows how a prediction can be changed with the least amount of effort. An optimal explanation provides the user with interpretable and potentially actionable feedback related to understanding the predictions of model  $f$ .

Wachter et al. [141] recognized that counterfactual examples can be found through gradient descent if the task is cast as an optimization problem. Specifically, they use a loss consisting of two components: (i) a prediction loss to change the prediction of  $f$ :  $\mathcal{L}_{pred}(x, \bar{x} \mid f)$ , and (ii) a distance loss to minimize the distance  $d$ :  $\mathcal{L}_{dist}(x, \bar{x} \mid d)$ . The complete loss is a linear combination of these two parts, with a weight  $\beta \in \mathbb{R}_{>0}$ :

$$\mathcal{L}(x, \bar{x} \mid f, d) = \mathcal{L}_{pred}(x, \bar{x} \mid f) + \beta \mathcal{L}_{dist}(x, \bar{x} \mid d). \quad (2.4)$$

The assumption here is that an optimal counterfactual example  $\bar{x}^*$  can be found by minimizing the overall loss:

$$\bar{x}^* = \arg \min_{\bar{x}} \mathcal{L}(x, \bar{x} \mid f, d). \quad (2.5)$$

Wachter et al. [141] propose a prediction loss  $\mathcal{L}_{pred}$  based on the mean-squared-error. A clear limitation of this approach is that it assumes  $f$  is differentiable. This excludes many commonly used ML models, including tree-based models, which we focus on in this work.

## 2.4 Method: FOCUS

To mimic many real-world scenarios, we assume there exists a trained model  $f$  that we need to explain. The goal here is not to create a new, inherently interpretable tree-based model, but rather to explain a model that already exists.

### 2.4.1 Loss Function Definitions

We use a hinge-loss since we assume a classification task:

$$\mathcal{L}_{pred}(x, \bar{x} \mid f) = \mathbb{1} \left[ \arg \max_y f(y \mid x) = \arg \max_{y'} f(y' \mid \bar{x}) \right] \cdot f(y' \mid \bar{x}). \quad (2.6)$$

Allowing for flexibility in the choice of distance function allows us to tailor the explanations to the end-users' needs. We make the preferred notion of *minimality* explicit through the choice of distance function. Given a differentiable distance function  $d$ , the distance loss is:

$$\mathcal{L}_{dist}(x, \bar{x}) = d(x, \bar{x}). \quad (2.7)$$

Building off of Wachter et al. [141], we propose incorporating differentiable approximations of non-differentiable models to use in the gradient-based optimization framework. Since the approximation  $\tilde{f}$  is derived from the original model  $f$ , it should match  $f$  closely:  $\tilde{f}(y \mid x) \approx f(y \mid x)$ . We define the approximate prediction loss as follows:

$$\tilde{\mathcal{L}}_{pred}(x, \bar{x} \mid f, \tilde{f}) = \mathbb{1} \left[ \arg \max_y f(y \mid x) = \arg \max_{y'} f(y' \mid \bar{x}) \right] \cdot \tilde{f}(y' \mid \bar{x}). \quad (2.8)$$

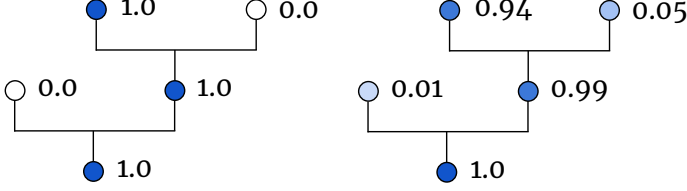


Figure 2.1: Left: A decision tree  $\mathcal{T}$  and node activations for a single instance. Right: a differentiable approximation of the same tree  $\tilde{\mathcal{T}}$  and activations for the same instance.

This loss is based both on the original model  $f$  and the approximation  $\tilde{f}$ : the loss is active as long as the prediction according to  $f$  has not changed, but its gradient is based on the differentiable  $\tilde{f}$ . This prediction loss encourages the perturbation to have a different prediction than the original instance by penalizing an unchanged instance. The approximation of the complete loss becomes:

$$\tilde{\mathcal{L}}(x, \bar{x} \mid f, \tilde{f}, d) = \tilde{\mathcal{L}}_{pred}(x, \bar{x} \mid f, \tilde{f}) + \beta \cdot \mathcal{L}_{dist}(x, \bar{x} \mid d). \quad (2.9)$$

Since we assume that it approximates the complete loss,

$$\tilde{\mathcal{L}}(x, \bar{x} \mid f, \tilde{f}, d) \approx \mathcal{L}(x, \bar{x} \mid f, d), \quad (2.10)$$

we also assume that an optimal counterfactual example can be found by minimizing it:

$$\bar{x}^* \approx \arg \min_{\bar{x}} \tilde{\mathcal{L}}(x, \bar{x} \mid f, \tilde{f}, d). \quad (2.11)$$

### 2.4.2 Tree-based Models

To obtain the differentiable approximation  $\tilde{f}$  of  $f$ , we construct a probabilistic approximation of the original tree ensemble  $f$ . Tree ensembles are based on decision trees; a single decision tree  $\mathcal{T}$  uses a binary-tree structure to make predictions about an instance  $x$  based on its features. Figure 2.1 shows a simple decision tree consisting of five nodes. A node  $j$  is activated if its parent node  $p_j$  is activated and feature  $x_{f_j}$  is on the correct side of the threshold  $\theta_j$ ; which side is the correct side depends on whether  $j$  is a *left* or *right* child. The root node is an exception, it is always activated. Let  $t_j(x)$  indicate if node  $j$  is activated:

$$t_j(x) = \begin{cases} 1, & \text{if } j \text{ is the root,} \\ t_{p_j}(x) \cdot \mathbb{1}[x_{f_j} > \theta_j], & \text{if } j \text{ is a left child,} \\ t_{p_j}(x) \cdot \mathbb{1}[x_{f_j} \leq \theta_j], & \text{if } j \text{ is a right child.} \end{cases} \quad (2.12)$$

$\forall x, t_0(x) = 1$ . Nodes that have no children are called *leaf nodes*; an instance  $x$  always ends up in a single leaf node. Every leaf node  $j$  has its own predicted

distribution  $\mathcal{T}(y | j)$ ; the prediction of the full tree is given by its activated leaf node. Let  $\mathcal{T}_{leaf}$  be the set of leaf nodes in  $\mathcal{T}$ , then:

$$(j \in \mathcal{T}_{leaf} \wedge t_j(x) = 1) \rightarrow \mathcal{T}(y | x) = \mathcal{T}(y | j). \quad (2.13)$$

Alternatively, we can reformulate this as a sum over leaves:

$$\mathcal{T}(y | x) = \sum_{j \in \mathcal{T}_{leaf}} t_j(x) \cdot \mathcal{T}(y | j). \quad (2.14)$$

Generally, tree ensembles are deterministic; let  $f$  be an ensemble of  $M$  many trees with weights  $\omega_m \in \mathbb{R}$ , then:

$$f(y | x) = \arg \max_y \sum_{m=1}^M \omega_m \cdot \mathcal{T}_m(y' | x). \quad (2.15)$$

### 2.4.3 Approximations of Tree-based Models

If  $f$  is not differentiable, we are unable to calculate its gradient with respect to the input  $x$ . However, the non-differentiable operations in our formulation are (i) the indicator function, and (ii) a maximum operation, both of which can be approximated by differentiable functions. First, we introduce the  $\tilde{t}_j(x)$  function that *approximates the activation of node  $j$* :  $\tilde{t}_j(x) \approx t_j(x)$ , using a sigmoid function with parameter  $\sigma \in \mathbb{R}_{>0}$ :  $\text{sig}(z) = (1 + \exp(\sigma \cdot z))^{-1}$  and

$$\tilde{t}_j(x) = \begin{cases} 1, & \text{if } j \text{ is the root,} \\ \tilde{t}_{p_j}(x) \cdot \text{sig}(\theta_j - x_{f_j}), & \text{if } j \text{ is left child,} \\ \tilde{t}_{p_j}(x) \cdot \text{sig}(x_{f_j} - \theta_j), & \text{if } j \text{ is right child.} \end{cases} \quad (2.16)$$

As  $\sigma$  increases,  $\tilde{t}_j$  approximates  $t_j$  more closely. Next, we introduce a *tree approximation*:

$$\tilde{\mathcal{T}}(y | x) = \sum_{j \in \mathcal{T}_{leaf}} \tilde{t}_j(x) \cdot \mathcal{T}(y | j). \quad (2.17)$$

The approximation  $\tilde{\mathcal{T}}$  uses the same tree structure and thresholds as  $\mathcal{T}$ . However, its activations are no longer deterministic but instead are dependent on the distance between the feature values  $x_{f_j}$  and the thresholds  $\theta_j$ . Lastly, we replace the maximum operation of  $f$  by a softmax with temperature  $\tau \in \mathbb{R}_{>0}$ , resulting in:

$$\tilde{f}(y | x) = \frac{\exp\left(\tau \cdot \sum_{m=1}^M \omega_m \cdot \tilde{\mathcal{T}}_m(y | x)\right)}{\sum_{y'} \exp\left(\tau \cdot \sum_{m=1}^M \omega_m \cdot \tilde{\mathcal{T}}_m(y' | x)\right)}. \quad (2.18)$$

The approximation  $\tilde{f}$  is based on the original model  $f$  and the parameters  $\sigma$  and  $\tau$ . This approximation is applicable to any tree-based model, and how well  $\tilde{f}$  approximates  $f$  depends on the choice of  $\sigma$  and  $\tau$ . The approximation is potentially perfect since

$$\lim_{\sigma, \tau \rightarrow \infty} \tilde{f}(y | x) = f(y | x). \quad (2.19)$$

### 2.4.4 Our Method: FOCUS

We call our method FOCUS: Flexible Optimizable Counterfactual Explanations for Tree EnsembleS. It takes as input an instance  $x$ , a tree-based classifier  $f$ , and two hyperparameters:  $\sigma$  and  $\tau$ , which we use to create the approximation  $\tilde{f}$ . Following Equation 2.11, FOCUS outputs the optimal counterfactual example  $\bar{x}^*$ , from which we derive the optimal counterfactual explanation  $\Delta_x^* = \bar{x}^* - x$ .

### 2.4.5 Effects of Hyperparameters

Increasing  $\sigma$  in  $\tilde{f}$  eventually leads to exact approximations of the indicator functions, while increasing  $\tau$  in  $\tilde{f}$  leads to a completely unimodal softmax distribution. It should be noted that our approximation  $\tilde{f}$  is not intended to replace the original model  $f$  but rather to create a differentiable version of  $f$  from which we can generate counterfactual examples through optimization. In practice, the original model  $f$  would still be used to make predictions and the approximation would solely be used to generate counterfactual examples.

## 2.5 Experimental Setup

---

We consider 42 experimental settings to find the best counterfactual explanations using FOCUS. We jointly tune the hyperparameters of FOCUS ( $\sigma, \tau, \beta, \alpha$ ) using Adam [67] for 1,000 iterations. We choose the hyperparameters that produce (i) a valid counterfactual example for every instance in the dataset, and (ii) the smallest mean distance between corresponding pairs  $(x, \bar{x})$ .

We evaluate FOCUS on four binary classification datasets and three types of tree-based models for each dataset. We compare against two baselines that generate counterfactual examples for tree ensembles based on the inner workings of the model: Feature Tweaking (FT) by Tolomei et al. [132] and Distribution-Aware Counterfactual Explanations (DACE) by Kanamori et al. [58].

### 2.5.1 Datasets

We evaluate FOCUS on four binary classification tasks using the following datasets:

- The WINE QUALITY dataset [134] has 4,898 instances and 11 features. The task is about predicting the quality of white wine on a 0–10 scale. We adapt this to a binary classification setting by labelling the wine as “high quality” if the quality is  $\geq 7$ .
- The HELOC dataset [34] has 10,459 instances and 23 features. The task is from the Explainable Machine Learning Challenge at NeurIPS 2017, where the task is to predict whether or not a customer will default on their loan.

- The COMPAS dataset [98] has 6,172 instances and 6 features. It is used for detecting bias in ML systems, where the task is predicting whether or not a criminal defendant will reoffend upon release.
- The SHOPPING dataset [135] has 12,330 instances and 9 features. The task entails predicting whether or not an online website visit results in a purchase.

We scale all features such that their values are in the range  $[0, 1]$  and remove categorical features.

### 2.5.2 Models

We train three types of tree-based models on 70% of each dataset: decision trees (DTs), random forests (RFs), and adaptive boosting (AB) with DTs as the base learners. We use the remaining 30% to find counterfactual examples for this test set. In total we have 12 models (4 datasets  $\times$  3 tree-based models).

### 2.5.3 Distance Functions

In our experiments, we generate different types of counterfactual explanations using different types of distance functions. We note that the flexibility of FOCUS allows for the use of any differentiable distance function. Euclidean distance measures the geometric displacement:

$$d_{Euclidean}(x, \bar{x}) = \sqrt{\sum_i (x_i - \bar{x}_i)^2}. \quad (2.20)$$

Cosine distance measures the angle by which  $\bar{x}$  deviates from  $x$  – whether  $\bar{x}$  preserves the relationship between features in  $x$ :

$$d_{Cosine}(x, \bar{x}) = 1 - \frac{\sum_i (x_i \cdot \bar{x}_i)}{\|x\| \|\bar{x}\|}. \quad (2.21)$$

Manhattan distance (i.e.,  $L1$ -norm) measures per feature differences, minimizing the number of features perturbed and therefore inducing sparsity:

$$d_{Manhattan}(x, \bar{x}) = \sum_i |x_i - \bar{x}_i|. \quad (2.22)$$

When comparing against DACE [58], we use the Mahalanobis distance, since this is the distance function used in their novel cost function (see Equation 2.27):

$$d_{Mahalanobis}(x, \bar{x}|C) = \sqrt{(x - \bar{x})C^{-1}(x - \bar{x})}. \quad (2.23)$$

$C$  is the covariance matrix of  $x$  and  $\bar{x}$ , which allows us to account for correlations between features. When all features are uncorrelated, the Mahalanobis distance is equal to the Euclidean distance.

### 2.5.4 Evaluation Metrics

We evaluate the counterfactual examples produced by FOCUS based on how close they are to the original input using three metrics, in terms of four distance functions (see Section 2.5.3). The first evaluation metric is distance from the original input averaged over all examples,  $d_{mean}$ . Let  $X$  be the set of  $N$  original instances and  $\bar{X}$  be the corresponding set of  $N$  generated counterfactual examples. The *mean distance* is defined as:

$$d_{mean}(X, \bar{X}) = \frac{1}{N} \sum_{n=1}^N d(x^{(n)}, \bar{x}^{(n)}). \quad (2.24)$$

The second evaluation metric is mean relative distance from the original input,  $d_{Rmean}$ . This metric helps us interpret individual improvements over the baselines; if  $d_{Rmean} < 1$ , FOCUS’s counterfactual examples are on average closer to the original input compared to the baseline. Let  $\bar{X}$  be the set of counterfactual examples produced by FOCUS and let  $\bar{X}'$  be the set of counterfactual examples produced by a baseline. Then the *mean relative distance* is defined as:

$$d_{Rmean}(\bar{X}, \bar{X}') = \frac{1}{N} \sum_{n=1}^N \frac{d(x^{(n)}, \bar{x}^{(n)})}{d(x^{(n)}, \bar{x}'^{(n)})}. \quad (2.25)$$

The third evaluation metric is the proportion of FOCUS’s counterfactual examples that are closer to the original input in comparison to the baselines. For  $d$  we consider Euclidean, Cosine, Manhattan, and Mahalanobis distance.

## 2.6 Experiment 1: FOCUS vs. FT

---

We compare FOCUS to the Feature Tweaking (FT) method by Tolomei et al. [132] in terms of the evaluation metrics in Section 2.5.4. We consider 36 experimental settings (4 datasets  $\times$  3 tree-based models  $\times$  3 distance functions) when comparing FOCUS to FT. The results are listed in Table 2.1.

### 2.6.1 Baseline: Feature Tweaking

FT identifies the leaf nodes where the prediction of the leaf nodes do not match the original prediction  $y_x$ : it recognizes the set of leaves that if activated,  $t_j(\bar{x}) = 1$ , would change the prediction of a tree  $\mathcal{T}$ :

$$\mathcal{T}_{change} = \left\{ j \mid j \in \mathcal{T}_{leaf} \wedge y_x \neq \arg \max_y T(y \mid j) \right\}. \quad (2.26)$$

For every  $\mathcal{T}$  in  $f$ , FT generates a perturbed example per node in  $\mathcal{T}_{change}$  so that it is activated with at least an  $\epsilon$  difference per threshold, and then selects the most optimal example (i.e., the one closest to the original instance). For every feature threshold  $\theta_j$  involved, the corresponding feature is perturbed



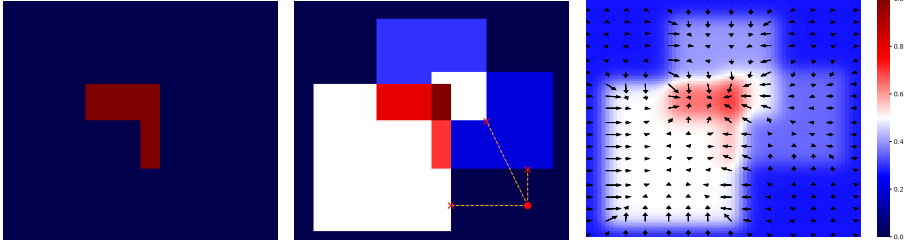


Figure 2.2: An example of how the FT baseline method (explained in Section 2.6.1) and our method handle an adaptive boosting ensemble with three trees. Left: decision boundary of the ensemble. Middle: three positive leaves that form the decision boundary, an example instance, and the perturbed examples suggested by FT. Right: approximated loss  $\tilde{\mathcal{L}}_{pred}$  and its gradient w.r.t.  $\bar{x}$ . The FT perturbed examples do not change the prediction of the forest, whereas the gradient of the differentiable approximation leads toward the true decision boundary.

accordingly:  $\bar{x}_{f_j} = \theta_j \pm \epsilon$ . The result is a perturbed example that was changed minimally to activate a leaf node in  $\mathcal{T}_{change}$ . In our experiments, we test  $\epsilon \in \{0.001, 0.005, 0.01, 0.1\}$ , and choose the  $\epsilon$  that minimizes the mean distance to the original input, while maximizing the number of counterfactual examples generated.

The main problem with FT is that the perturbed examples are not necessarily counterfactual examples, since changing the prediction of a single tree  $\mathcal{T}$  does not guarantee a change in the prediction of the full ensemble  $f$ . Figure 2.2 shows all three perturbed examples generated by FT for a single instance. In this case, none of the generated examples change the model prediction and therefore none are valid counterfactual examples.

Figure 2.2 shows how FOCUS and FT handle an adaptive boosting ensemble using a two-feature ensemble with three trees. On the left is the decision boundary for a standard tree ensemble; the middle visualizes the positive leaf nodes that form the decision boundary; on the right is the approximated loss  $\tilde{\mathcal{L}}_{pred}$  and its gradient w.r.t.  $\bar{x}$ . The gradients push features close to thresholds harder and in the direction of the decision boundary if  $\tilde{\mathcal{L}}$  is convex.

### 2.6.2 Results

In terms of  $d_{mean}$ , FOCUS outperforms FT in 20 settings while FT outperforms FOCUS in 8 settings. The difference in  $d_{mean}$  is not significant in the remaining 8 settings. In general, FOCUS outperforms FT in settings using Euclidean and Cosine distance because in each iteration, FOCUS perturbs many of the features by a small amount. Since FT perturbs only the features associated with an individual leaf, we expected that it would perform better for Manhattan distance but our results show that this is not the case. There is no clear winner between FT and FOCUS for Manhattan distance.

Table 2.1: Evaluation results for Experiment 1 comparing FOCUS and FT counterfactual examples. Significant improvements and losses over the baseline (FT) are denoted by ▼ and ▲, respectively ( $p < 0.05$ , two-tailed t-test,); ° denotes no significant difference; ⊗ denotes settings where the baseline cannot find a counterfactual example for every instance.

Dataset	Metric	Method	Euclidean			Cosine			Manhattan		
			DT	RF	AB	DT	RF	AB	DT	RF	AB
WINE	$d_{mean}$	FT	0.269	<b>0.174</b>	0.267 <sup>⊗</sup>	0.030	0.017	0.034 <sup>⊗</sup>	0.269	<b>0.223</b>	0.382 <sup>⊗</sup>
		FOCUS	<b>0.268</b> °	0.188▲	<b>0.188</b> ▼	<b>0.003</b> ▼	<b>0.008</b> ▼	<b>0.014</b> ▼	<b>0.268</b> °	0.312▲	<b>0.360</b> ▼
QUALITY	$d_{Rmean}$	FOCUS/FT	0.990	1.256	0.649	0.066	0.821	0.312	0.990	1.977	0.924
	% <sub>closer</sub>	FOCUS < FT	100%	21.0%	87.5%	100%	80.8%	95.1%	100%	5.4%	58.6%
HELOC	$d_{mean}$	FT	<b>0.120</b>	0.210	0.185	0.003	0.008	0.007	<b>0.135</b>	<b>0.278</b>	<b>0.198</b>
		FOCUS	0.133▲	<b>0.186</b> ▼	<b>0.136</b> ▼	<b>0.001</b> ▼	<b>0.002</b> ▼	<b>0.001</b> ▼	0.152▲	0.284°	0.203°
	$d_{Rmean}$	FOCUS/FT	1.169	0.942	0.907	0.303	0.285	0.421	1.252	1.144	1.364
	% <sub>closer</sub>	FOCUS < FT	16.6%	57.9%	71.9%	91.6%	91.5%	92.9%	51.3%	43.6%	24.2%
COMPAS	$d_{mean}$	FT	<b>0.082</b>	<b>0.075</b>	0.081	0.013	0.014	0.015	<b>0.086</b>	<b>0.078</b>	<b>0.085</b>
		FOCUS	0.092▲	0.079°	<b>0.076</b> ▼	<b>0.008</b> ▼	<b>0.011</b> ▼	<b>0.007</b> ▼	0.093▲	0.085°	0.090°
	$d_{Rmean}$	FOCUS/FT	1.162	1.150	1.062	0.473	0.965	0.539	1.182	1.236	1.155
	% <sub>closer</sub>	FOCUS < FT	29.4%	22.6%	44.8%	82.7%	68.0%	84.8%	65.8%	36.2%	66.9%
SHOPPING	$d_{mean}$	FT	<b>0.119</b>	0.028	0.126 <sup>⊗</sup>	<b>0.050</b>	0.027	0.131 <sup>⊗</sup>	<b>0.121</b>	0.030	0.142 <sup>⊗</sup>
		FOCUS	0.142▲	<b>0.025</b> ▼	<b>0.028</b> ▼	0.055▲	<b>0.013</b> ▼	<b>0.006</b> ▼	0.128°	<b>0.026</b> ▼	<b>0.046</b> ▼
	$d_{Rmean}$	FOCUS/FT	1.051	1.053	0.218	0.795	0.482	0.074	0.944	0.796	0.312
	% <sub>closer</sub>	FOCUS < FT	40.2%	36.1%	99.6%	44.4%	86.1%	99.5%	55.8%	81.9%	97.1%

We also see that FOCUS usually outperforms FT in settings using random forests and adaptive boosting, while the opposite is true for decision trees.

Overall, we find that FOCUS is effective and efficient for finding counterfactual explanations for tree-based models. Unlike the FT baseline, FOCUS finds valid counterfactual explanations for *every* instance across all settings. In the majority of tested settings, FOCUS’s explanations are substantial improvements in terms of distance to the original inputs, across all three metrics.

## 2.7 Experiment 2: FOCUS vs. DACE

The flexibility of FOCUS allows us to plug in our choice of differentiable distance function. To compare against DACE [58], we use the Mahalanobis distance for both (i) generation of FOCUS explanations, and (ii) evaluation in comparison to DACE, since this is the distance function used in the DACE loss function (see Equation 2.27 in Section 2.7.1).

We found two main limitations of DACE: (i) in all of our settings, it can only generate counterfactual examples for a subset of the test set, and (ii) it is limited by the size of the tree-based model. All hyperparameter settings are listed in the Appendix to this chapter.

### 2.7.1 Baseline: DACE

DACE generates counterfactual examples that account for the underlying data distribution through a novel cost function using Mahalanobis distance and a local outlier factor (LOF):

$$d_{DACE}(x, \bar{x}|X, C) = d_{Mahalanobis}^2(x, \bar{x}|C) + \lambda q_k(x, \bar{x}|X), \quad (2.27)$$

where  $C$  is the covariance matrix,  $q_k$  is the  $k$ -LOF [15],  $X$  is the training set, and  $\lambda$  is the trade-off parameter. The  $k$ -LOF measures the degree to which an instance is an outlier in the context of its  $k$ -nearest neighbors.<sup>1</sup> To generate counterfactual examples, DACE formulates the task as a mixed-integer linear optimization problem and uses the CPLEX Optimizer<sup>2</sup> to solve it. We refer the reader to the original paper for a more detailed overview of this cost function. The  $q_k$  term in the loss function penalizes counterfactual examples that are outliers, and therefore decreasing  $\lambda$  results in a greater number of counterfactual examples. In our experiments, we test  $\lambda \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$ , and choose the  $\lambda$  that minimizes the mean distance to the original input, while maximizing the number of counterfactual examples generated.

We were only able to run DACE on 6 out of our 12 models because the problem size is too large (i.e., there are too many model parameters for DACE) for the remaining 6 models when using the free Python API of CPLEX (the optimizer used in DACE). Specifically, we were unable to run DACE on the following settings:

<sup>1</sup>We use  $k = 1$  in our experiments, since this is the value supported in the original code.

<sup>2</sup><http://www.ibm.com/analytics/cplex-optimizer>

- Wine Quality AB (100 trees, max depth 4)
- Wine Quality RF (500 trees, max depth 4)
- HELOC RF (500 trees, max depth 4)
- HELOC AB (100 trees, max depth 8)
- COMPAS RF (500 trees, max depth 4)
- Shopping RF (500 trees, max depth 8).

Therefore, when comparing against DACE, we have 6 experimental settings (6 models  $\times$  1 distance function). We note that these are not unreasonable model sizes, and that unlike DACE, FOCUS can be applied to all 12 models (see Table 2.1).

### 2.7.2 Results

Table 2 shows the results for the 6 settings we could run DACE on. We were only able to run DACE on 6 out of our 12 models because the problem size is too large (i.e., DACE has too many model parameters) for the remaining 6 models when using the free Python API of CPLEX (the optimizer used in DACE). Therefore, when comparing against DACE, we have 6 experimental settings (6 models  $\times$  1 distance function).

We found that DACE can only generate counterfactual examples for a small subset of the test set, regardless of the  $\lambda$ -value, as opposed to FOCUS, which can generate counterfactual examples for the entire test set in all cases. To compute  $d_{mean}$ ,  $d_{Rmean}$ , and  $\%_{closer}$ , we compare FOCUS and DACE only on the instances for which DACE was able to generate a counterfactual example. We find that FOCUS significantly outperforms DACE in 5 out of 6 settings in terms of all three evaluation metrics, indicating that FOCUS explanations are indeed more minimal than those produced by DACE. FOCUS is also more reliable since (i) it is not restricted by model size, and (ii) it can generate counterfactual examples for all instances in the test set.

## 2.8 Discussion and Analysis

---

Figure 2.3 shows the mean Manhattan distance of the perturbed examples in each iteration of FOCUS, along with the proportion of perturbations resulting in valid counterfactual examples found for two datasets (we omit the others due to space considerations). These trends are indicative of all settings: the mean distance increases until a counterfactual example has been found for every  $x$ , after which the mean distance starts to decrease. This seems to be a result of the hinge-loss in FOCUS, which first prioritizes finding a valid counterfactual example (see Equation 2.1), then decreasing the distance between  $x$  and  $\bar{x}$ .

Table 2.2: Evaluation results for Experiment 2 comparing FOCUS and DACE counterfactual examples in terms of Mahalanobis distance. Significant improvements over the baseline are denoted by  $\blacktriangledown$  ( $p < 0.05$ , two-tailed t-test,).  $^\circ$  denotes no significant difference.

Metric	Method	WINE	HELOC	COMPAS		SHOPPING	
		DT	DT	DT	AB	DT	AB
$d_{mean}$	DACE	1.325	1.427	0.814	1.570	0.050	3.230
	FOCUS	<b>0.542<math>\blacktriangledown</math></b>	<b>0.810<math>\blacktriangledown</math></b>	<b>0.776<math>^\circ</math></b>	<b>0.636<math>\blacktriangledown</math></b>	<b>0.023<math>\blacktriangledown</math></b>	<b>0.303<math>\blacktriangledown</math></b>
$d_{Rmean}$	FOCUS / DACE	0.420	0.622	1.18	0.372	0.449	0.380
$\%_{closer}$	FOCUS < DACE	100%	94.5%	29.9%	96.1%	99.4%	90.8%
$\# CFs$	DACE	241	1342	842	700	362	448
$found$	FOCUS	1470	3138	1852	1852	3699	3699
$\# obs in$	<i>dataset</i>	1470	3138	1852	1852	3699	3699

### 2.8.1 Case Study: Credit Risk

As a practical example, we investigate what FOCUS explanations look like for individuals in the HELOC dataset. Here, the task is to predict whether or not an individual will default on their loan. This has consequences for loan approval: individuals who are predicted as defaulting will be denied a loan. For these individuals, we want to understand how they can change their profile such that they are approved. Given an individual who has been denied a loan from a bank, a counterfactual explanation could be:

*Your loan application has been denied. In order to have your loan application approved, you need to (i) increase your ExternalRiskEstimate score by 62, and (ii) decrease your NetFractionRevolvingBurden by 58.*

Figure 2.4 shows four counterfactual explanations generated using different distance functions for the same individual and same model. We see that the Manhattan explanation only requires a few changes to the individual’s profile, but the changes are large. In contrast, the individual changes in the Euclidean explanation are smaller but there are more of them. In settings where there are significant dependencies between features, the Cosine explanations may be preferred since they are based on perturbations that try to preserve the relationship between features. For instance, in the WINE QUALITY dataset, it would be difficult to change the amount of citric acid without affecting the pH level. The Mahalanobis explanations would be useful when it is important to take into account not only correlations between features, but also the training data distribution. This flexibility allows users to choose what kind of explanation is best suited for their problem.

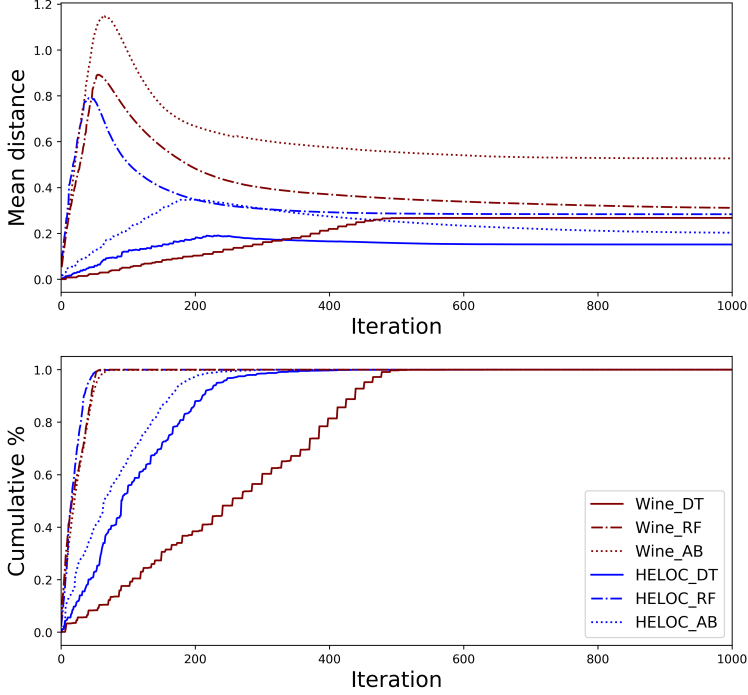


Figure 2.3: Mean distance (top) and cumulative % (bottom) of counterfactual examples in each iteration of FOCUS for Manhattan explanations.

Different distance functions can result in different *magnitudes* of feature perturbations as well as different *directions*. For example, the Cosine explanation suggests increasing *PercentTradesWBalance*, while the Mahalanobis explanations suggests decreasing it. This is because the loss space of the underlying RF model is highly non-convex, and therefore there is more than one way to obtain an alternative prediction. When using complex models such as tree ensembles, there are no monotonicity guarantees. In this case, both options result in valid counterfactual examples.

We examine the Manhattan explanation in more detail. We see that FOCUS suggests two main changes: (i) increasing the *ExternalRiskEstimate*, and (ii) decreasing the *NetFractionRevolvingBurden*. We obtain the definitions and expected trends from the data dictionary created by the authors of the dataset. The *ExternalRiskEstimate* is a “consolidated version of risk markers” (i.e., a credit score). A higher score is better: as one’s *ExternalRiskEstimate* increases, the probability of default decreases. The *NetFractionRevolvingBurden* is the “revolving balance divided by the credit limit” (i.e., utilization). A lower value is better: as one’s *NetFractionRevolvingBurden* increases, the probability of default increases. We find that the changes suggested by FOCUS are fairly consistent with the expected trends in the data dictionary, as opposed to suggesting non-

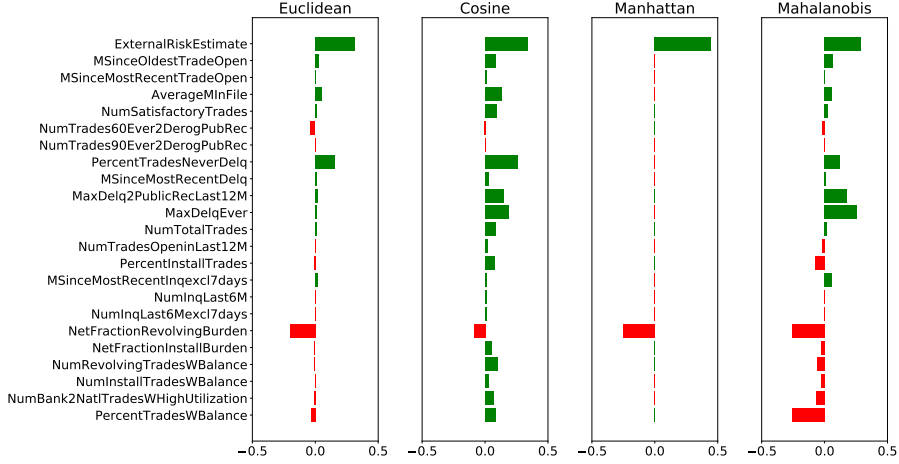


Figure 2.4: FOCUS explanations for the same model and same  $x$  from the HELOC dataset based on different distance functions. Green and red indicate increases and decreases in feature values, respectively. Perturbation values are based on normalized feature values. Left: Euclidean explanation perturbs several features, but only slightly. Middle Left: Cosine explanation perturbs almost all of the features. Middle Right: Manhattan explanation perturbs two features substantially. Right: Mahalanobis explanation perturbs almost all of the features.

sensical changes such as increasing one’s utilization to decrease the probability of default.

Decreasing one’s utilization is heavily dependent on the specific situation: an individual who only supports themselves might have more control over their spending in comparison to someone who has multiple dependents. An individual can decrease their utilization in two ways: (i) decreasing their spending, or (ii) increasing their credit limit (or a combination of the two). We can postulate that (i) is more “actionable” than (ii), since (ii) is usually a decision made by a financial institution. However, the degree to which an individual can actually change their spending habits is completely dependent on their specific situation: an individual who only supports themselves might have more control over their spending than someone who has multiple dependents. In either case, we argue that deciding what is (not) actionable is not a decision for the developer to make, but for the individual who is affected by the decision. Counterfactual examples should be used as part of a human-in-the-loop system and not as a final solution.

The individual should know that utilization is an important component of the model, even if it is not necessarily “actionable” for them. We also note that it is unclear how exactly an individual would change their credit score without further insight into how the score was calculated (i.e., how the risk markers were consolidated). It should be noted that this is not a shortcoming

of FOCUS, but rather of using features that are uninterpretable on their own, such as credit scores. Although FOCUS explanations cannot tell a user precisely how to increase their credit score, it is still important for the individual to know that their credit score is an important factor in determining their probability of getting a loan, as this empowers them to ask questions about how the score was calculated (i.e., how the risk markers were consolidated).

## 2.9 Conclusion

---

In this chapter, we propose an explanation method for tree-based classifiers, FOCUS, which casts the problem of finding counterfactual examples as a gradient-based optimization task and provides a differentiable approximation of tree-based models to be used in the optimization framework.

Given an input instance  $x$ , FOCUS generates an optimal counterfactual example based on the minimal perturbation to the input instance  $x$  which results in an alternative prediction from a model  $f$ . Unlike previous methods that assume the underlying classification model is differentiable, we propose a solution for when  $f$  is a non-differentiable, tree-based model that provides a differentiable approximation of  $f$ , which can be used to find counterfactual examples using gradient-based optimization techniques.

In the majority of experiments, examples generated by FOCUS are significantly closer to the original instances in terms of three different evaluation metrics compared to those generated by the baselines. FOCUS is able to generate valid counterfactual examples for all instances across all datasets, and the resulting explanations are flexible depending on the distance function.

This answers **RQ1**: we can generate counterfactual explanations for tree-based models using gradient-based optimization if we include differentiable approximations of tree-based models within the optimization framework. In the following chapter, we will investigate how to extend our method to accommodate different types of data such as graphs.

## Reproducibility

---

To facilitate the reproducibility of this work, our code is available at <https://github.com/a-lucic/focus>.



# Counterfactual Explanations for Graph Neural Networks

In the previous chapter, we developed a method for generating counterfactual explanations specific to tree-based models using gradient-based optimization techniques. In this chapter, we address the following research question:

**RQ2:** Can we extend our counterfactual explanation method for tree-based models to graph-based models?

Most existing methods for explaining predictions from graph neural networks (GNNs) are based on retrieving a subgraph of the original graph that is most relevant for the prediction. This differs from the counterfactual explanation problem where the task is to find the minimal perturbation to the original graph such that the prediction changes. The method we propose in this chapter is one of the first methods for generating counterfactual explanations for GNNs.

The answer to **RQ2** is yes: we first extend the counterfactual explanation problem formalization to the graph data setting, then apply the same gradient-based optimization techniques as in the previous chapter. Our experimental results show that our algorithm can reliably generate minimal and accurate counterfactual explanations for GNNs.

## 3.1 Introduction

Advances in machine learning (ML) have led to breakthroughs in several areas of science and engineering, ranging from computer vision, to natural language processing, to conversational assistants. Parallel to the increased performance of ML systems, there is an increasing call for the “understandability” of ML models [40]. Understanding *why* an ML model returns a certain output in response to a given input is important for a variety of reasons such as model debugging, aiding decision-making, or fulfilling legal requirements [32]. Having certified methods for interpreting ML predictions will help enable their use across a variety of applications [90].

Explainable artificial intelligence (XAI) refers to the set of techniques “*focused on exposing complex AI models to humans in a systematic and interpretable*

---

This chapter was published at the International Conference on Artificial Intelligence and Statistics (AISTATS 2022) under the title “CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks” [85].

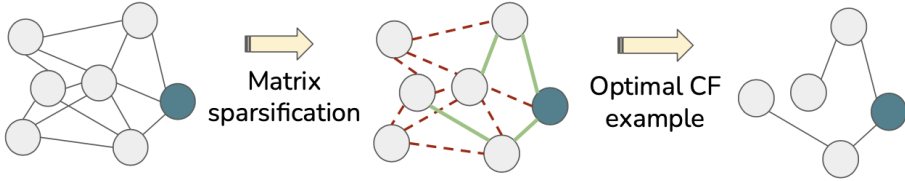


Figure 3.1: Intuition of counterfactual example generation by CF-GNNExplainer.

manner” [111]. A large body of work on XAI has emerged in recent years [14, 47]. Counterfactual explanations are used to explain predictions of individual instances in the form: “If X had been different, Y would not have occurred” [60, 114, 122]. Counterfactual explanations are based on counterfactual examples: modified versions of the input sample that result in an alternative output (i.e., prediction). If the proposed modifications are also *actionable*, this is referred to as achieving recourse [61, 136].

To motivate our problem, consider an ML application for computational biology: drug discovery is a task that involves generating new molecules that can be used for medicinal purposes [124, 143]. Given a candidate molecule, a GNN can predict if this molecule has a certain property that would make it effective in treating a particular disease [48, 97, 142]. If the GNN predicts it does not have this desirable property, counterfactual explanations can help identify the minimal change required such that the molecule is predicted to have this property. This could help not only inform the design of a new molecule that has this property, but also understand the molecular structures that contribute to this property.

Although GNNs have shown state-of-the-art results on tasks involving graph data [25, 150], existing methods for explaining the predictions of GNNs have primarily focused on generating subgraphs that are relevant for a particular prediction [6, 30, 74, 88, 103, 112, 140, 146, 148, 149]. However, *none of these methods are able to identify the minimal subgraph automatically* – they all require the user to specify the size of the subgraph,  $S$ , in advance. We show that even if we adapt existing methods to the counterfactual explanation problem, and try varying values for  $S$ , such methods are not able to produce valid, accurate counterfactual explanations, and are therefore not well-suited to solve the counterfactual explanation problem. To address this gap, we propose CF-GNNExplainer, a method for generating counterfactual explanations for GNNs.

Similar to other counterfactual methods for tabular or image data proposed in the literature [61, 139], CF-GNNExplainer works by perturbing input data at the instance-level. Unlike previous methods, CF-GNNExplainer can generate counterfactual explanations for graph data. In particular, our method iteratively removes edges from the original adjacency matrix based on matrix sparsification techniques, keeping track of the perturbation that leads to a change in prediction, and returning the perturbation with the smallest change w.r.t. the number of edges.

We evaluate CF-GNNExplainer on three public datasets for GNN explanations and measure its effectiveness using four metrics: fidelity, explanation size, sparsity, and accuracy. We find that CF-GNNExplainer is able to generate counterfactual examples with at least 94% accuracy, while removing fewer than 3 edges on average. We make the following contributions:

- (1) We formalize the problem of generating counterfactual explanations for GNNs (Section 3.4).
- (2) We propose CF-GNNExplainer, a novel method for explaining predictions from GNNs (Section 3.5).
- (3) We propose an experimental setup for holistically evaluating counterfactual explanations for GNNs (Section 3.6).

## 3.2 Related Work

---

Based on the taxonomy described in Chapter 1, our setting in this chapter is a *local explanation* problem for *neural networks*, specifically GNNs. We use *sensitivity analysis*, specifically counterfactual perturbations, on *graph* data to generate our explanations. Since our work is a counterfactual XAI approach for GNNs, it is related to GNN explainability (Section 3.2.1) as well as counterfactual explanations (Section 3.2.2). It is also related to adversarial attack methods (Section 3.2.3).

### 3.2.1 GNN Explainability

Several GNN XAI approaches have been proposed – a recent survey of the most relevant work is presented by Yuan et al. [148]. However, unlike our work, *none* of the methods in this survey generate counterfactual explanations.

The majority of existing GNN XAI methods provide an explanation in the form of a subgraph of the original graph that is deemed to be important for the prediction [6, 30, 74, 88, 103, 112, 140, 146, 148, 149]. We refer to these as *subgraph-generating methods*. Such methods are analogous to popular XAI methods such as LIME [107] or SHAP [86], which identify relevant features for a particular prediction for tabular, image, or text data. All of these methods require the user to specify the size of the explanation,  $S$ , in advance: the number of features (or edges) to keep. In contrast, CF-GNNExplainer generates counterfactual explanations, which can find the size of the explanation without requiring input from the user. Although both types of techniques are meant for explaining GNN predictions, they are solving fundamentally different problems: counterfactual explanations generate the minimal perturbation such that the prediction changes, while subgraph-retrieving methods identify a relevant (and not necessarily minimal) subgraph that matches the original prediction.

The work by Kang et al. [59] also generates counterfactual examples for GNNs, but they focus on a different task: link prediction. Other GNN XAI methods identify important node features [55] or similar examples [33]. The works of Yuan et al. [147] and Schnake et al. [113] generate model-level (i.e., global) explanations for GNNs, which differs from our work since we produce instance-level (i.e., local) explanations.

#### 3.2.2 Counterfactual Explanations

There exists a substantial body of work on counterfactual explanations for tabular, image, and text data [61, 122, 139]. Some methods treat the underlying classification model as a black-box [46, 70, 78], whereas others make use of the model’s inner workings [58, 84, 132, 136, 141]. All of these methods are based on perturbing feature values to generate counterfactual examples – they are not equipped to handle graph data with relationships (i.e., edges) between instances (i.e., nodes). In contrast, CF-GNNExplainer provides counterfactual examples specifically for graph data.

#### 3.2.3 Adversarial Attacks

Counterfactual examples are also related to adversarial attacks [126]: they both represent instances obtained from minimal perturbations to the input, which induce changes in the prediction made by the learned model. One difference between the two is in the intent: adversarial examples are meant to fool the model, while counterfactual examples are meant to explain the prediction [36, 84]. In the context of graph data, adversarial attack methods typically make minimal perturbations to the *overall graph* with the intention of degrading overall model performance, as opposed to attacking individual nodes. In contrast, we are interested in generating counterfactual examples for individual nodes, as opposed to identifying perturbations to the overall graph. We confirm that the counterfactual examples produced by CF-GNNExplainer are informative and not adversarial by measuring the accuracy of our method (see Section 3.6.3).

### 3.3 Background

---

In this section, we provide background information on GNNs (Section 3.3.1) and matrix sparsification (Section 3.3.2), both of which are necessary for understanding CF-GNNExplainer.

#### 3.3.1 Graph Neural Networks

Graphs are structures that represent a set of entities (nodes) and their relations (edges). GNNs operate on graphs to produce representations that can be used in downstream tasks such as graph or node classification. The latter is the focus of this work. We refer to the survey papers by Battaglia et al. [10] and Chami et al. [21] for an overview of existing GNN methods.

Let  $f(A, X; W) \rightarrow y$  be any GNN, where  $y$  is the set of possible predicted classes,  $A$  is an  $n \times n$  adjacency matrix,  $X$  is an  $n \times p$  feature matrix, and  $W$  is the learned weight matrix of  $f$ . In other words,  $A$  and  $X$  are the inputs of  $f$ , and  $f$  is parameterized by  $W$ .

A node’s representation is learned by iteratively updating the node’s features based on its neighbors’ features. The number of layers in  $f$  determines which neighbors are included: if there are  $\ell$  layers, then the node’s final representation only includes neighbors that are at most  $\ell$  hops away from that node in the graph  $\mathcal{G}$ . The rest of the nodes in  $\mathcal{G}$  are not relevant for the computation of the node’s final representation. We define the *subgraph neighborhood* of a node  $v$  as the set of the nodes and edges relevant for the computation of  $f(v)$  (i.e., those in the  $\ell$ -hop neighborhood of  $f$ ), represented as a tuple:  $\mathcal{G}_v = (A_v, X_v)$ , where  $A_v$  is the subgraph adjacency matrix and  $X_v$  is the node feature matrix for nodes that are at most  $\ell$  hops away from  $v$ . We then define a node  $v$  as a tuple of the form  $v = (A_v, x)$ , where  $x$  is the feature vector for  $v$ .

### 3.3.2 Matrix Sparsification

CF-GNNExplainer uses matrix sparsification to generate counterfactual examples, inspired by Srinivas et al. [121], who propose a method for training sparse neural networks. Given a weight matrix  $W$ , a binary sparsification matrix is learned which is multiplied element-wise with  $W$  such that some of the entries in  $W$  are zeroed out. In the work by Srinivas et al. [121], the objective is to remove entries in the weight matrix in order to reduce the number of parameters in the model. In our case, we want to *zero out entries in the adjacency matrix* (i.e., remove edges) in order to generate counterfactual explanations for GNNs. That is, we want to remove the important edges – those that are crucial for the prediction.

## 3.4 Problem Formulation

In general, a counterfactual example  $\bar{x}$  for an instance  $x$  according to a trained classifier  $f$  is found by perturbing the features of  $x$  such that  $f(x) \neq f(\bar{x})$  [141]. An optimal counterfactual example  $\bar{x}^*$  is one that minimizes the distance between the original instance and the counterfactual example, according to some distance function  $d$ . The resulting optimal counterfactual explanation is therefore  $\Delta_x^* = \bar{x}^* - x$  [84].

For graph data, it may not be enough to simply perturb node features, especially since they are not always available. This is why we are interested in generating counterfactual examples by perturbing the graph structure instead. In other words, we want to change the relationships between instances (i.e., nodes), rather than change the instances themselves. Therefore, a counterfactual example for graph data has the form  $\bar{v} = (\bar{A}_v, x)$ , where  $x$  is the feature vector and  $\bar{A}_v$  is a perturbed version of  $A_v$ , the adjacency matrix of the subgraph neighborhood of a node  $v$ .  $\bar{A}_v$  is obtained by removing some edges from  $A_v$ ,

such that  $f(v) \neq f(\bar{v})$ . Following Wachter et al. [141] and Lucic et al. [84], we generate counterfactual examples by minimizing a loss function of the form:

$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v} \mid d), \quad (3.1)$$

where  $v$  is the original node,  $f$  is the original model,  $g$  is the counterfactual model that generates  $\bar{v}$ , and  $\mathcal{L}_{pred}$  is a prediction loss that encourages  $f(v) \neq f(\bar{v})$ .  $\mathcal{L}_{dist}$  is a distance loss that encourages  $\bar{v}$  to be close to  $v$ , and  $\beta$  controls how important  $\mathcal{L}_{dist}$  is compared to  $\mathcal{L}_{pred}$ . We want to find  $\bar{v}^*$  that minimizes Equation 3.1: this is the optimal counterfactual example for  $v$ .

### 3.5 Method: CF-GNNExplainer

---

To solve the problem defined in Section 3.4, we propose CF-GNNExplainer, which generates  $\bar{v} = (\bar{A}_v, x)$  given a node  $v = (A_v, x)$ . Our method can operate on any GNN model  $f$ . To illustrate our method and avoid cluttered notation, let  $f$  be a standard, one-layer Graph Convolutional Network (GCN) [68] for node classification:

$$f(A, X; W) = \text{softmax} \left[ \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X W \right], \quad (3.2)$$

where  $\tilde{A} = A + I$ ,  $I$  is the identity matrix,  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  are entries in the degree matrix  $\tilde{D}$ ,  $X$  is the node feature matrix, and  $W$  is the weight matrix [68].

#### 3.5.1 Adjacency Matrix Perturbation

First, we define  $\bar{A}_v = P \odot A_v$ , where  $P$  is a binary perturbation matrix that sparsifies  $A_v$ . Our aim is to find  $P$  for a given node  $v$  such that  $f(A_v, x) \neq f(P \odot A_v, x)$ . To find  $P$ , we build upon the method by Srinivas et al. [121] for training sparse neural networks (see Section 3.3.2), where our objective is to zero out entries in the adjacency matrix (i.e., remove edges). That is, we want to find  $P$  that minimally perturbs  $A_v$ , and use it to compute  $\bar{A}_v = P \odot A_v$ . If an element  $P_{i,j} = 0$ , this results in the deletion of the edge between node  $i$  and node  $j$ . When  $P$  is a matrix of ones, this indicates that all edges in  $A_v$  are used in the forward pass.

Similar to the work by Srinivas et al. [121], we first generate an intermediate, real-valued matrix  $\hat{P}$  with entries in  $[0, 1]$ , apply a sigmoid transformation, then threshold the entries to arrive at a binary  $P$ : entries greater than or equal to 0.5 become 1, while those below 0.5 become 0. In the case of undirected graphs (i.e., those with symmetric adjacency matrices), we first generate a perturbation vector, which we then use to populate  $\hat{P}$  in a symmetric manner, instead of generating  $\hat{P}$  directly.

### 3.5.2 Counterfactual Generating Model

We want our perturbation matrix  $P$  to only act on  $A_v$ , not  $\tilde{A}_v$ , in order to preserve self-loops in the message passing of  $f$ . This is because we always want a node representation update to include its own representation from the previous layer. Therefore we first rewrite Equation 3.2 for our illustrative one-layer case to isolate  $A_v$ :

$$f(A_v, X_v; W) = \text{softmax} \left[ (D_v + I)^{-1/2} (A_v + I) (D_v + I)^{-1/2} X_v W \right]. \quad (3.3)$$

To generate CFs, we propose a new function  $g$ , which is based on  $f$ , but it is parameterized by  $P$  instead of  $W$ . We update the degree matrix  $D_v$  based on  $P \odot A_v$ , add the identity matrix to account for self-loops (as in  $\tilde{D}_v$  in Equation 3.2), and call this  $\bar{D}_v$ :

$$g(A_v, X_v, W; P) = \text{softmax} \left[ \bar{D}_v^{-1/2} (P \odot A_v + I) \bar{D}_v^{-1/2} X_v W \right]. \quad (3.4)$$

In other words,  $f$  learns the weight matrix while holding the data constant, while  $g$  generates new data points (i.e., counterfactual examples) while holding the weight matrix (i.e., model) constant. Another distinction between  $f$  and  $g$  is that the aim of  $f$  is to find the optimal set of weights that generalizes well on an unseen test set, while the objective of  $g$  is to generate an optimal counterfactual example, given a particular node (i.e.,  $\bar{v}$  is the output of  $g$ ).

### 3.5.3 Loss Function Optimization

We generate  $P$  by minimizing Equation 3.1, adopting the negative log-likelihood (NLL) loss for  $\mathcal{L}_{pred}$ :

$$\mathcal{L}_{pred}(v, \bar{v} | f, g) = \mathbb{1}[f(v) = f(\bar{v})] \cdot \mathcal{L}_{NLL}(f(v), g(\bar{v})). \quad (3.5)$$

Since we do not want  $f(\bar{v})$  to match  $f(v)$ , we put a negative sign in front of  $\mathcal{L}_{pred}$  and include an indicator function to ensure the loss is active as long as  $f(\bar{v}) = f(v)$ . Note that  $f$  and  $g$  have the same weight matrix  $W$  – the main difference is that  $g$  also includes the perturbation matrix  $P$ .

$\mathcal{L}_{dist}$  can be based on any differentiable distance function. In our case, we take  $d$  to be the element-wise difference between  $v$  and  $\bar{v}$ , corresponding to the difference between  $A_v$  and  $\bar{A}_v$ : the number of edges removed. For undirected graphs, we divide this value by 2 to account for the symmetry in the adjacency matrices. When updating  $P$ , we take the gradient of Equation 3.1 with respect to the intermediate  $\hat{P}$ , not the binary  $P$ .

**Algorithm 3.1** CF-GNNExplainer: given a node  $v = (A_v, x)$  where  $f(v) = y$ , generate the minimal perturbation,  $\bar{v} = (\bar{A}_v, x)$ , such that  $f(\bar{v}) \neq y$ .

---

**Input:** node  $v = (A_v, x)$ , trained GNN model  $f$ , CF model  $g$ , loss function  $\mathcal{L}$ , learning rate  $\alpha$ , number of iterations  $K$ , distance function  $d$ .

$f(v) = y$       *# Get GNN prediction*  
 $\hat{P} \leftarrow J_n$       *# Initialization*  
 $\bar{v}^* = []$

**for**  $K$  iterations **do**

$\bar{v} = \text{GET\_CF\_EXAMPLE}()$   
     $\mathcal{L} \leftarrow \mathcal{L}(v, \bar{v}, f, g)$       *# Eq 3.1 & 3.5*  
     $\hat{P} \leftarrow \hat{P} + \alpha \nabla_{\hat{P}} \mathcal{L}$       *# Update  $\hat{P}$*

**Function** GET\\_CF\\_EXAMPLE()

$P \leftarrow \text{threshold}(\sigma(\hat{P}))$   
     $\bar{A}_v \leftarrow P \odot A_v$   
     $\bar{v}_{cand} \leftarrow (\bar{A}_v, x)$   
    **if**  $f(v) \neq f(\bar{v}_{cand})$  **then**  
         $\bar{v} \leftarrow \bar{v}_{cand}$   
        **if not**  $\bar{v}^*$  **then**  
             $\bar{v}^* \leftarrow \bar{v}$       *# First CF*  
        **else if**  $d(v, \bar{v}) \leq d(v, \bar{v}^*)$  **then**  
             $\bar{v}^* \leftarrow \bar{v}$       *# Best CF*  
    **return**  $\bar{v}^*$

---

### 3.5.4 CF-GNNExplainer

We call our method CF-GNNExplainer and summarize its details in Algorithm 3.1. Given an node in the test set  $v$ , we first obtain its original prediction from  $f$  and initialize  $\hat{P}$  as a matrix of ones,  $J_n$ , to initially retain all edges. Next, we run CF-GNNExplainer for  $K$  iterations. To find a counterfactual example, we use Equation 3.4.

First, we compute  $P$  by thresholding  $\hat{P}$  (see Section 3.5.1). Then we use  $P$  to obtain the sparsified adjacency matrix that gives us a candidate counterfactual example,  $\bar{v}_{cand}$ . This example is then fed to the original GNN,  $f$ , and if  $f$  predicts a different output than for the original node, we have found a valid counterfactual example,  $\bar{v}$ .

We keep track of the “best” counterfactual example (i.e., the most minimal according to  $d$ ), and return this as the optimal counterfactual example  $\bar{v}^*$  after  $K$  iterations. Between iterations, we compute the loss following Equations 3.1 and 3.5, and update  $\hat{P}$  based on the gradient of the loss. In the end, we retrieve the optimal counterfactual explanation  $\Delta_v^* = v - \bar{v}^*$ .



### 3.5.5 Complexity

CF-GNNExplainer has time complexity  $O(KN^2)$ , where  $N$  is the number of nodes in the subgraph neighbourhood and  $K$  is the number of iterations. We note that high complexity is common for local XAI methods (i.e., SHAP [86], GNNExplainer [146], etc.), but in practice, one typically only generates explanations for a subset of the dataset.

## 3.6 Experimental Setup

In this section, we outline our experimental setup for evaluating CF-GNNExplainer, including the datasets and models used (Section 3.6.1), the baselines we compare against (Section 3.6.2), the evaluation metrics (Section 3.6.3), and the hyperparameter search method (Section 3.6.4). In total, we run approximately 375 hours of experiments on one Nvidia TitanX Pascal GPU with access to 12GB RAM.

### 3.6.1 Datasets and Models

Given the challenges associated with defining and evaluating the accuracy of XAI methods [29], we first focus on synthetic tasks where we know the ground-truth explanations. Although there exist real graph classification datasets with ground-truth explanations [26], there do not exist any real node classification datasets with ground-truth explanations, which is the task we focus on in this chapter. Building such a dataset would be an excellent contribution, but is outside the scope of this paper.

In our experiments, we use the TREE-CYCLES, TREE-GRIDS, BA-SHAPES datasets from the work by Ying et al. [146]. These datasets were created specifically for the task of explaining node classification predictions from GNNs. Each dataset consists of (i) a base graph, (ii) motifs that are attached to random nodes of the base graph, and (iii) additional edges that are randomly added to the overall graph. They are all undirected graphs. The classification task is to determine whether or not the nodes are part of the motif. The purpose of these datasets is to have a ground-truth for the “correctness” of an explanation: for nodes in the motifs, the explanation is the motif itself [88]. The dataset statistics are available in Table 3.1.

TREE-CYCLES consists of a binary tree base graph with 6-cycle motifs, TREE-GRIDS also has a binary tree as its base graph, but with  $3 \times 3$  grids as the motifs. For BA-SHAPES, the base graph is a Barabasi-Albert (BA) graph with house-shaped motifs, where each motif consists of 5 nodes (one for the top of the house, two in the middle, and two on the bottom). Here, there are four possible classes (not in motif, in motif: top, middle, bottom). We note that compared to the other two datasets, the BA-SHAPES dataset is much more densely connected – the node degree is more than twice as high as that of the TREE-CYCLES or TREE-GRID datasets, and the average number of nodes and edges in each node’s computation graph is order(s) of magnitude larger. We use the same experimental setup (i.e.,

Table 3.1: Dataset statistics. The # edges in the motif indicates the size of the ground truth (GT) explanation.

	TREE CYCLES	TREE GRID	BA SHAPES
# classes	2	2	4
# nodes in motif	6	9	5
# edges in motif (GT)	6	12	6
# nodes in total	871	1231	700
# edges in total	1950	3410	4100
Avg node degree	2.27	2.77	5.87
Avg # nodes in $A_v$	19.12	30.69	304.40
Avg # edges in $A_v$	18.99	33.94	1106.24

dataset splits, model architecture) as Ying et al. [146] to train a 3-layer GCN (hidden size = 20) for each task. Our GCNs have at least 87% accuracy on the test set.

### 3.6.2 Baselines

Since existing GNN XAI methods give explanations in the form of relevant subgraphs as opposed to counterfactual examples, it is not straightforward to identify baselines for our experiments that ensure a fair comparison between methods. To evaluate CF-GNNExplainer, we compare against 4 baselines: RANDOM, 1HOP, RM-1HOP, and GNNEXPLAINER. The random perturbation is meant as a sanity check. We randomly initialize the entries of  $\hat{P} \in [-1, 1]$  and apply the same sigmoid transformation and thresholding as described in Section 3.5.1. We repeat this  $K$  times and keep track of the most minimal perturbation resulting in a counterfactual example. Next, we compare against baselines that are based on the ego graph of  $v$  (i.e., its 1-hop neighbourhood): 1HOP keeps all edges in the ego graph of  $v$ , while RM-1HOP removes all edges in the ego graph of  $v$ .

Our fourth baseline is based on GNNEXPLAINER by Ying et al. [146], which identifies the  $S$  most relevant edges for the prediction (i.e., the most relevant subgraph of size  $S$ ). To generate counterfactual explanations, we remove the subgraph generated by GNNEXPLAINER. We include this method in our experiments in order to have a baseline based on a prominent GNN XAI method, but we note that subgraph-retrieving methods like GNNEXPLAINER are not meant for generating counterfactual explanations. Unlike our method, GNNEXPLAINER cannot automatically find a *minimal* subgraph and therefore requires the user to determine the number of edges to keep in advance (i.e., the value of  $S$ ). As a result, we cannot evaluate how minimal its counterfactual explanations are, but we can compare it against our method in terms of (i) its ability to generate valid counterfactual examples, and (ii) how accurate those counterfactual examples

are. We report results on GNNEXPLAINER for  $S \in \{1, 2, 3, 4, 5, \text{GT}\}$ , where GT is the size of the ground truth explanation (i.e., the number of edges in the motif, see Table 3.1).

### 3.6.3 Metrics

We generate a counterfactual example for each node in the graph separately and evaluate in terms of four metrics.

**Fidelity:** is defined as the proportion of nodes where the original predictions match the prediction for the explanations [92, 107]. Since we generate counterfactual examples, we do not want the original prediction to match the prediction for the explanation, so we want a low value for fidelity.

**Explanation Size:** is the number of removed edges. It corresponds to the  $\mathcal{L}_{dist}$  term in Equation 3.1: the difference between the original  $A_v$  and the counterfactual  $\hat{A}_v$ . Since we want to have *minimal* explanations, we want a small value for this metric. Note that we cannot evaluate this metric for GNNEXPLAINER.

**Sparsity:** measures the proportion of edges in  $A_v$  that are removed [148]. A value of 0 indicates all edges in  $A_v$  were removed. Since we want *minimal* explanations, we want a value close to 1. Note that we cannot evaluate this metric for GNNEXPLAINER.

**Accuracy:** is the mean proportion of explanations that are “correct”. Following the work by Luo et al. [88], Ying et al. [146], we only compute accuracy for nodes that are originally predicted as being part of the motifs, since accuracy can only be computed on instances for which we know the ground truth explanations. Given that we want *minimal* explanations, we consider an explanation to be correct if it *exclusively* involves edges that are inside the motifs (i.e., only removes edges that are within the motifs).

### 3.6.4 Hyperparameter Search

We experiment with different optimizers and hyperparameter values for the number of iterations  $K$ , the trade-off parameter  $\beta$ , the learning rate  $\alpha$ , and the Nesterov momentum  $m$  (when applicable). We choose the setting that produces the most counterfactual examples. We test the number of iterations  $K \in \{100, 300, 500\}$ , the trade-off parameter  $\beta \in \{0.1, 0.5\}$ , the learning rate  $\alpha \in \{0.005, 0.01, 0.1, 1\}$ , and the Nesterov momentum  $m \in \{0, 0.5, 0.7, 0.9\}$ . We test Adam, SGD and AdaDelta as optimizers. We find that for all three datasets, the SGD optimizer gives the best results, with  $k = 500$ ,  $\beta = 0.5$ , and  $\alpha = 0.1$ . For the TREE-CYCLES and TREE-GRID datasets, we set  $m = 0$ , while for the BA-SHAPES dataset, we use  $m = 0.9$ .

## 3.7 Results

We evaluate CF-GNNExplainer in terms of the metrics outlined in Section 3.6.3. The results are shown in Table 3.2 and Table 3.3. In cases where the baselines outperform CF-GNNExplainer on a particular metric, they perform poorly on the rest of the metrics, or on other datasets.

### 3.7.1 Main Findings

**Fidelity:** CF-GNNExplainer outperforms 1HOP across all three datasets, and outperforms RM-1HOP for TREE-CYCLES and TREE-GRID in terms of fidelity. We find that RANDOM has the lowest fidelity in all cases – it is able to find counterfactual examples for every single node. In the following subsections, we will see that this corresponds to poor performance on the other metrics.

Table 3.2: Results comparing our method (abbreviated as CF-GNN) to RANDOM, 1HOP, and RM-1HOP. Below each metric, ▼ indicates a low value is desirable, while ▲ indicates a high value is desirable.

Method	TREE-CYCLES				TREE-GRID				BA-SHAPES			
	Fid. ▼	Size ▼	Spar. ▲	Acc. ▲	Fid. ▼	Size ▼	Spar. ▲	Acc. ▲	Fid. ▼	Size ▼	Spar. ▲	Acc. ▲
RANDOM	<b>0.00</b>	4.70	0.79	0.63	<b>0.00</b>	9.06	0.75	0.77	<b>0.00</b>	503.31	0.58	0.17
1HOP	0.32	15.64	0.13	0.45	0.32	29.30	0.09	0.72	0.60	504.18	0.05	0.18
RM-1HOP	0.46	2.11	0.89	—	0.61	2.27	0.92	—	0.21	10.56	0.97	<b>0.99</b>
CF-GNN	0.21	<b>2.09</b>	<b>0.90</b>	<b>0.94</b>	0.07	<b>1.47</b>	<b>0.94</b>	<b>0.96</b>	0.39	<b>2.39</b>	<b>0.99</b>	0.96

**Explanation Size:** Figures 3.2 to 3.5 show histograms of the explanation size for CF-GNNExplainer and the baselines. We see that across all three datasets, CF-GNNExplainer has the smallest (i.e., most minimal) explanation sizes. This is especially true when comparing to RANDOM and 1HOP for the BA-SHAPES dataset, where we had to use a different scale for the  $x$ -axis due to how different the explanation sizes were. We postulate that this difference could be because BA-SHAPES is a much more densely connected graph; it has fewer nodes but more edges compared to the other two datasets, and the average number of nodes and edges in the subgraph neighborhood is order(s) of magnitude larger (see Table 3.1). Therefore, when performing random perturbations, there is substantial opportunity to remove edges that do not necessarily need to be removed, leading to much larger explanation sizes. When there are many edges in the subgraph neighborhood, removing everything except the 1-hop neighbourhood, as is done in 1HOP, also results in large explanation sizes. In contrast, the loss function used by CF-GNNExplainer ensures that only a few edges are removed, which is the desirable behavior since we want minimal explanations.

Table 3.3: Results comparing our method to GNNEXPLAINER. GNNEXPLAINER cannot find  $S$  automatically, so we try varying values of  $S$ . GT indicates the size of the ground truth explanation for each dataset. CF-GNNExplainer finds  $S$  automatically. Below each metric, ▼ indicates a low value is desirable, while ▲ indicates a high value is desirable.

GNNEXP	TREE-CYCLES				TREE-GRID				BA-SHAPES			
	Fid.	Size	Spars.	Acc.	Fid.	Size	Spars.	Acc.	Fid.	Size	Spars.	Acc.
	▼	▼	▲	▲	▼	▼	▲	▲	▼	▼	▲	▲
$S = 1$	0.65	1.00	0.92	0.61	0.69	1.00	0.96	0.79	0.90	1.00	0.94	0.52
$S = 2$	0.59	2.00	0.85	0.54	0.51	2.00	0.92	0.78	0.85	2.00	0.91	0.40
$S = 3$	0.56	3.00	0.79	0.51	0.46	3.00	0.88	0.79	0.83	3.00	0.87	0.34
$S = 4$	0.58	4.00	0.72	0.48	0.42	4.00	0.84	0.79	0.83	4.00	0.83	0.31
$S = 5$	0.57	5.00	0.66	0.46	0.40	5.00	0.80	0.79	0.81	5.00	0.81	0.27
$S = \text{GT}$	0.55	6.00	0.57	0.46	0.35	11.83	0.53	0.74	0.82	6.00	0.79	0.24
CF-GNN	<b>0.21</b>	2.09	0.90	<b>0.94</b>	<b>0.07</b>	1.47	0.94	<b>0.96</b>	<b>0.39</b>	2.39	0.99	<b>0.96</b>

**Sparsity:** CF-GNNExplainer outperforms the RANDOM, RM-1HOP, 1HOP baselines for all three datasets in terms of sparsity. We note CF-GNNExplainer and RM-1HOP perform much better on this metric in comparison to the other methods, which aligns with the results from explanation size.

**Accuracy:** We observe that CF-GNNExplainer has the highest accuracy for the TREE-CYCLES and TREE-GRID datasets, whereas RM-1HOP has the highest accuracy for BA-SHAPES. However, we are unable to calculate the accuracy of RM-1HOP for the other two datasets since it is unable to generate *any* counterfactual examples for motif nodes, contributing to the low sparsity on those datasets. We observe accuracy levels upwards of 94% for CF-GNNExplainer across *all* datasets, indicating that it is consistent in correctly removing edges that are crucial for the initial predictions in the vast majority of cases (see Table 3.2).

### 3.7.2 Comparison to GNNExplainer

Table 3.3 shows the results comparing our method to GNNEXPLAINER. We find that our method outperforms GNNEXPLAINER across all three datasets in terms of both fidelity and accuracy, for all tested values of  $S$ . However, this is not surprising since GNNEXPLAINER is not meant for generating counterfactual explanations, so we cannot expect it to perform well on a task it was not designed for. We cannot compare explanation size or sparsity fairly since GNNEXPLAINER requires the user to input the value of  $S$ .

### 3.7.3 Summary of Results

Evaluating on four distinct metrics for each dataset gives us a more holistic view of the results. We find that across all three datasets, CF-GNNExplainer can generate counterfactual examples for the majority of nodes in the test set (i.e., low fidelity), while only removing a small number of edges (i.e., low explanation

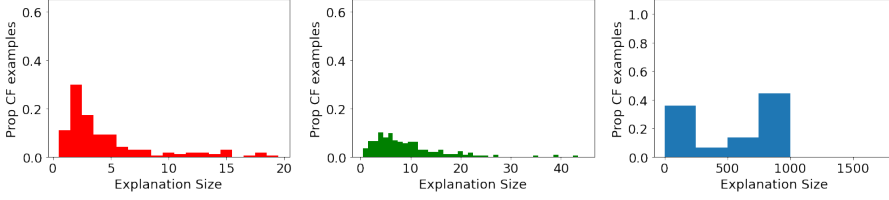


Figure 3.2: Histograms showing the proportion of counterfactual examples that have a certain explanation size from RANDOM. Note the  $x$ -axis for BA-SHAPES goes up to 1500. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

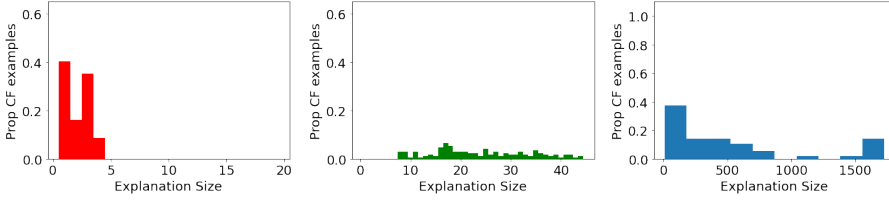


Figure 3.3: Histograms showing the proportion of counterfactual examples that have a certain explanation size from 1HOP. Note the  $x$ -axis for BA-SHAPES goes up to 1500. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

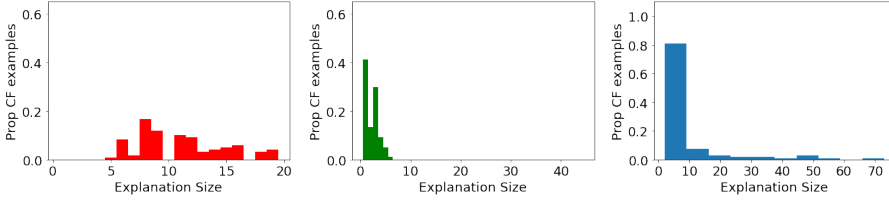


Figure 3.4: Histograms showing the proportion of counterfactual examples that have a certain explanation size from RM-1HOP. Note the  $x$ -axis for BA-SHAPES goes up to 70. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

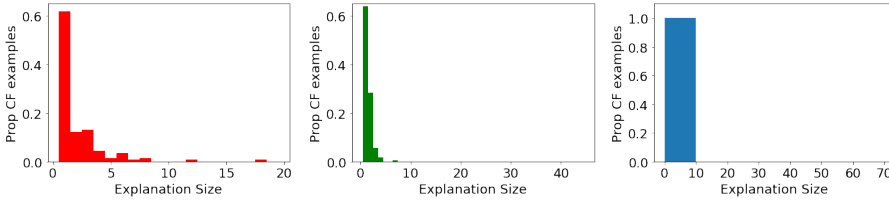


Figure 3.5: Histograms showing the proportion of counterfactual examples that have a certain explanation size from CF-GNNExplainer. Note the  $x$ -axis for BA-SHAPES goes up to 70. Left: TREE-CYCLES, Middle: TREE-GRID, Right: BA-SHAPES.

size, high sparsity). For nodes where we know the ground truth (i.e., those in the motifs) we achieve at least 94% accuracy.

Although RANDOM can generate counterfactual examples for every node, they are not very minimal or accurate. The latter is also true for 1HOP – in general, it has the worst scores for explanation size, sparsity and accuracy. GNNEXPLAINER performs at a similar level as 1HOP, indicating that although it is a prominent GNN XAI method, it is not well-suited for solving the counterfactual explanation problem.

RM-1HOP is competitive in terms of explanation size, but it performs poorly in terms of fidelity for the TREE-CYCLES and TREE-GRID datasets, and its accuracy on these datasets is unknown since it is unable to generate *any* counterfactual examples for nodes in the motifs. These results show that our method is simple and effective in solving the counterfactual explanation task, unlike the baselines we test.

## 3.8 Conclusion

---

In this chapter, we propose CF-GNNExplainer, a method for generating counterfactual explanations for any GNN. Our simple and effective method is able to generate counterfactual explanations that are (i) minimal, both in terms of the absolute number of edges removed (explanation size), as well as the proportion of the subgraph neighborhood that is perturbed (sparsity), and (ii) accurate, in terms of removing edges that we know to be crucial for the initial predictions.

We evaluate our method on three commonly used datasets for GNN explanation tasks and find that these results hold across all three datasets. We find that existing GNN XAI methods are not well-suited to solving the counterfactual explanation task, while CF-GNNExplainer is able to reliably produce minimal, accurate counterfactual explanations.

This answers **RQ2**: we can generate counterfactual explanations for graph-based models by extending the problem formalization from Chapter 2 to accommodate graph data. We do so by introducing a perturbation matrix that acts on the adjacency matrix to remove edges in the graph, then applying similar gradient-based optimization techniques as in Chapter 2 for each instance in the dataset. In the following chapter, we will investigate how to generate explanations that are specific to a particular real-world use case and evaluate them on real users.

## Reproducibility

---

To facilitate the reproducibility of the work in this chapter, our code is available at <https://github.com/a-lucic/cf-gnnexplainer>.





## Part II

# Users



# Contrastive Explanations for Forecasting Errors

---

In this second part of the thesis, we shift our focus to the consumers of explanations: users. In this chapter, we address the following research question:

**RQ3:** Given a real-world use case, can we create an explanation method based on this use case and evaluate it in a context-specific manner?

The answer to **RQ3** is yes: we first identify a use case where users seek explanations: understanding errors in sales forecasting. We then design an algorithm that generates explanations for large errors in forecasting predictions. We evaluate our method through a user study with both researchers and practitioners to investigate the impact our explanations have on users’ (i) objective abilities to understand the model’s predictions and (ii) subjective attitudes towards the model. Our experimental results show that explanations generated by our method help both types of users understand why large errors in predictions occur, but do not have an impact on their trust or confidence in the model.

## 4.1 Introduction

---

As more and more decisions about humans are made by machines, it becomes imperative to understand how these outputs are produced and what drives a model to a particular prediction [108]. As a result, algorithmic interpretability has gained significant interest and traction in the machine learning (ML) community over the past few years [29]. However, there exists considerable skepticism outside of the ML community due to a perceived lack of transparency behind algorithmic predictions, especially when errors are produced [28]. We aim to evaluate the effect of explaining model outputs, specifically large errors, on users’ attitudes towards trusting and deploying complex, automatically learned models.

Further motivation for explainable ML is provided by significant societal developments. Important examples include the recently enacted European General Data Protection Regulation (GDPR), which specifies that individuals will have the right to “the logic involved in any automatic personal data processing” [32]. In Canada and the United States, this right to an explanation is an integral

---

This chapter was published at the ACM Conference on Fairness, Accountability, and Transparency (FAccT 2020) under the title “Why Does My Model Fail: Contrastive Explanations for Retail Forecasting” [78], where it won a best paper award.

part of financial regulations, which is why banks have not been able to use high-performing “black-box” models to evaluate the credit-worthiness of their customers. Instead, they have been confined to easily interpretable algorithms such as decision trees (for segmenting populations) and logistic regression (for building risk scorecards) [66]. At NeurIPS 2017, the Explainable ML Challenge was launched to combat this limitation, indicating the finance industry’s interest in exploring algorithmic explanations [34].

We use explanations as a mechanism for supporting innovation and technological development while keeping the human “in the loop” by focusing on predictive modeling as a tool that aids individuals with a given task. Specifically, our interest lies with interpretability in a scenario where users with varying degrees of ML expertise are confronted with large errors in the outcome of predictive models. We focus on explaining large errors because people tend to be more curious about unexpected outcomes rather than ones that confirm their prior beliefs [53].

However, Dietvorst et al. [28] show that when users are confronted with errors in algorithmic predictions, they are less likely to use the model. Seeing an algorithm make mistakes significantly decreases confidence in the model, and users are more likely to choose a human forecaster instead, even after seeing the algorithm outperform the human [28]. This indicates that prediction mistakes have a significant impact on users’ perception of the model. By focusing on explaining mistakes, we hope to give insight into this phenomenon of algorithm aversion while also giving users the types of explanations they are interested in seeing.

Our work was motivated by the needs of analysts working on sales forecasting at Ahold Delhaize, a large retailer in the Netherlands. Current models in production are based on simple autoregressive methods, but there is an interest in exploring more complex techniques. However, the added complexity comes at the expense of interpretability, which is problematic for the retailer, especially when a complex model produces a forecast that is very different from the actual target value. This leads us to focus on explaining errors in regression predictions in this work. However, it should be noted that our method can be extended to classification predictions by defining “distances” between classes or by simply defining all errors as large errors.

We focus on two aspects of explainability in this scenario: the *generation* of explanations of large errors and the corresponding *effectiveness* of these explanations. Prior methods for generating explanations fail at generating explanations for large errors because they produce similar explanations for predictions resulting in large errors and those resulting in reasonable predictions (see Table 4.2 in Section 4.4 for an example). We propose a method for explaining large prediction errors, called Monte Carlo Bounds for Reasonable Predictions (MC-BRP), that shows users:

- (i) The required bounds of the most important features in order to have a prediction resulting in a reasonable prediction.
- (ii) The relationship between each of these features and the target.

It should be noted that in this chapter, we focus on explaining errors *in hindsight*, that is, we examine large errors once they have occurred and are not predicting them in advance without having access to the ground truth. We are also not using these explanations to improve the model, but rather examine the effectiveness of explaining large errors via MC-BRP on users' trust in the model and attitudes towards deploying it, as well as their understanding of the explanations. We test on a wide range of users, including both practitioners and researchers, and analyze the differences in attitudes between these users. We also reflect on the process of conducting a user study by outlining some limitations of our study and make some recommendations for future work. We address the following research questions:

**RQ3.1:** *Are the contrastive explanations generated by MC-BRP about large errors in predictions (i) interpretable, or (ii) actionable?* More specifically,

- (i) Can contrastive explanations about large errors give users enough information to simulate the model's output (forward simulation)?
- (ii) Can such explanations help users understand the model such that they can manipulate an observation's input values in order to change the output (counterfactual simulation)?

**RQ3.2:** *How does providing contrastive explanations generated by MC-BRP for large errors impact users' perception of the model?* Specifically, we want to investigate the following:

- (i) Does being provided with contrastive explanations generated by MC-BRP impact users' understanding of why the model produces errors?
- (ii) Does it impact their willingness to deploy the model?
- (iii) Does it impact their level of trust in the model?
- (iv) Does it impact their confidence in the model's performance?

Consequently, we make the following contributions:

- We contribute a method, MC-BRP, for generating contrastive explanations specifically for large errors in regression tasks.
- We evaluate our explanations through a user study with 75 participants in both objective and subjective terms.
- We conduct an analysis on the differences in attitudes between practitioners and researchers.

In Section 4.2 we discuss related work and identify how our problem relates to the current literature. In Section 4.3 we formally describe the methodology of explanations based on MC-BRP and in Section 4.4 we describe our experimental setup. In Section 4.5 we detail the results of the user study; we conduct further analyses in Section 4.6. In Section 4.7 we conclude and make recommendations for future work.

## 4.2 Related Work

---

Based on the taxonomy described in Chapter 1, our setting in this chapter is an *local explanation* problem. Our method is model-agnostic in nature, but we evaluate it specifically on *tree ensembles*. We use *sensitivity analysis*, specifically Monte Carlo simulations, on *tabular* data to generate our explanations.

### 4.2.1 Local Explanations for Tree Ensembles

Existing work on generating local explanations for tree ensembles involves finding counterfactual examples [132], identifying influential training samples [116], or identifying important features [87]. Importantly, none of these publications are specifically about (i) explaining errors, or (ii) explaining regressions. On the contrary, these publications are all based on binary classification tasks and the explanations do not necessarily provide insight into prediction mistakes.

Tolomei et al. [132] propose a method for generating counterfactual examples by identifying decision paths of interest that would result in a different prediction, then traversing down each of these paths and perturbing the instance  $x$  such that it satisfies the path in question. If this perturbation,  $x'$ , (i) satisfies the decision path, and (ii) changes the prediction in the overall ensemble, then it is a candidate transformation of  $x$ . After computing all possible candidate transformations by traversing over all paths of interest (i.e., those leading to a different prediction), the candidate transformation with the smallest distance from  $x$  is selected as the counterfactual example. The explanation, then, is the difference between  $x$  and  $x'$ . Although Tolomei et al. [132]’s method also produces contrastive explanations, our method differs from theirs since we are not aiming to identify one counterfactual example, but rather a range of feature values for which the prediction would be different. Another difference is that we do not assume full access to the original model.

Sharchilev et al. [116] also generate local explanations for tree ensembles. Their methodology is based on finding influential training samples in order to automatically improve the model, which differs from our work since their explanations are not of a contrastive nature. These influential training samples help us understand why a certain class was predicted for a given instance, but they make no reference to the alternative class(es). It should be noted that they include a use case on identifying harmful training examples — ones that contributed to incorrect predictions — which can be seen as a way to explain errors.

### 4.2.2 Feature Importance Explanations

Lundberg et al. [87] propose a method for determining how much each feature contributes to a prediction and present a ranked list of the most important features as the explanation. The approach is based on the computationally intensive Shapley values [86], for which the authors develop a tree-specific approximation. This differs from our method since identifying the most important

features is only a preliminary step in our pipeline — our work extends beyond this by including (i) feature bounds that result in reasonable predictions, and (ii) the relationship between the features and the target as a tool to help users inspect what goes wrong when the prediction error is large.

Ribeiro et al. [107] also propose a method for identifying local feature importances and this is the one we use in our pipeline. Their method, LIME, is model-agnostic and is based on approximating the original model locally with a linear model. We share their objective of evaluating users’ attitudes towards a model through local explanations but we further specify our task as explaining instances where there are large errors in predictions. Based on preliminary experiments, we find that LIME is insufficient for our task setting for two reasons:

- (i) For regression tasks, LIME’s approximation of the original model is not exact. This “added” error can be quite large given that our target is typically of order  $10^6$ , and this convolutes our definition of a large error.
- (ii) The features LIME deems most important are similar regardless of whether the prediction results in a large error or not, which does not provide any specific insight into why a large error occurs. These experiments are detailed in Section 4.4.

### 4.2.3 Contrastive Explanations

Other work on contrastive explanations includes identifying features that should be present or absent in order to justify a classification [27, 50] or model-agnostic counterfactuals [109, 141]. These all differ from our method since they are not specifically about explaining errors. Furthermore, the work by Dhurandhar et al. [27] and Hendricks et al. [50] is based on the binary presence or absence of input features, whereas our method perturbs inputs instead of removing them altogether.

### 4.2.4 Outlier Detection

Our work in this chapter can also be viewed as a form of outlier detection. However, it differs from the standard literature outlined by Pimentel et al. [102] with respect to the objective: we are not necessarily trying to identify outliers in terms of the training data but rather explain instances in the test set whose errors are so large that they are considered to be anomalies.

## 4.3 Method: MC-BRP

---

The intuition behind MC-BRP is based on identifying the unusual properties of a particular observation. We make the assumption that large errors occur due to unusual feature values in the test set that were not common in the training set.

Given an observation that results in a large error, MC-BRP generates a set of bounds for each feature that would result in a reasonable prediction as opposed to a large error. We also include the trend as part of the explanation in order to help users understand the relationship between each feature and the target, and how the input should be changed in order to change the output.

As pointed out previously, we consider our task of identifying and explaining large errors somewhat similar to that of an outlier detection problem. A standard definition of a statistical outlier is an instance that falls outside of a threshold based on the interquartile range. A widely used version of this, called Tukey’s fences, is defined as follows [133]:

$$[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)],$$

where  $Q_1$  and  $Q_3$  are the first and third quartiles, respectively.

**Definition 4.3.1.** Let  $x$  be an observation in the test set  $X$  and let  $t, \hat{t}$  be the actual and predicted target values of  $x$ , respectively. Let  $\epsilon$  be the corresponding prediction error for  $x$ , and let  $E$  be the set of all errors of  $X$ . Then  $\epsilon$  is a *large error* iff

$$\epsilon > Q_3(E) + 1.5(Q_3(E) - Q_1(E)),$$

where  $Q_1(E), Q_3(E)$  are the first and third quartiles of the set of errors, respectively. We denote this threshold as  $\epsilon_{large}$ .

We can view  $X$  in Definition 4.3.1 as a disjoint union of two sets:

- (i)  $R$ : the set of observations resulting in reasonable predictions, and
- (ii)  $L$ : the set of observations resulting in large errors.

We determine the  $n$  most important features based on LIME  $\Phi^{(x)} = \{\phi_j^{(x)}\}_{j=1}^n$ , for all  $x \in X$ . It should be noted there exist alternative methods for determining the most important features for a particular prediction [86], which would also be appropriate.

Given  $x \in X$ , for each  $\phi_j^{(x)} \in \Phi^{(x)}$ , we determine two sets of characteristics through Monte Carlo simulations:

- (i)  $[a_{\phi_j^{(x)}}, b_{\phi_j^{(x)}}]$ : the bounds for values of  $\phi_j^{(x)}$  such that  $x \in R, x \notin L$ .
- (ii)  $\rho_{\phi_j^{(x)}}$ : the relationship between  $\phi_j^{(x)}$  and the target,  $t$ .

We perturb the feature values for  $l \in L$  using Monte Carlo simulations in order to determine what feature values are required to produce a reasonable prediction. The algorithm for determining  $R'$ , the set of Monte Carlo simulations resulting in reasonable predictions, is detailed in Algorithm 4.1.

Given  $l \in L$ , we determine Tukey’s fences for each feature in  $\Phi^{(l)}$  based on the feature values from  $R$ . This gives us the bounds from which we sample for our feature perturbations.



Next, we randomly sample from these bounds for each  $\phi_j^{(l)} \in \Phi^{(l)}$   $m$ -times to generate  $mn$  versions of our original observation,  $l$ . We call the  $i$ -th perturbed version  $l'_i$ , where  $i \in \{1, \dots, mn\}$ .

We then test the original model  $f$  on each  $l'_i$ , obtain a new prediction,  $\hat{t}'_i$ , and construct  $R'$ , the set of perturbations resulting in reasonable predictions.

Once  $R'$  is generated, we compute the mean, standard deviation and Pearson coefficient [127] of the top  $n$  features of  $l \in L$ ,  $\Phi^{(l)}$ , based on this set.

---

**Algorithm 4.1** Monte Carlo simulation: creates a set of perturbed instances resulting in reasonable predictions  $R'$  for each large error  $l \in L$

---

**Input:** instance  $l$ , set of  $l$ 's most important features  $\Phi^{(l)}$ , ‘black-box’ model  $f$ , large error threshold  $\epsilon_{large}$ , number of MC perturbations per feature  $m$ .  
 $R' = \emptyset$   
**for all**  $\phi_j^{(l)}$  in  $\Phi^{(l)}$  **do**  
     $TF(\phi_j^{(l)}) \leftarrow \text{TUKEYS\_FENCES}(\phi_j^{(l)})$       *# Based on R*  
    **for**  $i$  in range  $(0, m)$  **do**  
         $\phi'_j{}^{(l)} \leftarrow \text{RANDOM\_SAMPLE}(TF(\phi_j^{(l)}))$   
         $l'_i \leftarrow l_i.\text{REPLACE}(\phi_j^{(l)}, \phi'_j{}^{(l)})$   
         $\hat{t}'_i \leftarrow f(l'_i)$       *#New prediction*  
        **if**  $|\hat{t}'_i - t_i| < \epsilon_{large}$  **then**  
             $R' \leftarrow R' \cup l'_i$   
**return**  $R'$

---

**Definition 4.3.2.** The *trend*,  $\rho_{\phi_j^{(x)}}$ , of each feature is the Pearson coefficient between each feature  $\phi_j^{(x)}$  and the predictions  $\hat{t}'_i$  based on the observations in  $R'$ . It is a measure of linear correlation between two variables [127].

The set of bounds for each feature in  $\Phi^{(x)}$  such that  $\hat{t}$  results in a reasonable prediction are based on the mean and standard deviation of each  $\phi_j^{(x)} \in \Phi^{(x)}$ .

**Definition 4.3.3.** The *reasonable bounds* for values of each feature  $\phi_j$  in  $\Phi^{(x)}$ ,  $[a_{\phi_j^{(x)}}, b_{\phi_j^{(x)}}]$ , are

$$\left[ \mu(\phi_j^{(x)}) - \sigma(\phi_j^{(x)}), \mu(\phi_j^{(x)}) + \sigma(\phi_j^{(x)}) \right],$$

where  $\mu(\phi_j^{(x)})$  and  $\sigma(\phi_j^{(x)})$  are the mean and standard deviation of each feature, respectively, based on  $R'$ .

We compute the trend and the reasonable bounds for each of the  $n$  most important features and present them to the user in a table. Table 4.1 shows an example of an explanation generated by MC-BRP; the dataset used for this example is detailed in Section 4.4.1.

Table 4.1: An example of an explanation generated by MC-BRP. Here, each of the input values is outside of the range required for a reasonable prediction, which explains why this particular prediction results in a large error.

Input	Definition	Trend	Value	Reasonable range
A	total_contract_hrs	As input ↑, sales ↑	9628.0	[4140,6565]
B	advertising_costs	As input ↑, sales ↑	18160.7	[8290,15322]
C	num_transactions	As input ↑, sales ↑	97332.0	[51219,75600]
D	total_headcount	As input ↑, sales ↑	226.0	[95,153]
E	floor_surface	As input ↑, sales ↑	2013.6	[972,1725]

## 4.4 Experimental Setup

Current explanation methods mostly serve individuals with ML expertise [12, 47], but they should be extended to cater to users outside of the ML community [90]. Unlike previous work, our method, MC-BRP, generates contrastive explanations by framing the explanation around the prediction error, and aims to help users understand (i) what contributed to the large error, and (ii) what would need to change in order to produce a reasonable prediction. Presenting explanations in a contrastive manner helps frame the problem and narrows the user’s focus regarding the possible outcomes [51, 75].

Our explanations are contrastive because they display to the user what would have needed to change in the input order to obtain an alternative outcome from the model — in other words, why this prediction results in a large error as opposed to a reasonable prediction.

### 4.4.1 Dataset and Model

Our task is predicting monthly store sales using an internal company dataset with 45 features including financial, workforce and physical store aspects. Since not all of our practitioners have experience with ML, using an internal dataset with familiar features allows them to leverage some of their domain expertise. The dataset includes 45628 observations from 563 stores, collected at four-week intervals spanning from 2010–2015. We split the data by year (training: 2010–2013, test: 2014–2015) to simulate a production environment, and we treat every unique combination of store, interval and year as an independent observation. After preprocessing, we have 21415 and 12239 observations in our training and test sets, respectively. We train the gradient boosting regressor from scikit-learn with the default settings and obtain an  $R^2$  of 0.96.

We verify our assumption that large errors are a result of unusual feature values by generating MC-BRP explanations for all instances in our test set using  $n = 5$  features and  $m = 10000$  Monte Carlo simulations. In our dataset, we find that 48% of instances resulting in large errors have feature values outside the reasonable range for all of the  $n = 5$  most important features, compared

Table 4.2: The top  $n = 5$  features according to LIME for observations resulting in large errors vs. reasonable predictions.

Large errors		Reasonable Predictions	
advertising_costs	0.188	advertising_costs	0.187
total_contract_hrs	0.175	total_contract_hrs	0.179
num_transactions	0.151	num_transactions	0.156
floor_surface	0.124	total_headcount	0.134
total_headcount	0.123	floor_surface	0.122
month	0.109	month	0.094
mean_tenure	0.046	mean_tenure	0.046
earnings_index	0.033	earnings_index	0.031

to only 24% of instances resulting in reasonable predictions. Although this is not perfect, it is clear that MC-BRP produces explanations that are at least somewhat able to distinguish between these two types of predictions.

#### 4.4.2 Comparison to LIME

Hilton [52] states that explanations are selective – it is not necessary or even useful to state all the possible causes that contributed to an outcome. The significant part of an explanation is what distinguishes it from the alternative outcome. If LIME explanations were suitable for our problem, then we would expect to see different features deemed important for instances resulting in large errors compared to those resulting in acceptable errors. This would help the user understand why a particular prediction resulted in a large error.

However, when generating LIME explanations for our test set using  $n = 5$  features, we do not see much of a distinction in the most important features between predictions that result in large errors and those that do not. For example, advertising\_costs is one of the top 5 most important features in 18.8% of instances with large errors and 18.7% of instances with reasonable predictions. These results are summarized in Table 4.2.

Furthermore, we originally tried to design our control group user study using explanations from LIME, but found that test users from the retailer could not make sense of the objective questions about prediction errors because LIME does not provide any insight about errors specifically. Given that we could not even ask questions about errors using LIME explanations to users without confusing them, it is clear that LIME is inappropriate for our task.

#### 4.4.3 User Study Design

We test our method on a real dataset with real users, both from the retailer. We include a short tutorial about predictive modeling along with some questions to check users’ understanding as a preliminary component of the study. This is because our users are a diverse set of individuals with a wide range of capabilities,



Figure 4.1: The visual description of the model shown to the users: a graph comparing the predicted sales and actual sales based on the original model. The red line depicts perfect predictions.

including data scientists, human resource strategists, and senior members of the executive team. We also include participants from the University of Amsterdam to simulate users who could one day work in this environment. In total, we have 75 participants: 44 in the treatment group and 31 in the control group.

All users are first provided with a visual description of the model: a simple scatter plot comparing the predicted and actual sales (as shown in Figure 4.1). We also show a pie chart depicting the proportion of predictions that result in large errors to give users a sense of how frequently these mistakes occur. In our case, this is 4%. Since our users are diverse, we want to make our description of the model as accessible as possible while allowing them to form their own opinions about how well the model performs. Participants in the treatment group are shown MC-BRP explanations, while those in the control group are not given any explanation.

The study contains two components, objective and subjective, corresponding to **RQ3.1** and **RQ3.2**, respectively. The objective component is meant to quantitatively evaluate whether or not users understand explanations generated by MC-BRP, while the subjective component assesses the effect of seeing the explanation on users’ attitudes towards the model. We base the objective component on *human-grounded metrics*, a framework proposed by Doshi-Velez

and Kim [29], where the tasks conducted by users are simplified versions of the original task. Instead of asking users to correctly predict retail sales values of order  $10^6$ , we ask them to determine whether or not a prediction will result in a large error.

Table 4.3: Summary of tasks performed in user study for the treatment and control groups. The subjective questions are asked twice.

Treatment	Control
1. Short modeling tutorial	1. Short modeling tutorial
2. Visual model description	2. Visual model description
3. Subjective questions	3. Subjective questions
4. Objective questions	4. Dummy questions
5. Subjective questions	5. Subjective questions

Table 4.4: Summary of simulations performed in objective portion of the user study.

Type	Provide user with	User’s task
Forward	(1) Input values (2) Explanation	Simulate output
Counterfactual	(1) Input values (2) Explanation (3) Output	Manipulate input to change output

To answer **RQ3.1**, we ask users in the treatment group to perform two types of simulations, both suggested by Doshi-Velez and Kim [29] and summarized in Table 4.4. The first is *forward simulation*, where we provide participants with the (i) input values, and (ii) explanation. We then ask them to simulate the output — whether or not this prediction will result in a large error. The second is *counterfactual simulation*, where we provide participants with the (i) input values, (ii) explanation, and (iii) output. We then ask them what they would have needed to change in the input in order to change the output. In other words, we want participants to determine how the input features can be changed (according to the trend) in order to produce a reasonable prediction as opposed to one that results in large error. These objective questions are designed to test whether or not a participant understands the explanations enough to predict or manipulate the model’s output. We ask every participant in the treatment group to perform two forward simulations and one counterfactual simulation, and we show the same examples to all users.

For the control group, we found that we could not ask the objective questions in the same way we did for the treatment group. This is because the objective component involves simulating the model based on the explanations (see Table 4.4), which is not possible if the explanations are not provided. In fact, we initially left the objective questions in the control group study, but preliminary

testing on some users from the retailer showed that this was confusing and unclear, similar to when we tried using LIME explanations. We were concerned this confusion would skew users’ perceptions of the model and therefore convolute the results of **RQ3.2**. Instead, we show participants in the control group the (i) input values, and (ii) output – whether or not the example resulted in a large error. In this case, we ask them *if they have enough information* to determine why the example does (or does not) result in a large error. This serves as a dummy question to engage users with the task without confusing them. We cannot ask users in the control group to simulate the model since they do not see the explanations, but we want to mimic the conditions of the treatment group as closely as possible. Therefore, **RQ3.1**, is solely evaluated on users from the treatment group.

To answer **RQ3.2**, we contrast results from the treatment and control groups. We ask both groups of users the same four subjective questions twice, once towards the beginning of the study and once again at the end. We ask the questions at the beginning of the study to evaluate the distribution of preliminary attitudes towards the model, based solely on the visual description. We ask the questions at the end of the study to evaluate the effectiveness of MC-BRP explanations, by comparing the results from the treatment and control groups. The questions we devised are based on the user study by ter Hoeve et al. [130]. Table 4.3 summarizes the experimental setup for the treatment and control groups. Again, the treatment and control groups are treated exactly the same with the exception of the objective questions – we only ask these to the treatment group since we cannot ask users to simulate the model without giving them the explanation.

## 4.5 Experimental Results

---

In this section, we evaluate the explanations generated by MC-BRP in terms of (i) objective questions, and (ii) subjective questions.

### 4.5.1 Objective Questions

The results for users’ objective comprehension of MC-BRP explanations are summarized in Table 4.5. We see that explanations generated by MC-BRP are both: (i) interpretable and (ii) actionable, with an average accuracy of

Table 4.5: Results from the objective questions in the user study.

Human accuracy	
Forward simulation 1	84.1%
Forward simulation 2	84.1%
Counterfactual simulation	75.0%
Average	81.1%

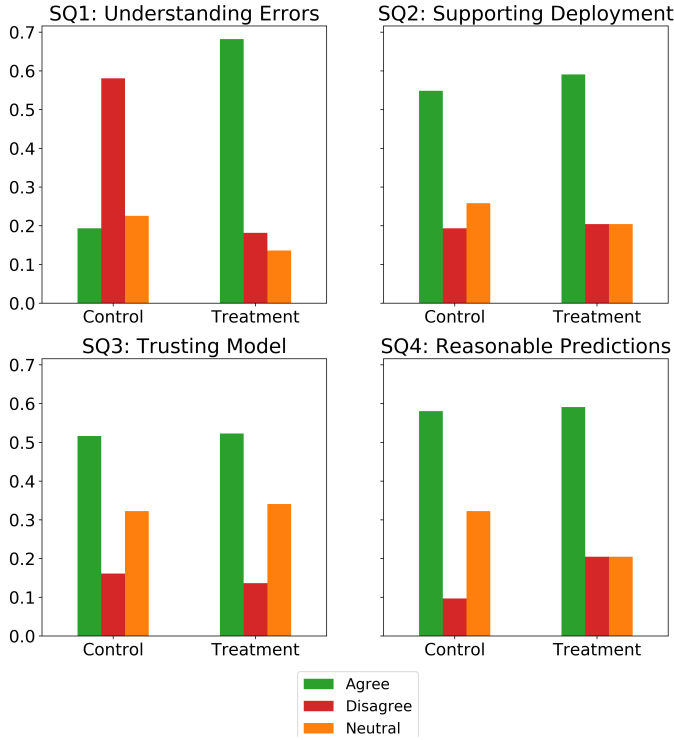


Figure 4.2: Results from a within-subject study comparing answers between the treatment (MC-BRP explanation) and control (no explanation) groups.

81.1%. This answers **RQ3.1**. When asked to perform forward simulations, the proportion of correct answers was 84.1% for both questions. This indicates that the majority of users were able to interpret the explanations in order to simulate the model’s output (**RQ3.1**: interpretable). When asked to perform counterfactual simulations, the proportion of correct answers was slightly lower at 75.0%, but still indicates that the majority of users were able to determine how to manipulate the model’s input in order to change the output (**RQ3.1**: actionable).

#### 4.5.2 Subjective Questions

In order to understand the impact of MC-BRP explanations on users’ attitudes towards the model, we ask them the following subjective questions:

- **SQ1:** I understand why the model makes large errors in predictions.
- **SQ2:** I would support using this model as a forecasting tool.

- **SQ3:** I trust this model.
- **SQ4:** In my opinion, this model produces mostly reasonable outputs.

To ensure our populations did not have different initial attitudes towards the model, we compared their answers on the subjective questions after only showing a visual description of the model. The visual description is a graph comparing the predicted sales to the actual sales, which allows users to see the distribution of errors made by the model (see Figure 4.1). We found no statistically significant difference ( $\chi^2$  test,  $\alpha = 0.05$ ) in initial attitudes towards the model, which allows us to postulate that any difference discovered between the two groups is a result of the treatment they were given (i.e., MC-BRP explanation vs. no explanation).

Figure 4.2 shows the distributions of answers to the four subjective questions in the treatment and control groups. The difference in distributions is significant for SQ1 ( $\chi^2 = 18.2$ ,  $\alpha = 0.0001$ ): users in the treatment group agree with the statement more than users in the control group. However, we find no statistically significant difference between the two groups for the remaining questions ( $\chi^2$  test,  $\alpha = 0.05$ ). That is, MC-BRP explanations help users understand why the model makes large errors in predictions, but do not have an impact on users' trust or confidence in the model, or on their willingness to support its deployment.

## 4.6 Discussion

---

Since our original motivation was to provide an explanation system that can be used by forecasting analysts, we conducted a more in-depth analysis of the results to determine if there was a difference in attitudes between users depending on their background (e.g., practitioners from the retailer or researchers from the university).

### 4.6.1 Comparing Attitudes Conditioned on Background

Table 4.6 shows the distribution of practitioners and researchers in the treatment and control groups. Since we have a slight imbalance in background between the treatment and control groups, we test whether or not our results still hold when conditioning on background and confirm that they do.

Again, we do not find statistically significant differences in initial attitudes towards the model ( $\chi^2$  test,  $\alpha = 0.05$ ). For researchers, the distribution of

Table 4.6: Distribution of practitioners and researchers in the treatment and control groups.

Background	Practitioners	Researchers
Treatment	52%	48%
Control	58%	42%



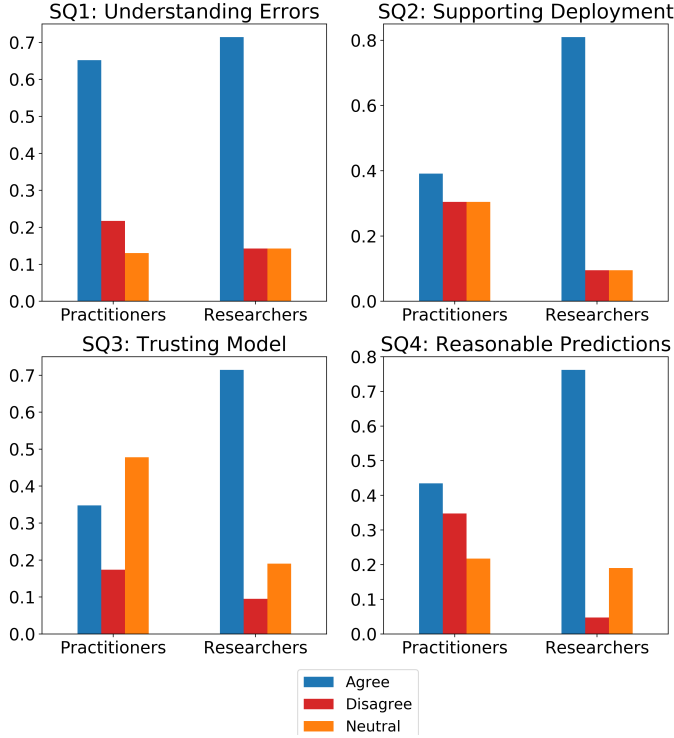


Figure 4.3: Results from a within-subject study comparing answers between participants who are practitioners or researchers (in the treatment group).

answers between treatment and control groups is significantly different for SQ1 ( $\chi^2 = 14.2$ ,  $\alpha = 0.001$ ), but does not differ for SQ2, SQ3, or SQ4 ( $\chi^2$  test,  $\alpha = 0.05$ ). The same holds for practitioners: the distributions are significantly different only for SQ1 ( $\chi^2 = 6.94$ ,  $\alpha = 0.05$ ). This is consistent with our results in Section 4.5. In both cases, users in the treatment group agree with SQ1 more than users in the control group, indicating that MC-BRP explanations help users understand why the model makes large errors in predictions, regardless of whether they are practitioners or researchers. Although the results are statistically significant for both groups, it should be noted that the results hold more strongly for researchers compared to those for practitioners, given the  $\chi^2$  values.

#### 4.6.2 Comparing Attitudes in the Treatment Group

Based on the users who saw the explanations, we compare the distributions of answers between practitioners and researchers in Figure 4.3 in order to understand the needs of different types of users. We find that there is a significant difference

between practitioners and researchers for SQ2 ( $\chi^2 = 7.94$ ,  $\alpha = 0.05$ ), indicating that more researchers are in favor of using the model as a forecasting tool, and less are against it or have a neutral attitude, in comparison to the practitioners. We also find a significant difference for SQ3 ( $\chi^2 = 5.98$ ,  $\alpha = 0.05$ ): a larger proportion of researchers trust the model, while the majority of practitioners have neutral feelings. The results for SQ4 are significant as well ( $\chi^2 = 6.86$ ,  $\alpha = 0.05$ ): although the majority of users in both groups believe the model produces reasonable predictions, a larger proportion of the practitioners disagree with this statement in comparison to the researchers.

We see no significant difference between groups for SQ1 ( $\chi^2$  test,  $\alpha = 0.05$ ), which makes sense given that we showed that MC-BRP explanations have a similar effect on both practitioners and researchers when comparing users in the treatment and control groups in Section 4.6.1.

Overall, these results suggest that our user study population is fairly heterogeneous, and that users from different backgrounds have different criteria for deploying or trusting a model, and varying levels of confidence regarding the accuracy of its outcomes.

### 4.6.3 User Study Limitations

Like any user study, ours has some limitations. It would have been preferable to distribute users more evenly in terms of the proportion of users in the treatment and control groups, as well as the proportion of practitioners and researchers in each of these groups. Unfortunately, this was not possible in our case because we recruited participants in two rounds: first for the treatment group, and then afterwards for the control group. One option could be to discard some practitioners in the control group in order to have a better balance in terms of background, but we felt it was more important to have as many users as possible, and it would not be clear how to choose which users to discard. Fortunately, we found that our results still hold when conditioning on background as mentioned in Section 4.6.1. In future work, we plan to recruit for both groups at the same time to avoid issues like these.

We also acknowledge that not having a baseline method to compare to is a limitation of our study. In our case, the main issue is that there simply does not exist a method that is specifically for explaining errors in regression predictions, which would make asking questions about errors (i) unfair, and (ii) confusing, as mentioned in Sections 4.4.2 and 4.4.3. However, now that MC-BRP exists, it can serve as a baseline for future work on erroneous predictions, which is another contribution of this paper.

## 4.7 Conclusion

---

In this chapter, we have proposed a method, MC-BRP, that provides users with contrastive explanations about predictions resulting in large errors based on: (i) the set of bounds for which reasonable predictions would be expected for each

of the most important features. (ii) the trend between each of these features and the target.

Given a large error, MC-BRP generates a set of perturbed versions of the original instance that result in reasonable predictions. This is done by performing Monte Carlo simulations on each of the features deemed most important for the original prediction. For each of these features, we determine the bounds needed for a reasonable prediction based on the mean and standard deviation of this new set of reasonable predictions. We also determine the relationship between each feature and the target through the Pearson correlation, and present these to the user as the explanation.

We evaluate MC-BRP both objectively (**RQ3.1**) and subjectively (**RQ3.2**) by conducting a user study with 75 real users, including both researchers and practitioners. We answer **RQ3.1** by conducting two types of simulations to quantify how (i) interpretable, and (ii) actionable our explanations are. Through forward simulations, we show that users are able to interpret MC-BRP explanations by simulating the model's output with an average accuracy of 84.5%. Through counterfactual simulations, we show that MC-BRP explanations are actionable with an accuracy of 76.2%.

We answer **RQ3.2** by conducting a between-subject experiment with subjective questions. The treatment group sees MC-BRP explanations, while the control group does not see any explanation. We find that explanations generated by MC-BRP help users understand why models make large errors in predictions (SQ1), but do not have a significant impact on support in deploying the model (SQ2), trust in the model (SQ3), or perceptions of the model's performance (SQ4). These results still hold when conditioning on users' background (practitioners vs. researchers). We also conduct an analysis on the treatment group to compare results between practitioners and researchers. We find significant differences for SQ2, SQ3 and SQ4, but do not find a significant difference in attitudes for SQ1.

The answer to **RQ3** is yes: we can create an explanation method based on a real-world use case by first identifying a use case where explanations are required and identifying what users want in the context of this use case. We can evaluate in a user-centric manner by conducting a user study based on the use case, which includes both objective and subjective components.

So far, this thesis has focused on creating knowledge about responsible AI practices, specifically on developing new methods for behavior-based explanations (see Chapter 1), where we explain predictions from ML models. In the next and final part of the thesis, we will focus more on process-based explanations, where we will investigate how to *translate* knowledge about responsible AI practices to the next generation of AI researchers.

## Reproducibility

---

To facilitate the reproducibility of the work in this chapter, our code is available at <https://github.com/a-lucic/mcbrp>.



# Part III

## Pedagogy



---

# Teaching Responsible AI through Reproducibility

---

In this final part of the thesis, we investigate how to communicate responsible artificial intelligence (AI) practices to the next generation of AI researchers. In this chapter, we address the following research question:

**RQ4:** How can we teach about responsible AI topics to a technical, research-oriented audience?

We answer **RQ4** by designing a course that is centered on a reproducibility project, where students work in teams to reimplement existing responsible AI algorithms from top AI conferences and reproduce the experiments reported in the papers. We report on our findings as we taught the course over two academic years and share recommendations for implementing similar courses in the future.

## 5.1 Introduction

---

For several decades, the University of Amsterdam has offered a research-oriented Master of Science (MSc) program in AI. The main focus of the program is on the technical machine learning (ML) aspects of the major sub-fields of AI, such as computer vision, information retrieval, natural language processing, and reinforcement learning. One of the most recent additions to the MSc AI curriculum is a mandatory course called *Fairness, Accountability, Confidentiality and Transparency in Artificial Intelligence (FACT-AI)*. This course was first taught during the 2019–2020 academic year and focuses on teaching FACT-AI topics through the lens of reproducibility. The main project involves students working in groups to re-implement existing FACT-AI algorithms from papers in top AI venues. There are approximately 150 students enrolled in the course each year.

The motivation for the course came from the MSc AI students themselves, who often play an important role in shaping the curriculum in order to meet the evolving requirements of researchers in both academia and industry. As the influence of AI on decision making is becoming increasingly prevalent in day-to-day life, there is a growing consensus that stakeholders who take part in the design

---

This chapter was published at the AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-AAAI 2022) under the title “Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence” [83].

or implementation of AI algorithms should reflect on the ethical ramifications of their work, including developers and researchers [19]. This is especially true in situations where data-driven AI systems affect some demographic sub-groups differently than others [5, 100]. As a result, our students have shown an increased interest in the ethical issues surrounding AI systems and requested that the university put together a new course focusing on responsible AI.

Since our MSc AI program is characterized by a strong emphasis on understanding, developing, and building AI algorithms, we believe that a new course on responsible AI in this program should also have a hands-on approach. The course is designed to address technical aspects of key areas in responsible AI: (i) fairness, (ii) accountability, (iii) confidentiality, and (iv) transparency, which we operationalize through a reproducibility project. We believe a strong emphasis on reproducibility is important from both an educational point of view and from the point of view of the AI community, since the (lack of) reproducible results has become a major point of critique in AI [56]. Moreover, the starting point of almost any junior AI researcher (and most AI research projects in general) is re-implementing existing methods as baselines. The FACT-AI course is situated at a point in the program where students have learned the basics of ML and are ready to start experimenting with, and building on top of, state-of-the-art algorithms. Given that our MSc AI program is fairly research-oriented, it is important for students to experience the process of reproducing work done by others (and how difficult this is) at an early stage in their careers. We also believe reproducibility is a fundamental component of FACT-AI: the cornerstone of fair, accountable, confidential and transparent AI systems is having correct and reproducible results. Without reproducibility, it is unclear how to judge if a decision-making algorithm adheres to any of the FACT principles.

In the 2019–2020 academic year, we operationalized our learning ambitions regarding reproducibility by publishing a public repository with selected code implementations and corresponding reports from the group projects. In the 2020–2021 academic year, we took the projects one step further and encouraged students to submit to the ML Reproducibility Challenge,<sup>1</sup> a competition that solicits reproducibility reports for papers published in conferences such as NeurIPS, ICML, ICLR, ACL, EMNLP, CVPR and ECCV. Although the challenge broadly focuses on all papers submitted to these conferences, we focus exclusively on papers covering FACT-AI topics in our course. Submitting to the challenge gives students a chance to experience the whole AI research pipeline, from running experiments, to writing rebuttals, to receiving the official notifications. Of the 23 papers that were accepted to the ML Reproducibility Challenge in 2021, 9 came from groups in the FACT-AI course.

In this chapter, we describe the FACT-AI course at the University of Amsterdam: a one month, full-time course based on examining ethical issues in AI using reproducibility as a pedagogical tool. Students work in groups to re-implement (and possibly extend) existing algorithms from top AI venues on FACT-AI topics. The course also includes lectures that cover the high-level

---

<sup>1</sup><https://paperswithcode.com/rc2020>



principles of FACT-AI topics, as well as paper discussion sessions where students read and digest prominent FACT-AI papers. In this chapter, we outline the setup for the FACT-AI course and the experiences we had while running the course during the 2019–2020 and 2020–2021 academic years at the University of Amsterdam.

The remainder of this chapter is structured as follows. In Section 5.2, we discuss related work, specifically other courses about responsible AI. In Section 5.3, we detail ongoing reproducibility efforts in the AI community. In Section 5.4, we explain the learning objectives for our course, and explain how we realized those objectives in Section 5.5. We reflect on the feedback we received about the course in Section 5.6, as well as what worked (Section 5.7) and what did not (Section 5.8), before concluding in Section 5.9.

## 5.2 Related Work

---

There have been multiple calls for introducing ethics in computer science courses in general, and in AI programs in particular [23, 45, 71, 100, 110, 119, 120]. Several surveys have investigated how existing responsible computing courses are organized [35, 38, 101, 106].

### 5.2.1 Characterizing Responsible AI Courses

There are two primary approaches to integrating such components into the curriculum: (i) stand-alone courses that focus on ethical issues such as FACT-AI topics, and (ii) a holistic curriculum where ethical issues are introduced and tackled in each course. In general, the latter is rare [35, 101, 110], and can be difficult to organize due to a lack of qualified faculty or relevant expertise [9, 106]. We opt for the first approach since our course is a new addition to an existing study program.

Fiesler et al. [35] analyze 202 courses on “tech ethics”. Their survey examines (i) the departments the courses are taught from, as well as the home departments of the course instructors, (ii) the topics covered in the courses, organized into 15 categories, and (iii) the learning outcomes in the courses. In our case, both the FACT-AI course and its instructors are from the Informatics Institute of the Faculty of Science at the University of Amsterdam. Our learning objectives (see Section 5.4) correspond to the following learning objectives from Fiesler et al. [35]: “*Critique*”, “*Spot Issues*”, and “*Create solutions*”. According to their content topic categorization, our course includes “*AI & Algorithms*” and “*Research Ethics*”: the former since the course deals explicitly with AI algorithms under the FACT-AI umbrella, and the latter due to its focus on reproducibility.

We note that “*AI & Algorithms*” is only the fifth-most popular topic according to the survey, after “*Law & Policy*”, “*Privacy & Surveillance*”, “*Philosophy*”, and “*Inequality, Justice & Human Rights*” [see Table 2, 35]). Although we believe these topics are important, we also wanted to avoid the feeling that the course was a “distraction from the real material” [72], especially since (i) the

majority of our students are coming from a technical background into a technical MSc program, and (ii) the FACT-AI course is mandatory for all students in the MSc AI program.

### 5.2.2 Similar Responsible AI Courses

The two courses that are the most similar to ours are those of Lewis and Stoyanovich [72] and Yildiz et al. [145].

Lewis and Stoyanovich [72] describe a course for responsible data science. Similar to our course, they focus on the technical aspects of AI, involving lectures, readings, and a final project. However, their course differs from ours since the main project in their course is focused on examining the interpretability of an automated decision making system, while the main project in our course is focused on reproducibility.

Yildiz et al. [145] describe a course based on reproducing experiments from AI papers, focusing on “low-barrier” reproducibility. Similar to our course, this course involves replicating a paper from scratch or reproducing the experiments using existing code, performing hyperparameter sweeps, and testing with new data or with variant algorithms. Another similarity is that they released a public repository of re-implemented algorithms,<sup>2</sup> which we also did for the 2019–2020 iteration of our course (see Section 5.5.3). However, their course differs from ours since theirs focuses on AI papers in general, while our course focuses exclusively on FACT-AI papers.

There are several courses that focus more on the philosophical or social science perspectives of AI ethics. Green [43] describes an undergraduate AI ethics course that teaches computer science majors to analyse issues using different ethical approaches and how to incorporate these into an *explicit* ethical agent. Shen et al. [117] introduce a toolkit in the form of “Value Cards” to inform students and practitioners about the social impacts of ML models through deliberation. Green and Crotts [44] propose an approach to ethics education using “argument schemes” that summarize key ethical considerations for specialized domains such as healthcare or national defense. Furey and Martin [37] introduce ethics concepts, primarily utilitarianism, into an existing AI course about autonomous vehicles by studying several variations of the Trolley Problem. Burton et al. [17] teach ethics through science fiction stories complemented with philosophy papers, allowing students to reflect and debate difficult content without emotional or personal investment since the stories are not tied to “real” issues. Skirpan et al. [120] describe an undergraduate course on human-centred computing which integrates ethical thinking throughout the design of computational systems. Unlike these courses, our course focuses more on the technical aspects of ethical AI. However, incorporating such non-technical perspectives is something we would like to do in future iterations of our course, perhaps through one of the mechanisms employed by some of these courses.

---

<sup>2</sup><https://reproducedpapers.org/>

## 5.3 Reproducibility in ML Research

---

There have been several criticisms about the lack of reproducibility in AI research. Some have postulated that this is due to a combination of unpublished code and high sensitivity of training parameters [56], while others believe the rapid rate of progress in ML research results in a lack of empirical rigor [115]. Although typically well-intentioned, some papers may disguise speculation as explanation, obfuscate content behind math or language, and fail to attribute the correct sources of empirical gains [76].

Several efforts have been made to investigate and increase the reproducibility of AI research. In 2021, NeurIPS introduced a paper checklist including questions about reproducibility, along with a template for submitting source code as supplementary material [11]. The Association of Computing Machinery introduced a badging system that indicates how reproducible a paper is [2]. Papers with Code is an organization that provides links to official code repositories and datasets in arXiv papers [123]. It also hosts an annual ML Reproducibility Challenge: a community-wide effort to investigate the reproducibility of papers accepted at top AI conferences, which we incorporated into the 2020–2021 iteration of the FACT-AI course.

In an ideal scenario, reproducibility issues would be handled prior to publication [115], but it can be difficult to catch such shortcomings in the review process due to the increasing number of papers submitted to AI conferences. Therefore, we believe it is of utmost importance that the next generation of AI researchers – including our own students – can (i) identify and (ii) avoid these pitfalls while conducting their own work. This, in combination with the fact that reproducibility is a fundamental component of responsible AI research, is why we opted to teach the FACT-AI course through the lens of reproducibility.

Our course is centered around a group project where students re-implement a recent FACT-AI algorithm from a top AI conference. This project has three components: (i) a reproducibility report, (ii) an associated code base, and (iii) a group presentation. In Section 5.5.3, we provide more details on the project and the outputs it resulted in.

## 5.4 Learning Objectives

---

In the FACT-AI course, we aim to make students aware of two types of responsibility: (i) towards society in terms of potential implications of their research, and (ii) towards the research community in terms of producing reproducible research. In this section, we outline the learning objectives for the FACT-AI course and explain how it fits within the context of the MSc AI program at the University of Amsterdam.

Table 5.1 shows the setup of the first year of the 2-year MSc AI program. Each semester at the University of Amsterdam is divided into three periods: two 8-week periods followed by one 4-week period. During an 8-week period, students follow two courses in parallel. During the 4-week period, they only

Table 5.1: The first year of the MSc AI program at the University of Amsterdam, 2019–2020.

Course	Sem. 1	Sem. 2	EC
Computer Vision 1	■ □ □ □ □ □		6
Machine Learning 1	■ □ □ □ □ □		6
Natural Language Processing 1	□ ■ □ □ □ □		6
Deep Learning 1	□ ■ □ □ □ □		6
Fairness, Accountability, Confidentiality and Transparency in AI	□ □ ■ □ □ □		6
Information Retrieval 1	□ □ □ ■ □ □		6
Knowledge Representation and Reasoning	□ □ □ ■ □ □		6
Elective 1	□ □ □ □ ■ □		6
Elective 2	□ □ □ □ ■ □		6
Elective 3	□ □ □ □ □ ■		6

follow a single course. The FACT-AI course takes place during the 4-week period at the end of the first semester, after students have taken Computer Vision 1, Machine Learning 1, Natural Language Processing 1 and Deep Learning 1. It is the only course students follow during this period, so we believe it is beneficial to have them focus on one main project – reproducing an existing FACT-AI paper. The learning objectives for the course are as follows:

- **LO #1: Understanding FACT topics.** Students can explain the major notions of fairness, accountability, confidentiality, and transparency that have been proposed in the literature, along with their strengths and weaknesses.
- **LO #2: Understanding algorithmic harm.** Students can explain, motivate, and distinguish the main types of algorithmic harm, both in general and in terms of concrete examples where AI is being applied.
- **LO #3: Familiarity with FACT methods.** Students are familiar with recent peer-reviewed algorithmic approaches to fairness, accountability, confidentiality, and transparency in the literature.
- **LO #4: Reproducing FACT solutions.** Students can assess the degree to which recent algorithmic solutions are effective, especially with respect to the claims made in the original papers, while understanding their limitations and shortcomings.

## 5.5 Course Setup

---

The FACT-AI course is organized around (i) lectures, (ii) paper discussions, and (iii) a group project. It has had two iterations so far: the 2019–2020 iteration was taught in person, while the 2020–2021 iteration was taught online due to

the COVID-19 pandemic. In this section, we detail how we realized the learning objectives from Section 5.4 and describe the challenges in adapting the course to an online format.

### 5.5.1 Lectures

To further the understanding of FACT-AI topics (LO1), we provide one general lecture for each of the 4 topics, along with a lecture specifically about reproducibility. Lectures are an opportunity for students to familiarize themselves with algorithmic harm (LO2). Students are encouraged to ask questions that lead to discussions about potential harm done by applications of AI. This was more challenging in the second iteration of the course due to the online format, but we hope that facilitating such discussions will be more straightforward once we return to in-person classes.

In addition to the general lectures, we also include some guest lectures. These are used to either discuss specific types of algorithmic harm (LO2), examine specific FACT-AI algorithms in depth (LO3), or expand on the non-technical aspects of FACT-AI. Some examples of guest lectures include a lecture on AI accountability from a legal perspective by an instructor from the law department of the University of Amsterdam, and a lecture by two former FACT-AI students who explained how they turned their group project into an ICML 2021 workshop paper [95].<sup>3</sup>

### 5.5.2 Paper Discussions

The goal of the paper discussion sessions is for students to learn about prominent FACT-AI methods (LO3), and learn to think critically about the claims made in the papers we discuss (LO4). Students first read a seminal FACT-AI paper on their own while trying to answer the following questions:

- What are the main claims of the paper?
- What are the research questions?
- Does the experimental setup make sense, given the research questions?
- What are the answers to the research questions? Are these supported by experimental evidence?

Once students have read the papers, they participate in smaller discussion sessions with their peers about their answers to the questions above. After each discussion session, all the groups are brought back together for a “dissection” session, where an instructor goes over the same seminal paper, giving an overview of the papers’ strengths and weaknesses. Each session was presented by a different instructor to show that there is no single way of examining a research paper, and that different researchers will bring different perspectives to their assessment of

---

<sup>3</sup>This was later extended to a full paper at the International Conference on Hybrid Human-AI (HHAI 2022) [96].

papers. The following papers were covered during the discussion sessions: Hardt et al. [49] on fairness; Ribeiro et al. [107] on transparency; and Abadi et al. [1] on confidentiality.

### 5.5.3 Group Project

#### Reproduction of a FACT-AI paper

The purpose of the group project is to have students investigate the claims made by the authors of recent FACT-AI papers by diving into the details of the methods and their implementations. Using what they have learned from the paper discussion sessions, students work in groups to re-implement an existing FACT-AI algorithm from a top AI conference and re-run the experiments in the paper to determine the degree to which they are reproducible (LO4). If the code is already available, then they must extend the method in some way. The project consisted of three deliverables: (i) a reproducibility report, (ii) an associated code base, and (iii) a group presentation. In order to ensure the project is feasible, we select 10–15 papers in advance for groups to choose from. Our criteria for including papers is as follows:

- The paper is on a FACT-AI topic.
- At least one dataset in the paper is publicly available.
- Experiments can be run on a single GPU (which we provide access to).
- It is reasonable for a group of 3–4 MSc AI students to re-implement the paper within the timeframe of the course. In our case, students work on this project for one-month full-time.

To ease the load for our teaching assistants (TAs), we have several groups working on the same paper. We assign papers to TAs based on their interests by asking them to rank the set of candidate papers in advance. We also encourage them to suggest alternative papers provided they fit the criteria. The TAs read the papers before the course starts in order to ensure they have a sufficient, in-depth understanding of the work such that they can guide students through the project. This also serves as an extra feasibility check, to ensure that the papers are indeed a good fit for our course.

Each group writes a report about their efforts following the structure of a standard research paper (i.e., introduction, methodology, experiments, results, conclusion). They also include aspects specific to reproducibility such as explaining the difficulties of implementing certain components, as well as describing any communication they had with the original authors. In addition to the source code, students provide all results in a Jupyter notebook along with a file to install the required environment.

### First Iteration: Contributing to an Open Source Repository

In the 2019–2020 iteration of the course, we created a public repository on GitHub, which contains a selection of the implementations done by our students: <https://github.com/uva-fact-ai-course/uva-fact-ai-course>. The TAs who assisted with the course decided which implementations to include and cleaned up the code so it all fit into one cohesive repository. This had multiple motivations. First, it taught students how to improve the reproducibility of their own work by releasing the code, while also giving them a sense of contributing to the open-source community. Second, a public repository can serve as a starting point for personal development in their future careers; companies often ask to see existing code or projects that prospective employees have worked on. Some students added the project to their CVs, while others wrote blog posts about their experiences,<sup>4</sup> linking back to the repository.

### Second Iteration: The ML Reproducibility Challenge

In the 2020–2021 iteration of the course, we formally participated in the annual ML Reproducibility Challenge [123] in order to expose our students to the peer-review process. This gave students something to strive towards and offered perspectives beyond simply getting a grade for the project. Most importantly, it gave them the opportunity to experience the full research pipeline: (i) reading a technical paper, (ii) understanding a paper’s strength and weaknesses, (iii) implementing (and perhaps also extending) the paper, (iv) writing up the findings, (v) submitting to a venue with a deadline, (vi) obtaining feedback, (vii) writing a rebuttal, and (viii) receiving the official notification. To encourage students to formally submit to the ML Reproducibility Challenge, we offered a 5% boost to their final grades if they submitted. Of the 32 groups in the FACT-AI course, 30 (94%) groups submitted their reproducibility reports to the ML Reproducibility Challenge, of which 9 groups had their papers accepted.

#### 5.5.4 Online Course Format

The second iteration of the course was taught in January 2021, when the COVID-19 pandemic forced us to move classes and interactions online. Students made use of various channels to communicate: WhatsApp, Discord, and Slack. Canvas was the primary mode of communication between the instructors and the students, allowing students to ask questions and instructors to communicate various announcements.

Lectures were live, with frequent Q&A breaks to stimulate interactivity. Paper discussion sessions were organized in different online meeting subrooms where students discussed the papers together. This proved to be a challenge: while some subrooms had productive discussions, others struggled to get the conversation going.

---

<sup>4</sup><https://omarelb.github.io/self-explaining-neural-networks/>

The reproducibility project was more difficult to launch remotely. Since students had done online classes for their entire first semester, some struggled to find a group of fellow students to team up with, especially those coming from outside the MSc AI program. Overall, while we had various communication means, the lack of physical interaction due to COVID-19 slowed down our course organization.

### 5.6 Feedback

---

In this section, we discuss the feedback we received about the course from the perspective of participating students (Section 5.6.1) and from the ML Reproducibility Challenge reviews (Section 5.6.2).

#### 5.6.1 Feedback from Students

Both iterations of the course were evaluated using the standard evaluation procedure provided by the University of Amsterdam. However, only 16% of students filled out the evaluation form (23 out of 144) in the 2020–2021 iteration, potentially because the evaluation forms were administered online instead of in-person. According to the evaluation procedure at our university, this is not enough for a reliable quantitative estimate of student satisfaction. Therefore, we focus on the 2019–2020 iteration when reporting student satisfaction statistics, since 53% of students filled out the form (79 out of 149) that year. The vast majority of students were (very) satisfied with the course overall (67.8%). More specifically, students expressed satisfaction with the following dimensions:

- Academic challenge: 75.2% were (very) satisfied
- Supervision: 76.9% were (very) satisfied
- Feedback: 81.3% were (very) satisfied
- Workload: 91.3% were (very) satisfied
- Level of the course: 79.7% were (very) satisfied
- Level of the report: 94.8% were (very) satisfied
- Level of the presentation: 96.6% were (very) satisfied

Table 5.2(a) shows some of the qualitative feedback we received from students. Based on this, we believe these high scores are mostly the result of the reproducibility project. Students enjoyed doing the project, especially due to the intensive supervision from our experienced TAs. The dimensions where we received the lowest scores were on the lectures and the final presentation, where only 30.6% and 30.2% were (very) satisfied with these aspects, respectively. This may be because we only provided four (high-level) lectures on each of the four topics, in order to give students as much time as possible to focus on the reproducibility project. However, it should be noted that the overall scores for these components were not poor, but average: 3.1/5 for lectures and 3.0/5 for the presentation.



Table 5.2: Feedback about the course.

---

(a) Feedback from students

---

- “Reproducing an article was hard and intensive but a really good experience.”
  - “Replicating another study, seeing how (poorly) other research is performed was really eye-opening.”
  - “Reproducing a paper: I believe this is a good thing to do and is an important part of academia.”
  - “Gave good insights into the trustworthiness of research papers, which is apparently not great.”
  - “I appreciate the critical view I have developed on papers as a result of this course. Normally I would easily accept the content of a paper, but I will be more critical from now on, as many papers are not reproducible.”
  - “I think it’s really good that we get some practical insights into reproducing results from other papers, not all papers are as good as they seem to be.”
  - “I really appreciated that this was the first course where students are judging state-of-the-art AI-models. In other words, students were able to experience the scientific workfield of AI.”
- 

(b) Feedback from the ML Reproducibility Challenge

---

- “The report reveals a lot of dark spots of the original paper.”
  - “Good reviews, strong reproducibility report, provides code reimplementation from scratch which is a strong contribution.”
  - “The discussion section is a great reference point for future work.”
  - “The additional experimentation is rather impressive and the report reflects an intuitive understanding of concepts such as coverage, correctness, and counterfactual explanations.”
  - “The report provides good insights on how the experiments in the original paper actually work, while also generating new hypothesis to be tested for future research, which is a positive outcome.”
  - “My main concern is that it remains unclear why some of the results are so far off from the original paper? I would have expected the authors to dig deeper on that.”
  - “It doesn’t go above and beyond the reproduction and does not offer novel insights into the workings of the original paper.”
  - “The submission failed at reproducing the original results. It is unclear whether this is due to a difference in the experimental setup or due to implementation errors. ”
-

### 5.6.2 Feedback from the ML Reproducibility Challenge

Of the 30 reproducibility reports submitted to the ML Reproducibility Challenge in the 2020–2021 iteration, 9 were accepted for publication in the ReScience Journal. In total, the ML Reproducibility Challenge accepted 23 reports, meaning that almost 40% of the reports accepted to the challenge were from the University of Amsterdam.<sup>5</sup>

The reviews were mostly positive, with the general consensus being that most teams had gone beyond the general expectation of simply re-implementing the algorithm and re-running the experiments. Our TAs encouraged students to examine the generalizability of the work that was reproduced, either by trying new datasets or hyperparameters, or by performing ablation studies. Multiple reproducibility reports managed to question the results of the original papers with experimentally-supported claims. Importantly, some reviewers emphasized that these reproducibility studies were solid starting points for future research projects. For the reports that were rejected, the main critiques were that (i) only a fraction of the original work was reproduced, or (ii) no new insights were given. Some projects also had flaws in the experimental setup. See Table 5.2(b) for quotes from the ML Reproducibility Challenge reviews.

## 5.7 Factors Contributing to a Successful Course

---

Understanding and re-implementing the work of other researchers is not a trivial task, especially for first-year MSc students. There were several aspects of the setup that we believe were beneficial for the students, which we organize along three dimensions: (i) general, (ii) concerning FACT-AI, and (iii) concerning reproducibility. We believe each of these factors is important for a successful implementation of this course, or other similar courses.

### 5.7.1 General

#### Timing of the course

It is important that students have prior knowledge of ML theory as well as some programming experience before completing a project-based course in groups. At the University of Amsterdam, the FACT-AI course takes place after students have completed 4 ML-focused courses (see Table 5.1). We believe it is extremely important that students have access to adequate preparation, especially in terms of programming experience, before setting off to reproduce experiments from prominent AI papers. Without this prior knowledge, we believe such a project would not be feasible in the allotted time frame.

---

<sup>5</sup>[https://openreview.net/group?id=ML\\_Reproducibility\\_Challenge/2020](https://openreview.net/group?id=ML_Reproducibility_Challenge/2020)

### **Regular contact with experienced TAs**

The TAs are there to help with two main components: (i) understanding the paper, and (ii) debugging the implementation process. In practice, we found that it is extremely important for the TAs to have excellent programming experience since this is the main aspect students need help with. We also had a dedicated Slack workspace for the TAs and course instructors to keep in touch regularly.

Since our course is only four weeks long, we found it was important for students to have regular contact with their TAs to ensure no one got stuck in the process. For the first (in-person) iteration of the course, groups had one-hour tutorials with their TAs twice per week, where all groups that were working on the same paper (and therefore had the same TA) were in the same tutorial. Since they were all working on the same paper, there were many overlapping questions, and students found it beneficial to be able to share their experiences with one another. For the second (online) iteration of the course, we thought it would be challenging to ensure each group got the attention they needed if everyone was in one large online tutorial, so the TAs met with each group separately for 30 minutes, twice per week.

### **Early feedback on the reports**

Approximately halfway through the course, we asked students to submit a draft report to their TAs in order to get feedback. We found this significantly increased the quality of the final reports.

## **5.7.2 Concerning FACT-AI**

### **Emphasizing the technical perspective of FACT-AI**

Given that the FACT-AI course is situated in the context of a technical, research-oriented MSc, having students re-implement research papers from top AI conferences was an effective way to teach FACT-AI topics for our students. Teaching FACT-AI from a primarily technical perspective aligns well with what students expect from the MSc AI program at the University of Amsterdam. Although we believe a technical focus makes sense for our MSc program, we also believe it is important to incorporate non-technical perspectives into the course – see Section 5.8.2.

### **Creating resources for the FACT-AI community**

We believe a significant motivating factor for students was creating concrete output that extended beyond simply completing a project for a course: creating resources for the FACT community. In the 2019–2020 iteration, this was done by creating a public repository with the best FACT-AI algorithm implementations, as selected by the TAs. In the 2020–2021 iteration, this was done by publicly submitting their reproducibility reports about FACT-AI algorithms to the ML Reproducibility Challenge, where the accepted reports were published in the

ReScience Journal. In the future, we plan to continue aligning our course with the ML Reproducibility Challenge since we found the process extremely beneficial for our students.

### 5.7.3 Concerning Reproducibility

#### Including extension as part of reproducibility

If source code was already available for the paper – which is fortunately becoming increasingly common for AI research papers – we asked students to think about how to extend the paper since the implementation was already available. This resulted in some creative and interesting ideas in the reports, and we believe this is why our students performed well at the ML Reproducibility Challenge.

#### Simple grading setup

For a 4-week, project-based course, we found it was beneficial for students to focus on one main deliverable consisting of three components: (i) the reproducibility report, (ii) the associated code base, and (iii) the group presentation. The report that students submitted for the course was the same one they submitted to the ML Reproducibility Challenge. This way, participating in the ML Reproducibility Challenge was not an extra task but rather an integral part of the course.

## 5.8 Areas of Improvement

---

Although we believe both iterations of the course went well, there are several aspects of the setup that we believe could use some improvement and other instructors should consider if they plan to implement a similar course.

### 5.8.1 General

Given that this is the first time most students are formally submitting a paper, it is not surprising that there were some logistical issues. Some groups made minor mistakes such as forgetting to submit their work double-blind or slightly missing the submission deadline. We also had some groups who wrote the introduction sections of their papers as an introduction to the FACT-AI course, rather than an introduction to the topic they were working on. In future iterations, we will explicitly state the standard procedures of writing and submitting a paper and provide some examples.

### 5.8.2 Concerning FACT-AI

Although focusing primarily on the technical aspects of FACT-AI is an effective way to engage our technical students in socially-relevant AI problems, we also believe that they would benefit from additional non-technical perspectives on FACT-AI topics. In the future, we plan to include perspective from outside of

computer science through (i) additional guest lectures, (ii) workshop sessions [117, 120], and (iii) broader impact statements [19] in the reproducibility reports.

### 5.8.3 Concerning Reproducibility

In future iterations, we believe it would be useful to show students more examples of what a high-quality reproducibility paper looks like and explain in-depth why it is high-quality. These could be papers that were previously accepted to the ML Reproducibility Challenge, or papers from other reproducibility efforts outlined in Section 5.3. We want the students to understand what makes a paper a good (reproducibility) paper, that is, it has a set of (reproducibility) claims, it argues for these claims, and shows evidence to support these claims.

## 5.9 Conclusion

---

In this chapter, we share our setup for the FACT-AI course at the University of Amsterdam, which teaches FACT-AI topics through reproducibility. The course set out to give students (i) an understanding of FACT-AI topics, (ii) an understanding of algorithmic harm, (iii) familiarity with recent FACT-AI methods, and (iv) an opportunity to reproduce FACT-AI solutions, through a combination of lectures, paper discussion sessions and a reproducibility project. Through their projects, our students engaged with the open-source community by creating a public code repository (in the 2019–2020 iteration), as well as with the research community via successful submissions to the ML Reproducibility Challenge (in the 2020–2021 iteration). We also detail how the 2020–2021 iteration brought about its own unique set of challenges due to the COVID-19 pandemic.

In this course, we illustrate that reproducibility should be viewed as a fundamental component of FACT-AI. We received very positive feedback on teaching FACT-AI topics through reproducibility. We believe this was an excellent fit for our students, which not only helped motivate them for the duration of the course, but also helped them develop skills that will be essential in their future research careers, whether in the private or public sector.

With this final chapter, we answer **RQ4**: we can use reproducibility as a mechanism for teaching responsible AI concepts to a technical, research-oriented audience. Structuring the course around a reproducibility project gives students the opportunity to learn about responsible AI concepts, such as explainability, in a hands-on manner. Since the publication of the paper on which this chapter is based [83], we ran another iteration of the FACT-AI course in 2021–2022 under the same setup as the previous year, where students submitted their reports to the 2022 edition of the ML Reproducibility Challenge. 21 of the 43 papers accepted to the ML Reproducibility Challenge in 2022 were from students in the FACT-AI course. These also included some awards: the Best Paper Award was awarded to a group from the FACT-AI course as well as 2 of the 4 Outstanding Paper Awards. We believe this indicates that our course setup can serve as a starting point for effective participation in the broader ML research community.



---

# Conclusions

---

In this thesis, we have investigated explainability in ML from three viewpoints: (i) algorithms, (ii) users, and (iii) pedagogy. In this final chapter of the thesis, we revisit the research questions from Chapter 1, state our main findings in Section 6.1, and identify directions for future work in Section 6.2.

## 6.1 Main Findings

---

In this section, we describe our main findings across the three parts of the thesis.

### 6.1.1 Algorithms

The first part of this thesis focused on investigating behavior-based explanations in order to explain individual predictions from specific types of ML models. In Chapter 2, we asked our first research question:

**RQ1** Can we generate counterfactual explanations for tree-based models using gradient-based optimization?

The answer to **RQ1** is yes: we are able to explain predictions for tree ensembles in a counterfactual manner by including differentiable approximations of tree-based models within a standard gradient-based optimization framework. In the majority of experimental settings, our method outperforms existing baselines in terms of (i) the number of counterfactual examples produced, (ii) the average distance between the counterfactual examples and the original examples, and (iii) the proportion of counterfactual examples that are closer to the original examples. Our method is flexible since it can produce different types of counterfactual explanations depending on which distance function we choose to include in the loss function. In practice, this allows the user to customize the explanations depending on the use case.

We then turned to our next research question:

**RQ2** Can we extend our counterfactual explanation method for tree-based models to graph-based models?

The answer to **RQ2** is yes: we can adapt our method from **RQ1** to the graph setting by introducing a binary perturbation matrix that is multiplied element-wise with the adjacency matrix in order to remove edges from the graph. Since

this was one of the first methods for generating counterfactual explanations for GNNs, we also had to design a corresponding experimental setup to evaluate it. In the majority of experimental settings, we found that our method outperformed the baselines in terms of (i) accuracy, (ii) the number of counterfactual examples produced, (iii) the number of edges removed, and (iv) the proportion of the subgraph neighborhood that was perturbed.

### 6.1.2 Users

In the second part of this thesis, we continued our work on behavior-based explanations, but shifted our predominantly algorithmic focus to also account for the users who consume the explanations. We investigated the following research question:

**RQ3** Given a real-world use case, can we create an explanation method based on this use case and evaluate it in a context-specific manner?

The answer to **RQ3** is yes: we developed an explanation method based on the needs of real-world analysts in order to help them understand large errors in sales forecasting predictions. We designed a user study to evaluate our method and found that for the vast majority of users, our explanations were both interpretable and actionable. We also found that most users believed the explanations helped them understand large errors in predictions, but they did not have an impact on other aspects such as trust or confidence in the model.

### 6.1.3 Pedagogy

In the third part of this thesis, we transitioned from creating knowledge about ML model predictions to communicating knowledge about responsible ML practices. Process-based explanations play an important role here, specifically in the context of documentation practices which help ensure research is conducted in a responsible and reproducible manner. We asked our final research question:

**RQ4** How can we teach about responsible AI topics to a technical, research-oriented audience?

We answered **RQ4** by developing a course that was centered on a reproducibility project, where students worked in groups to reimplement algorithms from top-AI conferences on responsible AI topics. We shared our experiences with teaching the course over two academic years and suggested best practices for implementing similar reproducibility courses in the future.

## 6.2 Future Directions

---

In this section, we describe some limitations of the methods proposed in this thesis and identify potential avenues for future work.



### Limitations of counterfactual explanations

Researchers have raised concerns about the hidden assumptions behind the use of counterfactual examples [8], as well as potentials for misuse [65]. When explaining ML models using counterfactual examples, it is important to account for the context in which the systems are deployed. Counterfactual explanations are not a guarantee to achieving recourse [136] – changes suggested should be seen as candidate changes, not absolute solutions, since what is pragmatically actionable differs depending on the end user and context. While existing research from the cognitive sciences has shown that humans are able to interpret counterfactual explanations, the notion of what constitutes a *minimal* perturbation is not clear [18]. Further research into the interpretability and cognitive efficacy of counterfactual explanations could help the field better understand the appropriate criteria to optimize for.

### Accommodating different types of perturbations

In Chapter 3, we proposed a method for generating counterfactual explanations for GNNs. In its current form, our method is limited to performing edge deletions for node classification tasks. Given that many graph datasets also include node features, future work should involve incorporating node feature perturbations in our framework. We could also extend our method to accommodate graph classification tasks.

### Including additional criteria in loss functions

In Chapter 2, we proposed a method for generating flexible counterfactual explanations for tree-based models using gradient optimization techniques. The flexibility comes from varying the distance function used in the loss function, which results in different types of counterfactual explanations depending on which distance function is chosen. Future work could involve trying alternative distance functions or including additional criteria in the loss function, such as proximity to other points in the dataset or stability of the counterfactual example. This could also be applied to the method proposed in Chapter 3.

### Evaluating with users

Although the counterfactual explanations proposed in Chapters 2 and 3 perform well on various distance metrics, we should conduct a user study to evaluate how useful they are in practice. We could build on our existing user study design from Chapter 4 to test how (i) varying the distance functions, and (ii) introducing new components into the loss functions impacts user preferences for counterfactual explanations.

### **Improving trust in explanations**

In Chapter 4, we proposed a method for explaining errors in forecasting predictions based on identifying unusual feature values. We find that although explanations from our method help users understand large errors in predictions, they do not have an impact on users' trust, deployment support, or perception of the model's performance. Future work could place more emphasis on trying to improve these aspects, for example by allowing a predictive model to abstain from prediction when a particular instance has unusual feature values beyond a certain threshold.

### **Developing robust protocols for XAI evaluation**

In general, we believe it is crucial for the ML community to invest in developing more rigorous evaluation protocols for XAI methods, both in terms of user studies as well as formal metrics. The XAI community could pursue collaborations with researchers from human-computer interaction to design human-centered user studies about evaluating the utility of XAI methods in practice. To design metrics, the XAI community could try borrowing ideas from information theory or collaborating with ML evaluation researchers in order to ensure that the explanations we generate are truly representative of the model's behavior.

### **Teaching reproducibility as a fundamental component of ML research**

Reproducibility mechanisms such as checklists and challenges can help promote reproducible research practices, but we do not believe they alone are enough to cause a shift in the ML community. We believe the key to fostering reproducible research starts in the classroom. It is important to teach the next generation of ML researchers that reproducibility is not an afterthought, but rather a fundamental component of conducting ML research responsibly. In addition to conducting reproducibility projects, we could also introduce reproducibility components in programming assignments across all courses within an ML study program.

### **Identifying consistent terms for explainability**

Due to growing collection of XAI literature, there are many definitions for various distinct but related concepts such as explainability, interpretability, transparency, and intelligibility. As a community, we should make an effort to standardize the terms we use in order to facilitate easier communication, especially with researchers from non-ML disciplines. Developing XAI that is useful in practice requires interdisciplinary collaboration, which is more straightforward if we can all speak the same language. This could be achieved through a workshop-style event with researchers working on XAI to consolidate a standardized terminology.

### Final thoughts

Overall, our main advice for future work is to continue prioritizing explainability in the ML community, whether it is *behavior-based* or *process-based*. For both types of explanations, we should explore developing explainability techniques that cater to different types of users with varying levels of granularity, as well as robust mechanisms for evaluation. As a community, we need to prioritize both correctness and interpretability of explanations – incorrect explanations that are interpretable do not provide the user with any concrete information, and correct explanations that are uninterpretable are not useful to the user. To promote correctness, we need to first identify what it means for an explanation to be “correct” and create datasets that allow us to explore this task explicitly. To promote interpretability, we need to approach the explainability problem from an interdisciplinary perspective, and suggest that XAI researchers spend more time connecting to the communities they are designing the explanations for.



---

# Bibliography

---

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. *ACM SIGSAC Conference on Computer and Communications Security*, 2016. (Cited on page 74.)
- [2] ACM. Artifact Review and Badging. <https://www.acm.org/publications/policies/artifact-review-badging>, 2019. (Cited on page 71.)
- [3] Ahmed M. Alaa, Michael Weisz, and Mihaela van der Schaar. Deep Counterfactual Networks with Propensity-Dropout. *ICML Workshop on Principled Approaches to Deep Learning*, 2017. (Cited on page 13.)
- [4] Dario Amodei and Danny Hernandez. AI and Compute. <https://openai.com/blog/ai-and-compute/>, 2018. (Cited on page 1.)
- [5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>, May 2016. (Cited on page 68.)
- [6] Federico Baldassarre and Hossein Azizpour. Explainability Techniques for Graph Convolutional Networks. *ICML Workshop on Learning and Reasoning with Graph-Structured Representations*, 2019. (Cited on pages 30 and 31.)
- [7] Randall Balestriero. Neural Decision Trees. *arXiv*, 2017. (Cited on page 14.)
- [8] Solon Barocas, Andrew D. Selbst, and Manish Raghavan. The Hidden Assumptions Behind Counterfactual Explanations and Principal Reasons. *ACM Conference on Fairness, Accountability, and Transparency*, 2020. (Cited on page 85.)
- [9] Jo Bates, David Cameron, Alessandro Checco, Paul Clough, Frank Hopfgartner, Suvodeep Mazumdar, Laura Sbaffi, Peter Stordy, and Antonio de la Vega de León. Integrating FATE/Critical Data Studies into Data Science Curricula: Where Are We Going and How Do We Get There? *ACM Conference on Fairness, Accountability, and Transparency*, 2020. (Cited on page 69.)
- [10] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational Inductive Biases, Deep Learning, and Graph Networks. *arXiv*, 2018. (Cited on page 32.)
- [11] Alina Beygelzimer, Yann Dauphin, Percy Liang, and Jennifer Wortman-Vaughan. Introducing the NeurIPS 2021 Paper Checklist. Medium: <https://neuripsconf.medium.com/introducing-the-neurips-2021-paper-checklist>, 2021. (Cited on page 71.)
- [12] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. Explainable Machine Learning in Deployment. *ACM Conference on Fairness, Accountability, and Transparency*, 2020. (Cited on page 54.)
- [13] Battista Biggio and Fabio Roli. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Pattern Recognition*, 2018. (Cited on page 13.)
- [14] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and Survey of Explanation Methods for Black Box Models. *arXiv*, 2021. (Cited on page 30.)
- [15] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF:

- Identifying Density-Based Local Outliers. *ACM SIGMOD Record*, 2000. (Cited on page 23.)
- [16] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial Patch. *Advances in Neural Information Processing Systems*, 2018. (Cited on page 13.)
- [17] Emanuelle Burton, Judy Goldsmith, and Nicholas Mattei. How to Teach Computer Ethics through Science Fiction. *Communications of the ACM*, 2018. (Cited on page 70.)
- [18] Ruth M.J. Byrne. Counterfactual Thought. *Annual Review of Psychology*, 2016. (Cited on page 85.)
- [19] Rosie Campbell, Madhulika Srikumar, and Hudson Hongo. Managing the Risks of AI Research: Six Recommendations for Responsible Publication. Partnership on AI, <https://www.partnershiponai.org/wp-content/uploads/2021/05/PAI-Managing-the-Risks-of-AI-Research-Responsible-Publication.pdf>, 2021. (Cited on pages 68 and 81.)
- [20] Davide Castelvecchi. Can We Open the Black Box of AI? *Nature*, 2016. (Cited on page 1.)
- [21] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine Learning on Graphs: A Model and Comprehensive Taxonomy. *arXiv*, 2021. (Cited on page 32.)
- [22] Jack Clark and Ray Perrault. Artificial Intelligence Index Report 2022. <https://hai.stanford.edu/research/ai-index-2022>, 2022. (Cited on page 1.)
- [23] Andrea Danyluk, Paul Leidig, Andrew McGettrick, Lillian Cassel, Maureen Doyle, Christian Servin, Karl Schmitt, and Andreas Steffik. Computing Competencies for Undergraduate Data Science Programs: An ACM Task Force Final Report. *ACM Technical Symposium on Computer Science Education*, 2021. (Cited on page 69.)
- [24] Kim de Bie, Ana Lucic, and Hinda Haned. To Trust or Not to Trust a Regressor: Estimating and Explaining Trustworthiness of Regression Predictions. *ICML Workshop on Human in the Loop Learning*, 2021.
- [25] Andreea Deac, Yu-Hsiang Huang, Petar Veličković, Pietro Liò, and Jian Tang. Drug-Drug Adverse Effect Prediction with Graph Co-Attention. *ICML Workshop on Computational Biology*, 2019. (Cited on page 30.)
- [26] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Compounds. Correlation with Molecular Orbital Energies and Hydrophobicity. *Journal of Medicinal Chemistry*, 1991. (Cited on page 37.)
- [27] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. *Advances in Neural Information Processing Systems*, 2018. (Cited on pages 13 and 51.)
- [28] Berkeley J. Dietvorst, Joseph P. Simmons, and Cade Massey. Algorithm Aversion: People Erroneously Avoid Algorithms After Seeing Them Err. *Journal of Experimental Psychology*, 2015. (Cited on pages 47 and 48.)
- [29] Finale Doshi-Velez and Been Kim. *Considerations for Evaluation and Generalization in Interpretable Machine Learning*. Springer International Publishing, 2018. (Cited on pages 11, 37, 47, and 57.)
- [30] Alexandre Duval and Fragkiskos D. Malliaros. GraphSVX: Shapley Value Explanations for Graph Neural Networks. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2021. (Cited on pages 30 and 31.)
- [31] Upol Ehsan, Q. Vera Liao, Michael Muller, Mark O. Riedl, and Justin D. Weisz. Expanding explainability: Towards social transparency in ai systems. *CHI Conference on Human Factors in Computing Systems*, 2021. (Cited on page 4.)
- [32] EU. Regulation (EU) 2016/679 of the European Parliament (GDPR). *Official Journal of the European Union*, 2016. (Cited on pages 11, 29, and 47.)
- [33] Lukas Faber, Amin K Moghaddam, and Roger Wattenhofer. Contrastive Graph Neural Network Explanation. *ICML Workshop on Graph Representation Learning and Beyond*, 2020. (Cited on page 32.)
- [34] FICO. Explainable Machine Learning Challenge. <https://community.fico.com/s/explainable-machine-learning-challenge>, 2017. (Cited on pages 18 and 48.)

- 
- [35] Casey Fiesler, Natalie Garrett, and Nathan Beard. What Do We Teach When We Teach Tech Ethics? A Syllabi Analysis. *ACM Technical Symposium on Computer Science Education*, 2020. (Cited on page 69.)
  - [36] Timo Freiesleben. The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds and Machines*, 2021. (Cited on page 32.)
  - [37] Heidi Furey and F. Martin. Introducing Ethical Thinking About Autonomous Vehicles Into an AI Course. *AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018. (Cited on page 70.)
  - [38] Natalie Garrett, Nathan Beard, and Casey Fiesler. More Than “If Time Allows”: The Role of Ethics in AI Education. *AAAI/ACM Conference on AI, Ethics, and Society*, 2020. (Cited on page 69.)
  - [39] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets. *Communications of the ACM*, 2021. (Cited on page 2.)
  - [40] Randy Goebel, Ajay Chander, Katharina Holzinger, Freddy Lecue, Zeynep Akata, Simone Stumpf, Peter Kieseberg, and Andreas Holzinger. Explainable AI: The New 42? *International IFIP Cross Domain Conference for Machine Learning and Knowledge Extraction*, 2018. (Cited on page 29.)
  - [41] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations*, 2015. (Cited on page 13.)
  - [42] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. Interpretable Credit Application Predictions With Counterfactual Explanations. *NeurIPS Workshop on Challenges and Opportunities for AI in Financial Services*, 2018. (Cited on page 13.)
  - [43] Nancy Green. An AI Ethics Course Highlighting Explicit Ethical Agents. *AAAI/ACM Conference on AI, Ethics, and Society*, 2021. (Cited on page 70.)
  - [44] Nancy Green and Larry Crotts. Argument Schemes for AI Ethics Education. *COMMA Workshop on Computational Models of Natural Argument*, 2020. (Cited on page 70.)
  - [45] Barbara J. Grosz, David Gray Grant, Kate Vredenburg, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. Embedded EthiCS: Integrating Ethics across CS Education. *Communications of the ACM*, 2019. (Cited on page 69.)
  - [46] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local Rule-Based Explanations of Black Box Decision Systems. *arXiv*, 2018. (Cited on page 32.)
  - [47] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 2018. (Cited on pages 1, 2, 30, and 54.)
  - [48] Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V. Chawla. Few-Shot Graph Learning for Molecular Property Prediction. *The Web Conference*, 2021. (Cited on page 30.)
  - [49] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. *Advances in Neural Information Processing Systems*, 2016. (Cited on page 74.)
  - [50] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding Visual Explanations. *European Conference on Computer Vision*, 2018. (Cited on page 51.)
  - [51] Dennis J. Hilton. Conversational Processes and Causal Explanation. *Psychological Bulletin*, 1990. (Cited on page 54.)
  - [52] Dennis J. Hilton. Social Attribution and Explanation. *The Oxford Handbook of Causal Reasoning*, 2017. (Cited on page 55.)
  - [53] Dennis J. Hilton and Ben R. Slugoski. Knowledge-based Causal Attribution: The Abnormal Conditions Focus Model. *Psychological Review*, 1986. (Cited on page 48.)
  - [54] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *NeurIPS Workshop on Deep Learning*, 2014. (Cited on page 14.)
  - [55] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *arXiv*, 2020. (Cited on page 32.)
  - [56] Matthew Hutson. Artificial Intelligence Faces Reproducibility Crisis. *Science*, 2018.
-

- (Cited on pages 68 and 71.)
- [57] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems. *International Conference on Artificial Intelligence and Statistics*, 2020. (Cited on page 13.)
  - [58] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. *International Joint Conference on Artificial Intelligence*, 2020. (Cited on pages 11, 13, 18, 19, 23, and 32.)
  - [59] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. ExplaiNE: An Approach for Explaining Network Embedding-based Link Predictions. *arXiv*, 2019. (Cited on page 32.)
  - [60] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. Model-Agnostic Counterfactual Explanations for Consequential Decisions. *International Conference on Artificial Intelligence and Statistics*, 2020. (Cited on pages 13, 14, and 30.)
  - [61] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A Survey of Algorithmic Recourse: Definitions, Formulations, Solutions, and Prospects. *arXiv*, 2020. (Cited on pages 3, 30, and 32.)
  - [62] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic Recourse Under Imperfect Causal Knowledge: A Probabilistic Approach. *Advances in Neural Information Processing Systems*, 2020. (Cited on page 13.)
  - [63] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic Recourse: From Counterfactual Explanations to Interventions. *ACM Conference on Fairness, Accountability, and Transparency*, 2021. (Cited on pages 4 and 13.)
  - [64] Surya Karunagaran, Ana Lucic, and Christine Custis. XAI Toolsheets: An Evaluation Framework for XAI Tools. *IJCAI Workshop on Explainable AI*, 2022.
  - [65] Atoosa Kasirzadeh and Andrew Smart. The Use and Misuse of Counterfactuals in Ethical Machine Learning. *ACM Conference on Fairness, Accountability, and Transparency*, 2021. (Cited on page 85.)
  - [66] Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer Credit-risk Models via Machine Learning Algorithms. *Journal of Banking and Finance*, 2010. (Cited on page 48.)
  - [67] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015. (Cited on page 18.)
  - [68] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations*, 2017. (Cited on page 34.)
  - [69] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. *International Conference on Machine Learning*, 2017. (Cited on page 1.)
  - [70] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detryniecki. Inverse Classification for Comparison-based Interpretability in Machine Learning. *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2018. (Cited on pages 13, 14, and 32.)
  - [71] Sabina Leonelli. Locating Ethics in Data Science: Responsibility and Accountability in Global and Distributed Knowledge Production Systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2016. (Cited on page 69.)
  - [72] Armanda Lewis and Julia Stoyanovich. Teaching Responsible Data Science: Charting New Pedagogical Territory. *International Journal of Artificial Intelligence in Education*, 2021. (Cited on pages 69 and 70.)
  - [73] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. *AAAI Conference on Artificial Intelligence*, 2018. (Cited on page 1.)
  - [74] Wanyu Lin, Hao Lan, and Baochun Li. Generative Causal Explanations for Graph Neural Networks. *International Conference on Machine Learning*, 2021. (Cited on pages 30 and 31.)
  - [75] Peter Lipton. Contrastive Explanation. *Royal Institute of Philosophy Supplement*, 1990. (Cited on page 54.)
  - [76] Zachary C Lipton and Jacob Steinhardt. Research for Practice: Troubling Trends in



- 
- Machine-Learning Scholarship. *Communications of the ACM*, 2019. (Cited on page 71.)
- [77] Ana Lucic, Hinda Haned, and Maarten de Rijke. Explaining Predictions from Tree-based Boosting Ensembles. *SIGIR Workshop on Fairness, Accountability, Confidentiality, and Transparency in Information Retrieval*, 2019.
- [78] Ana Lucic, Hinda Haned, and Maarten de Rijke. Why Does My Model Fail? Contrastive Local Explanations for Retail Forecasting. *ACM Conference on Fairness, Accountability, and Transparency*, 2020. (Cited on pages 32 and 47.)
- [79] Ana Lucic, Madhulika Srikumar, Umang Bhatt, Alice Xiang, Ankur Taly, Q. Vera Liao, and Maarten de Rijke. A Multistakeholder Approach Towards Evaluating AI Transparency Mechanisms. *CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI*, 2021.
- [80] Ana Lucic, Sheeraz Ahmad, Amanda Furtado Brinhosa, Q. Vera Liao, Himani Agrawal, Umang Bhatt, Krishnaram Kenthapadi, Alice Xiang, Maarten de Rijke, and Nicholas Drabowski. Towards the Use of Saliency Maps for Explaining Low-Quality Electrocardiograms to End Users. *IJCAI Workshop on Explainable AI*, 2022.
- [81] Ana Lucic, Maurits Bleeker, Samarth Bhargav, Jessica Zosa Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, and Robert Stojnic. Towards Reproducible Machine Learning Research in Natural Language Processing. *ACL Tutorial on Reproducibility*, 2022.
- [82] Ana Lucic, Maurits Bleeker, Maarten de Rijke, Koustuv Sinha, Sami Jullien, and Robert Stojnic. Towards Reproducible Machine Learning Research in Information Retrieval. *SIGIR Tutorial on Reproducibility*, 2022.
- [83] Ana Lucic, Maurits Bleeker, Sami Jullien, Samarth Bhargav, and Maarten de Rijke. Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence. *AAAI Symposium on Educational Advances in Artificial Intelligence*, 2022. (Cited on pages 67 and 81.)
- [84] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles. *AAAI Conference on Artificial Intelligence*, 2022. (Cited on pages 11, 32, 33, and 34.)
- [85] Ana Lucic, Maartje ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on pages 4 and 29.)
- [86] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 2017. (Cited on pages 1, 31, 37, 50, and 52.)
- [87] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature Machine Intelligence*, 2020. (Cited on page 50.)
- [88] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized Explainer for Graph Neural Network. *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 30, 31, 37, and 39.)
- [89] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable Reinforcement Learning Through a Causal Lens. *AAAI Conference on Artificial Intelligence*, 2019. (Cited on page 13.)
- [90] Tim Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial Intelligence*, 2019. (Cited on pages 3, 29, and 54.)
- [91] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. *ACM Conference on Fairness, Accountability, and Transparency*, 2019. (Cited on page 2.)
- [92] Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Lulu.com, 2019. (Cited on page 39.)
- [93] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *ACM Conference on Fairness, Accountability, and Transparency*, 2020. (Cited on page 13.)
- [94] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. (Cited on page 1.)
-

- [95] Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. Order in the Court: Explainable AI Methods Prone to Disagreement. *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021. (Cited on page 73.)
- [96] Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. A Song of (Dis)agreement: Evaluating the Evaluation of Explainable Artificial Intelligence in Natural Language Processing. *International Conference on Hybrid Human-Artificial Intelligence*, 2022. (Cited on page 73.)
- [97] Cuong Q. Nguyen, Constantine Kreatsoulas, and Kim M. Branson. Meta-Learning GNN Initializations for Low-Resource Molecular Property Prediction. *ICML Workshop on Graph Representation Learning and Beyond*, 2020. (Cited on page 30.)
- [98] Dan Ofer. COMPAS Dataset. Kaggle: <https://www.kaggle.com/danofer/compass>, 2017. (Cited on page 19.)
- [99] Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, and Michael D. Ekstrand. Facts-ir: Fairness, accountability, confidentiality, transparency, and safety in information retrieval. *SIGIR Forum*, 2021. (Cited on page 1.)
- [100] Cathy O’Neil. The Ivory Tower Can’t Keep Ignoring Tech. The New York Times: <https://www.nytimes.com/2017/11/14/opinion/academia-tech-algorithms.html>, 2017. (Cited on pages 68 and 69.)
- [101] Evan M. Peck. The Ethical Engine: Integrating Ethical Design into Intro Computer Science. Medium, <https://medium.com/bucknell-hci/the-ethical-engine-integrating-ethical-design-into-intro-to-computer-science-4f9874e756af>, 2017. (Cited on page 69.)
- [102] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A Review of Novelty Detection. *Signal Processing*, 2014. (Cited on page 51.)
- [103] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability Methods for Graph Convolutional Neural Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. (Cited on pages 30 and 31.)
- [104] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and Actionable Counterfactual Explanations. *AAAI/ACM Conference on AI, Ethics, and Society*, 2020. (Cited on page 13.)
- [105] Inioluwa Deborah Raji and Joy Buolamwini. Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products. *AAAI/ACM Conference on AI, Ethics, and Society*, 2019. (Cited on page 3.)
- [106] Inioluwa Deborah Raji, Morgan Klaus Scheuerman, and Razvan Amironesei. You Can’t Sit With Us: Exclusionary Pedagogy in AI Ethics Education. *ACM Conference on Fairness, Accountability, and Transparency*, 2021. (Cited on page 69.)
- [107] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust You? Explaining the Predictions of Any Classifier. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. (Cited on pages 1, 31, 39, 51, and 74.)
- [108] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-Agnostic Interpretability of Machine Learning. *ICML Workshop on Human Interpretability in Machine Learning*, 2016. (Cited on page 47.)
- [109] Chris Russell. Efficient Search for Diverse Coherent Explanations. *ACM Conference on Fairness, Accountability, and Transparency*, 2019. (Cited on pages 13 and 51.)
- [110] Jeffrey Saltz, Michael Skirpan, Casey Fiesler, Micha Gorelick, Tom Yeh, Robert Heckman, Neil Dewar, and Nathan Beard. Integrating Ethics within Machine Learning Courses. *ACM Transactions on Computing Education*, 2019. (Cited on page 69.)
- [111] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019. (Cited on page 30.)
- [112] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. *International Conference on Learning Representations*, 2021. (Cited on pages 30 and 31.)
- [113] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. XAI for Graphs: Explaining Graph Neural Network Predictions by Identifying Relevant Walks. *arXiv*, 2020. (Cited on page 32.)

- 
- [114] Lisa Schut, Oscar Key, and Rory McGrath. Generating Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties. *International Conference on Artificial Intelligence and Statistics*, 2021. (Cited on page 30.)
  - [115] David Sculley, Jasper Snoek, Alexander B. Wiltschko, and Ali Rahimi. Winner's Curse? On Pace, Progress, and Empirical Rigor. *International Conference on Learning Representations*, 2018. (Cited on page 71.)
  - [116] Boris Sharchilev, Yury Ustinovsky, Pavel Serdyukov, and Maarten de Rijke. Finding Influential Training Samples for Gradient Boosted Decision Trees. *International Conference on Machine Learning*, 2018. (Cited on pages 1 and 50.)
  - [117] Hong Shen, Wesley H. Deng, Aditi Chattopadhyay, Zhiwei Steven Wu, Xu Wang, and Haiyi Zhu. Value Cards: An Educational Toolkit for Teaching Social Impacts of Machine Learning through Deliberation. *ACM Conference on Fairness, Accountability, and Transparency*, 2021. (Cited on pages 70 and 81.)
  - [118] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *ICML Workshop on Automated Machine Learning*, 2021. (Cited on page 4.)
  - [119] Natasha Singer. Tech's Ethical 'Dark Side': Harvard, Stanford and Others Want to Address It. The New York Times: <https://www.nytimes.com/2018/02/12/business/computer-science-ethics-courses.html>, 2018. (Cited on page 69.)
  - [120] Michael Skirpan, Nathan Beard, Srinjita Bhaduri, Casey Fiesler, and Tom Yeh. Ethics Education in Context: A Case Study of Novel Ethics Activities for the CS Classroom. *ACM Technical Symposium on Computer Science Education*, 2018. (Cited on pages 69, 70, and 81.)
  - [121] Suraj Srinivas, Akshayvarun Subramanya, and R. Venkatesh Babu. Training Sparse Neural Networks. *CVPR Workshops*, 2017. (Cited on pages 33 and 34.)
  - [122] Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access*, 2021. (Cited on pages 1, 3, 4, 30, and 32.)
  - [123] Robert Stojnic. Papers with Code partners with arXiv. Medium: <https://medium.com/paperswithcode/papers-with-code-partners-with-arxiv-ecc362883167>, 2020. (Cited on pages 71 and 75.)
  - [124] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A Deep Learning Approach to Antibiotic Discovery. *Cell*, 2020. (Cited on page 30.)
  - [125] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 2019. (Cited on page 13.)
  - [126] Lichao Sun, Yingdong Dou, Carl Yang, Ji Wang, Philip S. Yu, Lifang He, and Bo Li. Adversarial Attack and Defense on Graph Data: A Survey. *arXiv*, 2018. (Cited on page 32.)
  - [127] Thomas Douglas Victor Swinscow. *Statistics at Square One*. BMJ Publishing Group, 1997. (Cited on page 53.)
  - [128] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. *International Conference on Learning Representations*, 2014. (Cited on page 13.)
  - [129] Hui Fen Tan, Giles Hooker, and Martin T. Wells. Tree Space Prototypes: Another Look at Making Tree Ensembles Interpretable. *NeurIPS Workshop on Interpretable Machine Learning in Complex Systems*, 2016. (Cited on page 1.)
  - [130] Maartje ter Hoeve, Mathieu Heruer, Daan Odijk, Anne Schuth, Martijn Spitters, and Maarten de Rijke. Do News Consumers Want Explanations for Personalized News Rankings? *FATREC Workshop on Responsible Recommendation*, 2017. (Cited on page 58.)
  - [131] John Thornhill. Is AI Really that Intelligent? Financial Times, <https://www.ft.com/content/dd621de6-4047-46c3-833a-b21bd2423113>, 2021. (Cited on page 1.)
  - [132] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable
-

- Predictions of Tree-based Ensembles via Actionable Feature Tweaking. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017. (Cited on pages 11, 12, 13, 14, 18, 20, 32, and 50.)
- [133] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977. (Cited on page 52.)
- [134] UCI. Wine Quality Data Set. <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>, 2009. (Cited on page 18.)
- [135] UCI. Online Shoppers Intention Dataset. <https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset>, 2019. (Cited on page 19.)
- [136] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. *ACM Conference on Fairness, Accountability, and Transparency*, 2019. (Cited on pages 13, 30, 32, and 85.)
- [137] Ilse van der Linden, Hinda Haned, and Evangelos Kanoulas. Global Aggregations of Local Explanations for Black Box models. *SIGIR Workshop on Fairness, Accountability, Confidentiality, and Transparency in Information Retrieval*, 2019. (Cited on page 2.)
- [138] Arnaud Van Looveren and Janis Klaise. Interpretable Counterfactual Explanations Guided by Prototypes. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2021. (Cited on page 13.)
- [139] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual Explanations for Machine Learning: A Review. *arXiv*, 2020. (Cited on pages 1, 4, 30, and 32.)
- [140] Minh N. Vu and My T. Thai. PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 30 and 31.)
- [141] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law and Technology*, 2018. (Cited on pages 1, 3, 12, 13, 15, 32, 33, 34, and 51.)
- [142] Oliver Wieder, Stefan Kohlbacher, Méline Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A Compact Review of Molecular Property Prediction with Graph Neural Networks. *Drug Discovery Today: Technologies*, 2020. (Cited on page 30.)
- [143] Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. MARS: Markov Molecular Sampling for Multi-objective Drug Discovery. *International Conference on Learning Representations*, 2021. (Cited on page 30.)
- [144] Yongxin Yang, Irene Garcia Morillo, and Timothy M. Hospedales. Deep Neural Decision Trees. *ICML Workshop on Human Interpretability in Machine Learning*, 2018. (Cited on page 14.)
- [145] Burak Yildiz, Hayley Hung, Jesse H Krijthe, Cynthia CS Liem, Marco Loog, Gosia Migut, Frans A Oliehoek, Annibale Panichella, Przemysław Pawełczak, Stjepan Picek, et al. ReproducedPapers.org: Openly Teaching and Structuring Machine Learning Reproducibility. *International Workshop on Reproducible Research in Pattern Recognition*, 2021. (Cited on page 70.)
- [146] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GN-Explainer: Generating Explanations for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 2019. (Cited on pages 30, 31, 37, 38, and 39.)
- [147] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: Towards Model-Level Explanations of Graph Neural Networks. *arXiv*, 2020. (Cited on page 32.)
- [148] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in Graph Neural Networks: A Taxonomic Survey. *arXiv*, 2020. (Cited on pages 30, 31, and 39.)
- [149] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On Explainability of Graph Neural Networks via Subgraph Explorations. *International Conference on Machine Learning*, 2021. (Cited on pages 30 and 31.)
- [150] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *Bioinformatics*, 2018. (Cited on page 30.)

---

# Summary

---

Model explainability has become an important problem in artificial intelligence (AI) due to the increased effect that algorithmic predictions have on humans. Explanations can help users understand not only why AI models make certain predictions, but also how these predictions can be changed. In the first part of this thesis, we investigate counterfactual explanations: given a data point and a trained model, we want to find the minimal perturbation to the input such that the prediction changes. We frame the problem of finding counterfactual explanations as a gradient-based optimization task and first focus on tree ensembles. We extend previous work that could only be applied to differentiable models by incorporating probabilistic model approximations in the optimization framework, and find that our counterfactual examples are significantly closer to the original instances than those produced by other methods specifically designed for tree-based models.

We then extend our method for generating counterfactual explanations for tree ensembles to accommodate graph neural networks (GNNs), given the increasing promise of GNNs in real-world applications such as fake news detection and molecular simulation. We do so by introducing a perturbation matrix that acts on the adjacency matrix in order to iteratively remove edges from the graph, and find that our method primarily removes edges that are crucial for the original predictions, resulting in minimal counterfactual explanations.

In the second part of this thesis, we investigate explanations in the context of a real-world use case: sales forecasting. We propose an algorithm that generates explanations for large errors in forecasting predictions based on Monte Carlo simulations. To evaluate, we conduct a user study with 75 users and find that the majority of users are able to accurately answer objective questions about the model's predictions when provided with our explanations, and that users who saw our explanations understand why the model makes large errors in predictions significantly more than users in the control group. We also conduct an in-depth analysis of the difference in attitudes between practitioners and researchers, and confirm that our results hold when conditioning on the users' background.

In the final part of the thesis, we explain the setup for a technical, graduate-level course on responsible AI topics at the University of Amsterdam, which teaches responsible AI concepts through the lens of reproducibility. The focal point of the course is a group project based on reproducing existing responsible AI algorithms from top AI conferences and writing a corresponding report. We reflect on our experiences teaching the course over two years and propose guidelines for incorporating reproducibility in graduate-level AI study programs.



---

# Samenvatting

---

De uitlegbaarheid van voorspellende modellen is een belangrijk probleem geworden in kunstmatige intelligentie (KI) vanwege het toegenomen effect dat algoritmische voorspellingen hebben op mensen. Een uitleg kan gebruikers niet alleen helpen om te begrijpen waarom KI-modellen bepaalde voorspellingen doen, maar ook hoe deze voorspellingen beïnvloed kunnen worden. In het eerste deel van dit proefschrift onderzoeken we contrafeitelijke verklaringen: we willen, gegeven een datapunt en een getraind model, de minimale verandering van de input vinden die de voorspelling verandert. We formuleren het probleem van het vinden van contrafeitelijke verklaring als een op gradiënten gebaseerde optimalisatietask en richten ons eerst op *tree ensembles*. We bouwen voort op eerder werk dat alleen kon worden toegepast op differentieerbare modellen, door probabilistische modelbenaderingen op te nemen in het optimalisatiekader, en komen tot de bevinding dat onze contrafeitelijke voorbeelden significant dichter bij het oorspronkelijke datapunt liggen dan de voorbeelden die geproduceerd worden door andere methoden, die specifiek zijn ontworpen voor modellen die op bomen zijn gebaseerd.

Vervolgens breiden we onze methode voor het genereren van contrafeitelijke verklaringen uit voor *tree ensembles* zodat die ook werkt voor *graph neural networks* (GNNs), gezien de toenemende belofte van GNNs voor toepassingen in de echte wereld, zoals de detectie van nepnieuws en moleculaire simulatie. We bereiken dit door een perturbatiematrix te introduceren die de elementen uit de bogenmatrix vermenigvuldigt om iteratief zijden van de graaf te verwijderen, en komen tot de bevinding dat onze methode voornamelijk zijden verwijdert die cruciaal zijn voor de oorspronkelijke voorspelling, wat resulteert in een minimale contrafeitelijke verklaring.

In het tweede deel van dit proefschrift onderzoeken we verklaringen in de context van een praktijkvoorbeeld: verkoopprognoses. We introduceren een algoritme dat verklaringen genereert voor grote fouten bij het doen van regressievoorspellingen op basis van Monte Carlo-simulaties. We evalueren door middel van een gebruikersonderzoek met 75 deelnemers. We komen tot de bevinding dat de meerderheid van de gebruikers in staat is accuraat antwoord te geven op meerkeuzevragen over de voorspellingen van het model wanneer zij voorzien zijn van onze uitleg. Daarnaast komen we tot de bevinding dat gebruikers die onze uitleg zagen, significant vaker dan gebruikers uit de controlegroep begrijpen waarom het model grote fouten maakt in voorspellingen. We voeren ook een analyse uit van het verschil in attitudes tussen praktijkbeoefenaars en onderzoekers, en bevestigen dat onze resultaten stand houden gegeven de

achtergrond van de gebruiker.

In het laatste deel van dit proefschrift behandelen we de opzet van een technisch vak op masterniveau over verantwoord gebruik van KI, gegeven aan de Universiteit van Amsterdam. In dit vak worden verantwoorde KI-concepten vanuit het oogpunt van reproduceerbaarheid gedoceerd. Het speerpunt van de cursus is een groepsproject dat gebaseerd is op het reproduceren van bestaande verantwoorde KI-algoritmen van vooraanstaande KI-conferenties en het schrijven van een bijbehorend rapport. We reflecteren op onze ervaringen met het geven van de cursus gedurende twee jaar en stellen richtlijnen voor voor het opnemen van reproduceerbaarheid in KI-studieprogramma's op masterniveau.