UNIVERSITAT ROVIRA i VIRGILI

Universitat Oberta de Catalunya

Final Master's Degree Project

Master's Degree in Computational Engineering and Mathematics

An Agent-Based Simulation Study of the Social Contagion of Divorce

Jon Perez Visaires

Director:
Jordi Duch Gavaldà

Universitat Oberta de Catalunya
Universitat Rovira i Virgili

Bilbao, September 2023

Summary

An agent-base model based on an epidemiological SIRa model of infectious diseases is used for the simulation of the social contagion of divorce in this work. First, two simple SIR and SIRa models are implemented and the results of their simulation outcomes analyzed. Then, using machine learning classification models to predict the probability that each couple has of facing relationship instability, multiple modifications are introduced in the SIRa model. Having more realistic and heterogeneous agents in the agent-based simulation leads to more realistic and accurate results and less uncertainty in the outcomes of the simulation runs.

Keywords: Agent-based modeling, agent-based simulation, machine learning, social contagion, computational social science.

Contents

1 Introduction

Based on data from the National Center for Health Statistics of the United States [16], approximately 43 percent of marriages dissolve within the initial fifteen years. Furthermore, as of 2015, the average annual divorce rate in the European Union hovers around 2% [83]. These statistics hold significance due to the notable individual physical and mental health repercussions for those experiencing separation [88], as well as the potential impact on the development of children from divorced families [5]. However, these figures also raise questions about the potential existence of a separation "epidemic", and whether a mechanism known as social contagion plays a part in this phenomenon. While instances of apparent "mini-epidemics" among celebrity circles are widely known [30], the broader prevalence of such a pattern in everyday life warrants further examination.

A significant amount of sociological research focuses on understanding the factors influencing marriage and divorce. Within this body of research, some theoretical frameworks conceptualize marriage as a type of social exchange, where inner benefits (such as intimacy) and costs (like time) are measured against external costs (financial expenses) and benefits (social approval) [11]. From this point of view, external stressors imposed on a relationship, like financial difficulties, for instance, can increase the likelihood of divorce [94]. According to this type of model, the risk of separation escalates when the advantages of staying in a partnership dwindle compared to the associated costs, or when one or both partners perceive more attractive alternatives [4]. Aspects like available employment opportunities and the extent and nature of outside activities also impact the potential for finding viable alternative partners significantly [98].

When considering the intrinsic advantages and drawbacks of relationships, individuals often weigh these factors in comparison to their close social circle. Consequently, social norms regarding equity, commitment, or other relationship facets are likely to shape perceptions of a relationship's value and the acceptability of a potential separation. Operating from this perspective centered on individual-focused cost-benefit analysis, those encountering substantial drawbacks and limited rewards are more prone to separate [93]. However, these individual or partnership-oriented dyadic approaches often overlook the dynamics through which a couple's own social context and connections can also impact the stability of their relationships and their probability of separation.

In this work, the influence of separations within a couple's social network on their own likelihood of experiencing marital dissolution is explored. One conceivable scenario is that couples who go through separation might indirectly encourage the dissolution of partnership in others. They might do so by exemplifying that separation brings personal advantages or at least manageable consequences for the individuals involved. Additionally, they might also offer the support necessary for someone to contemplate and withstand the end of their primary relationship. Those unsatisfied in their relationships might discover greater happiness independently, and these emotions might be of a contagious nature. Thus, partnership dissolution might be disseminated across a social network through a process of social transmission that could include various social mechanisms, traveling from one couple to another.

Therefore, the fundamental question revolves around whether interacting with closely connected friends increases the inclination of discontented partners to end unsatisfactory relationships. In a broader context, the extent to which interpersonal ties impact separation is not well understood, and previous studies have not explored the potential for ripple effects in a social network of couples. If one couple's decision to separate impacts another's likelihood of initiating a breakup in their

own relationship, it raises the question of why these effects wouldn't propagate more extensively throughout society. Limited research has been conducted on how perspectives regarding separation could spread within social networks.

In this work, adjustments to a conventional infectious disease model are made in order to better depict the dissemination of separations in romantic partnerships in the context of a social network. These modifications take into consideration the following hypothesis: that separations can occur due to the personal characteristics and circumstances of the couple, but also influenced by the couple's own social circle. In other words, separations can occur either spontaneously or through social transmission.

A conceptual framework for these kinds of propagations is presented here, using agent-based models as the main tool in the research approach. To achieve this, some theoretical methodologies drawn from the field of epidemiology are used as the initial basis and then expanded to better fit the phenomenon of social contagion, using models and techniques from the field of machine learning to design better and more flexible agents.

2 Theoretical Background and Previous Research

2.1 Computational Social Science

Computational Social Science (CSS) can be defined as an interdisciplinary exploration of the social world across various scales, from individual agents to larger groups, using computational methods. The term many scales refers to the wide array of organizational, temporal, and spatial dimensions found in social groups, often simultaneously. Additionally, the notion of computation or computational approaches involves a wide spectrum of computer-based tools, as well as significant concepts and theories, ranging from algorithms for extracting information to computer simulation models. In essence, CSS presents an expansive realm of scientific investigation at the intersection of various social science disciplines, applied computer science, and other related fields.

CSS operates within an information processing paradigm of society, where the role of information is pivotal in understanding the functioning of social systems and dynamics. Specifically, the processing of information holds a foundational significance in describing and understanding the nature of social complexity. The information processing paradigm within CSS has both substantive and methodological dimensions.

From a substantive viewpoint, this paradigm uses information processing as the main tool to describe and explain the operations of society and human behavior, leading to the emergence of complex systems. Consequently, it underscores the indispensability of recognizing the role of human and social information processing as a fundamental phenomenon in comprehending social complexity.

From a methodological stance, the information processing paradigm highlights computing as a core instrumental approach for modeling and understanding social complexity. This does not negate the relevance of other approaches like historical, statistical, or mathematical methods. Rather, computational methods often rely on these established approaches, as well as other methodologies, to enhance the understanding and explanations of social complexity [29].

There exists a link between individual behavior and social phenomena, and the exploration of this relationship remains a central area of research within the realm of social sciences. Diverse methodologies for examining individual behavior have evolved within the behavioral sciences, contributing to the foundation of what is nowadays known as CSS. The recent advancement of Big Data methodologies, coupled with the widespread availability of data and the accessibility of cost-effective computing power, has significantly impacted the progression of this area of research.

Nevertheless, certain limitations still persist: the diversity inherent in individuals, the nonlinear dynamics characteristic of social systems and behavior, and the challenge of effectively bridging the gap between individual behavior and social structures and phenomena. Three primary research strategies are integral to the domain of CSS:

1. Data exploration is essential for describing and categorizing information. This fundamental approach is one of the cornerstones of scientific research, offering insights into unknown phenomena and systems.
2. Relationships within the data can be identified through the use of statistical models, thereby discovering interconnections and correlations among variables. Variables exhibiting these kinds

of associations might be promising candidates for more in-depth examination.

3. The study of causality requires not only data but also the use of modeling tools. The modeling process facilitates the creation, testing, and validation of knowledge. Within CSS, social simulation serves as the main methodology for conducting such modeling experiments.

Social phenomena show a distinct complexity compared to natural physical phenomena that originates from the intricate nature of the human brain, which stands as the epitome of complexity in social science. Furthermore, the intricacy of social phenomena is rooted in the fact that their underlying causes include both individual behavior and characteristics of the social structure, such as institutions, social norms, and the many ways of social engagement. Conventional causal explanations often prove inadequate and unreliable due to the limitations imposed by the analytical tools employed, which lead to an oversimplification of the intricate dynamics inherent to social systems [13].

2.2 Causality

Causality is one of the fundamental concepts within CSS and draws from the principles of complexity science, a discipline that studies complex systems: systems that exhibit distinct characteristics, mainly non-linearity and non-equilibrium. Despite this, they are not synonymous with chaotic systems, but they have the capacity to return outcomes that prove difficult to anticipate only based on an examination of the elements that compose them [44].

In the context of a complex system, two main tiers emerge: a micro level comprising individual components and a macro level of aggregated results. This can be illustrated through the example of temperature: a macroscopic property of materials not present at the microscopic level of atoms. The exploration of complexity initially took root within the realms of non-linear and non-equilibrium thermodynamics, subsequently extending its reach into the social sciences [85].

Complexity science is thus concerned with understanding the behavior of systems that have non-linear and non-equilibrium dynamics, rather than with static equilibria. To explore these systems, simulations are employed as the main tool for analysis, a method that enables the investigation of causality in complex systems.

One particularly intriguing type of complex systems is those that can self-organize, usually referred to as complex adaptive systems. These systems possess the ability to independently handle sudden changes in the environment and adapt to new situations [67]. Examples of such systems can be found in any kind of living organism and most social systems.

Another significant factor in CSS is the emphasis on social causal mechanisms as explanations for social occurrences. Social mechanisms can be defined as social processes that result in specific outcomes in certain parts of the social structure [80]. Given the hierarchical nature of social structures, these mechanisms operate between different levels, such as between individuals and groups, for example.

Social mechanisms can also be characterized through their explanatory power. A social mechanism should always play a pivotal role in clarifying a social phenomenon, contrasting with the concept of black boxes [60], and must have the capability to explain the causality behind a social event. However, their structure differs from that of scientific laws, since such laws tend to oversimplify and inadequately simplify the complex nature of causality [13].

Combining these viewpoints, another definition of a social mechanism can be a well-defined collection of factors responsible for a specific social phenomenon, where the requirement for mechanisms is closely linked to the nature of causality within such phenomena [15]. Social mechanisms do not promote a rigid and static view of social systems, putting the focus on attributes like self-organization, self-adaptation, and goal-oriented behavior instead: all of them qualities inherent to complex adaptive systems.

Social mechanisms contribute to CSS in two main aspects. The first is the importance of causality, which is crucial for understanding the set of factors giving rise to social phenomenons while preserving their complex nature at the same time. The second centers on delineating and bridging the gap between the micro and macro levels. Social mechanisms are often said to involve two categories of conceptual elements that facilitate understanding their systemic effects: actors and stages [90]. This clear conceptual differentiation between macro structural and micro individual levels allows the study of how the social structure impacts individual actions and, conversely, how individual actions shape the social structure [104].

The distinction between micro and macro levels depends on the specific phenomenon under examination and on the elements of the analysis. This division is generally characterized by the following attributes [13]:

1. It's a relative differentiation depending on the nature of the phenomenon being studied and the research process employed.
2. It considers a micro level composed of elements that engage in actions and interactions: individuals, groups, institutions or nations, for example.
3. It also considers a macro level, representing a complex system enveloping these micro components. The macro level is often conceptualized as a structure emerging from the collective social actions of the micro elements. The characteristics of this macro level must be observable and measurable for it to be useful.
4. The micro and macro aspects are separated mainly for analytical clarity. However, their relationship is also significant: social mechanisms aim to clarify how the macro level influences the micro level and vice versa, explaining the reciprocal dynamic between these two levels.

A variant of Coleman's boat can serve as an useful analogy for the micro-macro interplay [33]. This example aims to represent three distinct causal connections that define social phenomenons between levels. In the diagram, numerical values denote the causal relationships behind a social phenomenon, while the arrow's orientation indicates the causality's direction.

Relationships of causality in which the macro level impacts the micro level include situational mechanisms found in analytical sociology [63], second-order emergence common in social simulation [53], and the concept of "immergence" used in cognitive modeling and social simulation [95]. Various models that explore the formation of beliefs and preferences rely on mechanisms that connect the social structure (the macro level) with the beliefs, desires, opportunities and behavior of individuals (the micro level).

Another example can be found in the complex set of boundaries and opportunities encountered by individuals in social dynamics. Here, the surrounding environment, encompassing the social structure

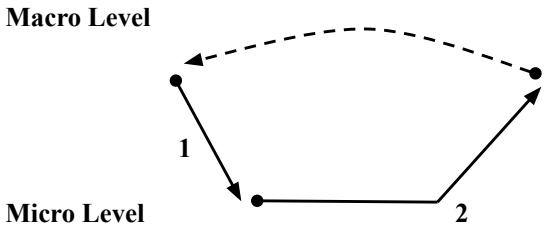

Figure 2.1: Coleman's boat diagram.

or natural surroundings, stands as a macro component defining the limitations and opportunities accessible to its components. Furthermore, a collection of social norms holds the capacity to shape the convictions and deeds of individuals, particularly those who exhibit limited self-awareness or possess a strong social identity [13].

The second form of causal relationships encompasses those at the micro level, as well as those that have an impact on the macro level through the interplay and aggregation of micro elements. This includes micro level dynamics that operate independently from the macro level, alongside those not dependent on social interaction, such as utilitarian decision-making. However, in reality, dynamics often intertwine: an individual's beliefs may be molded by both macro level factors and interactions with other individuals.

One widely recognized social mechanism is the collective behavior threshold theory. Threshold-based behavior refers to a situation where an individual's decision to adopt a specific behavior is influenced by the number of other individuals who have already engaged in that specific behavior. Heterogeneous thresholds in individuals lead to various threshold-based social mechanisms that can account for a wide range of observed empirical phenomena [57].

A significant portion of social mechanisms are based on interplay among the micro elements, such as imitation and communication, where social interactions shape how individuals shape or alter the larger macro level. The notions of social interaction within CSS frequently draw upon those formulated in social network analysis.

2.3 Social Networks

Social phenomena cannot be fully understood by only examining the individual factors; instead, the integral role that social interactions play needs to also be taken into account. These interactions, which are varied, widespread, interwoven, and dynamic, constitute an important aspect of any social analysis. They can be broadly classified into two main categories for analytical purposes.

Firstly, there exist indirect interactions, referring to interactions that take place through changes in the shared environment of the social agents. In these cases, individuals may not directly identify one another, even though they are well aware that shifts occurring in the environment are a result of the actions of other individuals.

Secondly, there are direct interactions, in which individuals come into personal contact, often through

the use of diverse means such as language, technology, or media. These direct interactions enable participants to identify one another, enabling the emergence of social dynamics like trust and reciprocity. Currently, direct interaction through social networks stands out as one of the most prevalent characteristics in contemporary social simulation.

Social network analysis (SNA) falls under the sociometrics field and aims to observe, measure, and statistically explore how individuals socially interact. Over recent times, it has regained attention due to the abundant data sets produced by modern social media platforms, as SNA can be used to examine social networks and faithfully replicate social interactions using diverse statistical metrics [13]. Within this context, network diffusion mechanisms encompass things like novel products, habits, values, and similar concepts. These mechanisms are affected both by the structure of the social network and the inclination of individuals to embrace innovations or changes if their nearby social contacts have already embraced them [32].

The understanding of human behavior benefits greatly from considering the impacts of social network effects. Human beings engage with other individuals, showing different levels of interaction and this social interplay significantly shapes behavioral patterns. Researchers from a wide array of disciplines, such as sociology, economics, physics, mathematics, and public health, have extensively studied social networks, in an exploration that encompasses not only the structure of these networks but also their significant influence on the dynamics of social interactions.

For example, various aspects of economic behavior, including decisions related to purchases [73], financial insolvency [100], employment opportunities [23], investments [31], and productivity [21], have demonstrated susceptibility to the influence of network connections. Additional behaviors, like academic performance measured by college GPA [92] and criminal activities [39], have also been associated with the effects of peers on the individual.

Recent focus has extended to the study of social networks as determinants of health [97]. This domain spans from describing the propagation patterns of infectious diseases [89] to the dissemination of behaviors like contemplation of suicide [10], obesity [27] and giving up smoking [28]. Moreover, networks also have a pivotal role to play in the realm of evolutionary biology, where population structures can serve to facilitate the emergence of cooperative behaviors [49].

Human emotions are clearly swayed due to interactions within an individual's social network. The phenomenon of emotional contagion [62] is well-documented, occurring among individuals sharing frequent and close bonds, such as roommates [68], family members [79] and team members [9], or even in controlled experimental settings [46].

The transmission of both positive and negative emotional states also extends to workplace dynamics, including areas like collaborative efforts [9], negotiations [105] and effective leadership [14]. Propagating a positive mood is a prevalent interpersonal technique in sales, exemplified by the familiar 'service with a smile' approach [86]. On the other hand, even the diffusion of depressive moods and symptoms, both by media exposure and interactions with unfamiliar individuals or acquaintances, has also been observed [72].

Associations in the levels of happiness among individuals linked within a social network have also been identified [45]. It was discovered that if your friends exhibit happiness, it significantly increases the likelihood of your own future happiness. This observed correlation is derived, at least in some part, from the transmission of happiness between individuals, implying that happiness can be analogously

considered as a type of social contagion.

Mathematical frameworks designed to illustrate the propagation of microbial infections serve as valuable tools for comprehending and anticipating disease dissemination [6]. These types of models have also proven surprisingly useful in investigating the spread of other kinds of contagions, such as rumors [36] or computer viruses [84]. Nowadays, the study of the dynamics of epidemiological models within social networks is an active field of research, where many models have been designed to explore the influence of network structures on the transmission of diseases [74].

Existing research on individual-to-individual transmission of couple separation has focused on aspects tied to the inter-generational transfer of divorce risk from parents to children. A commonly proposed idea is that parents who have undergone divorce are more inclined to raise offspring who are similarly predisposed to experiencing marital dissolution. This tendency is even more pronounced when both partners have parents who have undergone divorce themselves [22].

In particular, the daughters of divorced parents exhibit a heightened probability of experiencing divorce [40]. A comprehensive study revealed that within the initial five years of marriage, the likelihood of divorce escalated by 70 percent among daughters of divorced parents [22]. This susceptibility may transmit more substantially to daughters due to such women's stronger commitment to employment and career and intention to have fewer children, compared to the daughters of non-divorced parents. These factors reduce their reliance on men for economic sustenance, and may render divorce more financially viable for them [56].

2.4 Models of Contagion

Contagion, in its broadest sense, refers to the transmission of an entity or influence between individuals within a population, either through direct or indirect interactions. The existing mathematical models of contagion can be categorized into two main groups based on the inter-dependencies between successive contacts between individuals, which revolve around the impact of prior exposures on the chance of infection [38]. In all mathematical models of infectious disease spread (the classic SIR model) and some social contagion models [55], the assumption is that contacts are independent of each other. This means that the probability of infection is assumed to be consistent and unrelated to previous exposures. Such models fall into the category of independent interaction models.

On the contrary, threshold models propose that an individual becomes infected only after surpassing a critical number of exposures, at which point infection becomes highly likely. This presence of a threshold signifies strong inter-dependencies: contacts near an individual's threshold have a significant impact, while others have minimal or no effect. Threshold models are often utilized to depict social contagion scenarios (the spread of trends or rumors) where individuals decide whether to adopt a behavior based partly or entirely on the decisions of others, either deterministically or stochastically [38]. Another way to understand the relationship between successive events is using the concept of memory. Threshold models inherently imply the existence of memory, whereas independent interaction models assume that the infection process lacks memory.

2.4.1 Social Contagion

Social contagion is a phenomenon characterized by the diffusion of a specific behavior or attribute within a social group, subsequently extending its influence across a population. Analogous to the transmission of a disease, this macro level impact emerges from individual micro level social interactions. People engage in social networks constituted by friendships and familial connections, exerting mutual influence on each other that can reinforce behaviors deemed socially acceptable by each social circle. Named social mechanisms [64], these influential factors play a pivotal role in the phenomenon of social contagion.

Complex social networks provide an accurate representation of the connectivity within diverse systems and are commonly used as the fundamental social structure for dynamic processes like disease propagation, innovation diffusion, and opinion formation [1]. For example, when simulating the spread of an epidemic in a population, transmission between infected and healthy individuals is typically assumed to happen through direct interactions, and a single exposure to an infected person can cause infection. Such straightforward contagion processes are easily illustrated by transmission mechanisms along the network links connecting individuals.

However, when addressing social contagion phenomena, like the adoption of norms, behaviors, products, or the spread of rumors, the situation becomes more complex. Although simple epidemic-like contagion can aptly explain certain scenarios, such as easily persuasive rumors or ripple effects, these approaches fall short in capturing more complex dynamics involving peer influence and other social reinforcement mechanisms. To account for these dynamics, complex contagion mechanisms have been proposed.

Complex contagion (threshold models), necessitates individuals to be exposed to multiple sources of activation for transmission to occur [24]. In other words, a single contact with an active neighbor is insufficient to trigger adoption, so this type of contagion requires exposure to multiple active sources to induce the spread. Empirical evidence supporting the need for multiple exposures in describing social contagion has been observed in various experiments and contexts [103].

The modeling of social contagion processes has evolved based on these insights. Threshold models propose that individuals need to be influenced by a fraction of their social connections surpassing a threshold to adopt a new behavior, often being models of a deterministic nature [26]. Conversely, another modeling framework for social contagion involves extensions of epidemic-like processes, using stochastic contagion rates that may depend on an individual's number of exposure sources. Despite this complexity, all these models remain rooted in networks of interactions between individuals. Even when multiple interactions are necessary for contagion to occur, both threshold and epidemic-like models fundamentally rely on pairwise interactions, defined by the links in the network where the process unfolds [70].

On the other hand, social media has transformed the way information is produced and consumed. Unlike traditional media, which is passively consumed, social media relies on users actively sharing the information they receive with their social connections. This process is also another form of social contagion, one which can greatly enhance the spread of information within a social network. While the dissemination of information is often compared to the spread of a contagious disease, social contagion is distinct in that users on social media actively seek out information and make conscious decisions to propagate it [65].

From a theoretical standpoint, one of the most basic and extensively studied models of social contagion is the independent cascade model (ICM) [82]. In ICM-type models, each instance of exposure of an uninformed individual by an informed peer corresponds to an independent chance of transmitting the information package. Consequently, the probability of an uninformed individual becoming informed increases progressively with the number of exposures, which can potentially lead to a widespread epidemic involving a significant portion of the population.

However, studies focusing on information dissemination within social media have identified behaviors that differ significantly from ICM predictions. For example, when examining how individuals react to their friends recommending specific news articles, repeated exposure initially heightens the likelihood of transmission, but eventually exposure seems to impede transmission, contradicting key ICM assumptions. Multiple explanations have been suggested for this anomaly, including the concept of complex contagion, where the probability of adopting a behavior or idea varies with the level of exposure, implying that social phenomena can influence responses and interact with network structure [25].

Another explanation involves the linear threshold model, where the proportion of friends (beyond a certain threshold) adopting a behavior determines contagion [75]. Other factors influencing social contagion include the novelty or persistence of information and competition with other information [108].

2.5 Agent-Based Modeling

Agent-Based Models (ABMs) are tools for scientific modeling originally from the field of micro-simulation that consist of autonomous entities commonly referred to as agents. When applied to modeling social systems, these agents typically embody individuals, although they can also be used to represent entities such as couples, families, groups, institutions or organizations [54].

ABMs facilitate analyses across multiple levels and aren't constrained by the number of levels they can simulate: there's potential for simulating more than two levels. These models can capture various types of social structures, either static ones or dynamically generated by the agents and their interactions. This type of framework is notably open-ended, with the only requirement being the presence of interacting agents, engaging with the environment or each other [13].

These models revolve around two main elements: a collection of agents exhibiting autonomous behaviors (often heterogeneous in nature) and an interaction structure (which can be static, dynamic, independent, or influenced by agents' actions). The model's execution, known as simulation, yields macro-level outcomes that can be statistically analyzed. The distinction between micro and macro levels arises from ABMs' construction starting at the bottom, resulting in emergent macro results through the effects of interactions of micro elements.

ABMs can be understood as a structured tool for simulating social systems, distinct from argumentation for its use of formal modeling and also different from equation-based formalization common in social sciences [53]. ABMs have also been described as opaque mental experiments where outcomes stem from premises in non-obvious ways, showing the need for conducting systematic exploration of these models. Analyzing, verifying, and validating ABMs can quickly become a difficult task due to the inherent complexity encompassed within the model; a complexity that often yields many different

potential outcomes, complicating the understanding of the model's internal dynamics.

While the researcher's micro level specifications shape the macro results produced by the model, connecting outcomes with specific model components isn't straightforward, as instances of interconnected social mechanisms and overlapping dynamics can arise. Systemic sensitivity analysis needs to be done in order to illuminate the relationship between aggregated macro outcomes and the characteristics of micro level elements. By scrutinizing the model, the system it mimics, and through trial-and-error approaches, a deeper understanding of the model's mechanisms and the nature of their interconnections can be obtained [13].

2.5.1 Tools

Tools designed for agent-based modeling benefit from the adoption of object-oriented programming (OOP) principles. This approach establishes a correlation between software objects and model agents, simplifying the model programming process and making it resemble mental or paper-based definitions. Typically provided in the form of software libraries, these tools encompass several categories of functionalities: model setup, time management within the model, internal state inspection, and dynamic visualization of results.

Model setup is facilitated through pre-established components such as agents, interaction structures, and scheduling systems for actions and events. Effective management of time and actions is crucial in ABMs, where time is inherently discrete and model dynamics are driven by events. The capacity to dynamically access an agent's internal state also proves useful for debugging models or understanding model behavior.

2.5.2 Critical Issues

Certain elements within ABMs can have probabilistic characteristics that collectively lead to non-deterministic outcomes. Empirical data usually guides these probabilistic aspects of the model by defining the probability distributions of these random variables. Given that computer programs are inherently deterministic, this probabilistic behavior is recreated through the use of algorithms known as pseudo-random number generators. To ensure consistent model outcomes, a single instance of such a generator with a fixed random seed is typically used, guaranteeing the generation of the same sequence of random numbers during each model run. However, relying solely on one seed can erroneously imply determinism, being necessary to use multiple distinct seeds.

Determining the needed number of random seeds in advance is challenging, as probabilistic elements can impact the frequency of other non-deterministic components. In other words, the propagation of uncertainty before initiating the simulation cannot be predicted. The solution lies in the use of a large enough number of random seeds in multiple simulation runs, so that model outcomes can become statistically significant after they are aggregated. Thus, in order to analyze ABMs and pinpoint which probabilistic components contribute the most to outcome uncertainty, sensitivity analysis is a commonly used tool.

Before presenting simulation results and validating them against empirical data, a comprehensive model description should always be provided. This entails detailing the number of agents considered,

defining the individual features of agents, and explaining the attributes of both agents' behavior and the interaction structure. Although sharing the model's complete computer code implementation in a clear manner offers a comprehensive solution for verification and replication purposes, practical challenges arise in reality. ABMs can be implemented in multiple and diverse programming languages and tool-kits, making it unreasonable to assume that all readers possess knowledge of every technical aspect possible.

One potential solution involves adopting a straightforward, standardized, and easily comprehensible metalanguage; however, a universally accepted language of this nature is currently absent within the agent-based modeling community. Presently, the most prevalent solution revolves around using protocols that describe the model through mathematical formulations and diagrams that explain the algorithms and model structure, such as the Unified Modeling Language (UML) which covers agents' attributes, behavior, and interaction structure. Another widely accepted and more detailed protocol is the "Overview, Design Concepts, and Details" (ODD) protocol [58].

2.6 Behavioral Agents

Behavior in agents can be modeled using two distinct methodological approaches. The first approach solely focuses on behavior and omits the task of explaining behavior within the model. The second approach employs more advanced agent models that encompass cognitive functions and structures, aiming to simultaneously explain both behavior and social phenomena.

Behavioral agents are not black boxes, so the causal connections that lead to specific decisions must be transparent and grounded in empirical evidence. Essentially, the behavior of agents can be perceived as a function that maps the available information in the social environment to specific possible actions [13].

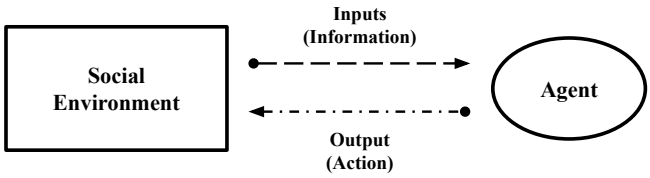

Figure 2.2: Agent interaction with the social environment.

In this approach, the model does not consider the underlying reasons that drive agent behavior. If necessary, these aspects can be explored using more intricate agents and alternative tools. Given that behavior constitutes one of the model's central elements for explanation, empirical validation becomes crucial to establish credible explanations of social phenomena and behavioral agents should be characterized by their strong empirical relevance.

In terms of modeling behavioral agents, two main approaches are often used: developing a model of agent behavior requiring calibration using data (at the individual or population level) or categorizing observations based on their alignment with theoretical models. When calibration is required, regression analysis stands as the primary tool. The chosen regression model and estimation technique depend on the units of measurement of observed variables.

The nature of data collection shapes the observed variables and their measurement scales, but even in cases where data collection has already occurred, data types can often be adjusted. For example, a continuous cardinal variable in reality can become discretized if it was collected through a survey with limited precision, losing some information in the process. Additionally, measurements taken on a bounded scale may suffer from data exclusion or transformation.

Behavioral agents are constructed with causal connections that dictate the output in the form of an action based on the input they receive from the environment. Considering the structure of causal relationships, not all behavioral models are identical. Some models have a single decision variable, where agents pick just one action, while others involve multiple decision variables to be simultaneously determined. These models can range from straightforward additive functions to much more complex decision trees. Furthermore, there are models and calibration tools suited for scenarios where a single behavioral model can represent an entire population, and others that are more applicable when individual calibration is required for each agent.

2.7 Extending Agents with Machine Learning

Machine learning (ML) as a field includes a wide range of techniques where algorithms employ statistical methods to autonomously learn from data, without the need for explicit programming. With ML, underlying relationships, patterns, and associations within a dataset are extracted without requiring a previously predetermined theory or model. However, it's important to acknowledge that some general model structure must be assumed, such as accepting that data sets can be represented by linear functions, neural networks, inherent clusters of associations and similar. The way these necessary assumptions are realized is often referred to as the inductive bias of a machine learning model.

2.7.1 Machine Learning Concepts

At the beginning, every ML project begins with a question, data, and a model. With a model in hand, the data itself, and not not the programmers, defines what to do next in the project [3]. The process of how this learning is achieved becomes clearer when examining ML as a means to determine an optimal function that maps a dataset (a collection of independent variables, often named features) to outcomes (dependent variables). In essence, the goal is to identify a function $\mathcal{F}$ that connects the input data to the results, denoted as $\{d\} \longrightarrow \{r\}$. This functional mapping can range from a simple linear model (a weighted sum of linear terms) to a complex deep neural network (DNN).

In the case of artificial neural networks (ANNs), the many hyper-parameters (weights, activation thresholds, biases...) that define a DNN model are learned through backpropagation algorithms, such as Stochastic Gradient Descent, and it's this learned arrangement of connections and weights that creates the functional mapping. Generally, the aim is to discover a mapping that is optimal by minimizing the difference between predicted outcomes and true outcomes that are already known (considered ground truth) for a specific subset of data, usually called the training set.

The learning aspect of machine learning involves iteratively approximating parameters to optimize a model according to an objective function. This training process consists of finding a set of weights/parameters that enhance the accuracy of the input-output mapping, which is equivalent

to minimizing the model's prediction error. The weights can be as straightforward as numerical coefficients of variable terms in the optimization function, with complexity arising from the great number of such parameters.

As an example, in linear regression, minimizing the mean-squared error is equivalent to maximizing the likelihood of the observed normally-distributed data under the optimized linear model. Once this optimization is achieved, we have a trained model, and the subsequent phase of an ML project is typically the inference stage, where the trained model is used to make predictions. This next phase involves using the trained model on new and unfamiliar data, which could be in the shape of test data sets while still refining and validating the model, or real-world data, if the model is intended for operational use.

The precise distinction that identifies an algorithm as a ML algorithm, as opposed to other approaches like statistical modeling, is still subject to debate. In this context, a broad characterization of machine learning algorithms based on Mitchell's criterion can be adopted: an algorithm is considered to be learning if it enhances its performance, denoted as $\mathcal{P}$, in relation to a specific task, denoted as $\mathcal{T}$, through the execution of computational processes, denoted as $\mathcal{E}$ [81].

Machine learning algorithms can generally be grouped into four main types, primarily determined by the role of labeled data in the training and learning stages (although the boundaries between these types can sometimes be blurry): supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

- Supervised learning algorithms establish connections between independent and dependent variables by applying a model trained on previously known data characteristics [76]. This dataset, often referred to as ground-truth data, comprises accurately labeled samples that can be divided into various subsets (such as training, test, and validation sets) as part of the model training process.
- In unsupervised learning, an ML algorithm independently identifies patterns, trends, or groupings (clusters) within a dataset, without relying on prior human knowledge about the correct associations. This approach only employs unlabeled data that lacks annotations from human experts [52].
- Semi-supervised learning, which is useful when a significant amount of data is available but only a small portion is properly labeled, involves training sets containing both labeled and unlabeled data, with the latter being predominant. Informed ML, or expert knowledge-driven ML, is regarded as a form of semi-supervised learning, where the modeler can manually adjust model weights based on known system behaviors and properties [107].
- In reinforcement learning, the ML process operates based on a set of rules (policies) and system states (similar to ABM), receiving rewards or penalties according to the outcome of an action. The ML algorithm evolves the system through state transitions to maximize rewards within a given environment, thus learning [102].

2.7.2 Model Assessment

Training, selecting, and assessing models are crucial stages in a ML development pipeline, and this significance holds true when thinking about potential integration with an ABM framework. Even when approaching these topics from a broader standpoint, subjects like model selection and evaluation of the performance of ML models form an entirely different field of study, with only a few key points being explored in this work.

Firstly, the process of model training (the core learning aspect of a ML algorithm) and model selection are not usually isolated matters. The evaluation strategy used for a ML approach depends on the structure and complexity of the model in use. What might be a suitable evaluation approach for regression-based or other shallow learning methods, both in terms of subset ratios and how data is allocated to training, validation, and test sets, might not be as fitting for more complex models, such as deep neural networks (DNNs).

Secondly, the process of model development and its performance is tied to the core objective in ML of balancing overfitting and underfitting. Striking the optimal balance in this regard yields the most effective model, characterized by the lowest generalization error, which improves the accuracy of predictions with unseen data or data points that are distant from those in the training set.

Lastly, both the general approaches (cross-validation, Bayesian model selection...) and the types of evaluation metrics (log-loss function for classification tasks, mean squared error for regression...) can significantly vary based on the specific type of ML project being conducted (ANN versus non-ANN). In summary, when contemplating the integration of ML approaches with an ABM framework, meticulous attention is required to ensure that the different approaches used in training, testing, validation, and other facets of evaluating ML models are as correct as possible. As with any applications of ML, caution should be exercised when adapting validation and evaluation strategies to the specific type of model in use.

2.7.3 ML / ABM Integration

Considering their individual strengths and limitations, the methodologies of ABM and ML can be seen as complementary strategies for modeling social systems, especially when each approach's strengths can address the weaknesses of the other one. For instance, ML algorithms like DNNs are often criticized for generating predictions in a black box manner. Moreover, many supervised algorithms require abundant accurately labeled training data and can be prone to overfitting.

On the other hand, ABMs have explicit, well-defined representations of interactions between system components, making them relatively straightforward to formulate. They can also generate substantial output data through simulations. However, creating rules in ABMs is frequently done manually by the researcher, potentially resulting in biased or simplified abstractions of social complexity. Here, ML algorithms can computationally deduce optimal agent rules, enhancing the accuracy of interactions and discovering new potential rules previously omitted by researchers. Some studies have merged ML and ABMs in the following ways to create more advanced and precise computational models of complex systems at both the micro and macro levels [18]:

- Learning ABM Rules via ML: reinforcement and supervised learning can infer and refine agent

rules, which are pivotal in defining the ABM's agents' behavior at each discrete time step.

- Parameter Calibration: stochastic optimization techniques like genetic algorithms and particle swarm methods can be used to calibrate ABM parameters.
- Exploring ABMs via ML: ML methods can help navigate the intricate, high-dimensional parameter space of ABMs, performing sensitivity analysis and evaluating model robustness.

Despite the considerable achievements of ML in various fields, there can be notable limitations when applying ML to computational social science (CSS). Constructing a precise and resilient ML model needs extensive experimental data that can be challenging to acquire due to factors like measurement inaccuracies, variable discretization in surveys, sparse data sets or privacy concerns.

One significant challenge for modelers is that ML architectures and algorithms often result in black box solutions. Unfortunately, these solutions can obscure some of the internal mechanisms connecting predictor and outcome variables, thereby failing to reveal the causal mechanisms underlying system-wide behavior. This gap has prompted the emergence of the research field known as explainable artificial intelligence [106].

2.7.4 Agent Rules Creation

The rules of an ABM define the actions an agent can take based on its state and on reactions to changes in its environment. For example, a healthy individual could contract a virus when in close proximity to an infected individual. It's important to note that the terms 'can' and 'may' are used because ABM rules are usually probabilistic in nature. Thus, these rules establish causal relationships that agents (such as individuals, couples or organizations) in the studied population follow.

Traditionally, these rules are manually crafted by the model creator, who combines empirical data about the system with expert insights or existing literature. The rule-set in an ABM is verified only after running simulations and comparing aggregate model outcomes to independent experimental data or validation data sets. Consequently, a common critique of the traditional ABM rule-generation process is that the modeler's inherent subjectivity might introduce bias into the rules, potentially influencing the significance of subsequent results and forecasts. Moreover, crafting rules for complex social systems requires a substantial amount of expert sociological knowledge about the social phenomenon being modeled, one that might not be accessible to a vast amount of researchers, including those coming from other fields. This can be a difficult barrier in the process of creating ABMs and defining agent rules, one that might limit innovation and the variety of research being done in this area.

To address this potential concern, recent ABMs have started to utilize ML to determine, in a more systematic and data-driven manner, the rules governing agent behaviors. Instead of manually creating rules, which might unintentionally lean towards certain predictions that don't accurately represent the target population or overall system behavior, ML algorithms can learn these rules in a more objective way, in a learning process that involves analyzing experimental data to extract significant relationships between variables. This capacity to learn from data comes from ML's origins in information theory and statistical learning [61].

In epidemiological ABMs, supervised learning algorithms have been employed to define agent behaviors

for simulating the spread of both infectious and non-communicable diseases. The utility of supervised learning in ABM rule generation stems from its ability to learn agent rules from labeled data sets that establish connections between agent attributes and behaviors under different conditions.

For example, in a micro-simulation focused on diabetic retinopathy (DR) within a group of individuals, a multivariate logistic regression algorithm was used to create rules governing the progression of each human agent to the next stage of DR [37]. These rules were developed based on features like age, gender, diabetes duration, tobacco use, and hypertension. Instead of manually deducing probabilities of DR stage advancement from literature, these rules were learned computationally by training a multivariate logistic regression model on a dataset encompassing 535 DR patients. The logistic regression algorithm derived a function mapping individual patient attributes to the likelihood of DR stage advancement that was applied at the start of each simulated year to determine whether human agents would transition to the next DR stage. This approach demonstrated that a simulated cohort of 501 patients closely matched a real patient cohort and even aided in identifying key predictors of DR stage advancement. This example highlights the general principle that regression models can be valuable for constructing agent rules when enough population data is available.

Another study evaluated various supervised learning methods within an ABM framework to predict individual responses of DR patients to pregabalin. The research found that combining ensemble methods that incorporate several instance-based learning approaches, including supervised k-nearest neighbors and fuzzy c-means, resulted in the highest classification accuracy [2]. Real-world data concerning the transmission of COVID-19 in hospitals and other settings has also been harnessed to construct and implement ABMs aimed at optimizing policy measures and investigating epidemiological questions [35].

Ideally, an ABM's rule-set encapsulates the fundamental mechanisms guiding the behaviors of individual entities within their local environments. Hence, when ML is employed to extend agent rules, a key implication is that the ML model's structure and quantitative formulation must faithfully mirror the decision-making processes of such agents. Additionally, the model should always exhibit a level of generalizability, within reasonable limits.

The use of different ML approaches can influence the accuracy and generalization ability of ABM predictions. Such a comparative analysis has been conducted in an epidemiological ABM of cholera spread, comparing a decision-tree-based algorithm with a naive Bayes approach for simulating the decision-making capabilities of agents in the ABM, and the ABM's emergent predictions differed based on the chosen ML approach [8]. It's important to acknowledge that no universal approach suits all possible scenarios, and thus it's prudent to systematically evaluate various ML/ABM integration methods.

The selection of a ML algorithm is likely to be influenced by two key factors: the nature and availability of data, and the ability to validate the ML algorithm both individually and when integrated into an ABM framework. In scenarios related to epidemiology, substantial training datasets can be generated from survey data, enabling effective training and validation of supervised learning algorithms in isolation to define agent behaviors. This environment allows for the training of multiple supervised learning algorithms, and the one with the highest predictive accuracy can subsequently be incorporated into an ABM simulation. As said before, it's recommended to explore a broad range of ML/ABM integrations whenever possible, and to provide justifications for the chosen learning algorithm employed for rule generation.

2.7.5 Model Calibration

ABMs commonly incorporate a range of parameters that determine how agents behave and influence model outcomes. While certain parameters can be directly set through experimentation or are well-defined in literature, such as the contagious period of an infected individual, many parameters usually remain impractical to measure experimentally. Parameters like the extent over which a virus spreads between individuals pose challenges due to their inherent difficulty in accurate experimental measurement. Furthermore, parameters with significant variability may inherently possess wide distributions that are difficult to capture accurately. This variation can be very extensive, spanning greater than an order of magnitude in some cases.

A significant obstacle in ABM development is determining these parameter values in a way that statistically aligns model outcomes with empirical measurements. Typically, this process involves creating an appropriate error or fitness function that compares model outputs to experimental results. An optimization algorithm then fine-tunes multiple parameters to minimize error or maximize fitness. For example, in an ABM of infectious disease spread, an error function might calculate the squared difference between the final fraction of infected individuals in the model and in real-world data. The parameter calibration algorithm then seeks an optimal parameter combination that minimizes this error function, but this process can be computationally expensive. Exhaustive systematic sweeps of parameters become infeasible due to the explosive growth in search space size with increased dimension, a phenomenon known as the curse of dimensionality.

Given ABMs' highly multidimensional parameter spaces, efficient calibration pipelines are essential to explore the space and minimize the number of parameter combinations requiring evaluation. Genetic algorithms (GAs) and other biologically-inspired evolutionary methods provide effective stochastic optimization strategies for searches with high dimensions. This approach has been acknowledged in ABM contexts, where genetic algorithms have demonstrated their potential in optimizing parameters [99].

GAs are a widely embraced ML technique used for parameter calibration and, more broadly, for tackling numerical problems that aim to identify global optimal solutions within extensive multi-dimensional search spaces. Drawing inspiration from the biological processes of natural selection and molecular evolution [66], GAs excel in locating combinations of parameters (referred to as solutions or individuals in a virtual population) that probabilistically optimize a defined fitness function. In a general sense, the operation of a GA involves four distinct stages:

1. Initialization: A population of individuals is initialized with randomized parameter combinations, encoded as genotypes (each genotype corresponding to an individual).
2. Evaluation: The fitness of each individual within the population is numerically assessed.
3. Selection: Individuals with relatively high fitness are selected as parents through specific criteria or protocols, promoting higher overall fitness in the population. Selection protocols can include stochastic elements, such as tournament selection or roulette wheel selection [109].
4. Genetic Operators: Well-defined genetic operators, including crossover (recombination) and mutation, are stochastically applied to a subset of the population to produce the subsequent generation of offspring.

This iterative process continues, with each subsequent set of individuals becoming a new generation, and the steps from the evaluation stage onward being repeated. The GA cycles can stop based on a predetermined number of iterations/generations or when a convergence threshold is met. At the end of this process, the available individuals (each represented by an encoded genotype) depict various solutions to the original problem. The optimal solution is extracted from the final set of genotypes, corresponding to the fittest individuals and their optimal phenotypes.

As the iterations of evaluation → selection → reproduction/mutation progress, exploring new regions of the search space with each repetition, the average fitness of the generations progressively approaches more optimal values. Once the GA reaches this stage and identifies a parameter combination that optimizes the fitness function sufficiently enough, it can be considered to have successfully converged.

While GAs offer the advantage of optimizing multi-objective fitness functions to avoid running an ABM for every conceivable parameter combination, they still come with a substantial computational cost. The inherent stochasticity of an ABM necessitates running it multiple times to reach stable values for a single parameter combination (genotype) within a given generation of the evolving GA. As the complexity and computational demands of the ABM increase, traditional GAs become less practically feasible, particularly for calibrating intricate ABMs.

An ongoing research area seeks to devise strategies that enhance the numerical efficiency and computational robustness of GA/GA-like methods, especially in challenging fitness landscapes. Covariance matrix adaptation-evolution strategy (CMA-ES) algorithms, including probabilistic model-building GAs (PMBGAs), are being explored as potential solutions for more efficient optimization [96].

To increase GA's computational efficiency, another approach involves reducing the number of parameters being optimized, which in turn reduces the dimensionality of the search space and the steps needed for convergence. In this context, ML techniques can be used to perform sensitivity analyses on an ABM and pinpoint the most crucial parameters for calibration.

Random forests (RFs), an ensemble of decision trees, are a well-known supervised learning method for sensitivity analysis [34]. For example, RFs have been used to identify sensitive parameters in a multi-cellular ABM focused on vocal fold surgical injury and repair [51]. This approach involved initially running the ABM with various input parameter values to build a training dataset, linking inputs to model outputs. The RF was then trained on this data to classify model outputs based on input parameter values and ranked these input parameters hierarchically using the Gini index, a variance-based measure that returns the likelihood of output misclassification if a certain parameter is chosen at random. This study then selected the top three parameters associated with each cell type for calibration with a GA. This type of combined approach reduced the number of parameters needed for GA calibration, enhancing the computational efficiency of the calibration process thanks to the dimensionality reduction being applied.

3 Pairfam Study

The data used in this work comes from the German "Panel Analysis of Intimate Relationships and Family Dynamics" (Pairfam) study , which was designed to provide empirical insights to improve the understanding of couples and family dynamics. The Pairfam study operates under the premise that progress in family research relies on the integration of expertise from diverse disciplines and various fields, the development of comprehensive theoretical perspectives and the use of longitudinal approaches supported by a substantial, representative database and a wide range of data [69].

Extensive longitudinal studies focusing on aspects like union formation, fertility, and inter-generational relationships have primarily emphasized economic and socio-structural factors, often overlooking the psychological aspects that underlie detailed family-related decision-making processes. While psychological research yields extensive data, the samples are frequently small or fail to sufficiently incorporate the wide variety of socio-structural variables.

Using a design that spans from adolescence to late adulthood and encompasses multiple generations, the Pairfam study enables investigations into developmental aspects, varying trajectories in relationship development, and the reciprocal influences among family members. This design also allows the exploration of various contextual conditions within both immediate and distant environments. The study began in 2008, involving a sample of 12,400 participants spanning adolescence, young adulthood, and middle adulthood.

Longitudinal analyses are crucial not only for uncovering common pathways of partnership development but also for addressing questions of causality. In partnership research, additional complexity emerges from the concept that partners' life courses are intertwined [12]. The attitudes and actions of each partner shape the context for the other's decisions, and this interplay must be considered (linked lives). To explore how partners influence one another, such as in educational, occupational, or family planning choices, joint life courses need to be examined using suitable dyadic data and corresponding statistical models. Currently, there's an emerging focus on how partners reciprocally influence each other, although this area is still gaining traction in research.

Psychological analyses highlight several key factors influencing partnership quality and stability, including past relationship experiences, individual coping skills for emotionally charged social interactions, attributions partners make about each other's behavior, conflict resolution strategies, and dyadic stress management. More recently, the emphasis has shifted from solely addressing destructive elements in relationships to examining repair mechanisms that help couples overcome crises and build resilience in their relationships. These mechanisms include elements such as religious beliefs and the willingness to forgive and make compromises. While it's pertinent to connect these micro level perspectives with the macro level trends mentioned earlier, interdisciplinary approaches that bridge these areas are developing slowly. However, such approaches hold significant promise in facilitating comprehensive yet nuanced analyses of partnership development concerning relationship quality, division of labor, power dynamics, and relationship stability [7].

One underlying assumption is that the challenges of navigating between different life domains increase the cost of being in an intimate relationship or marriage in modern welfare societies, as balancing commitment within a relationship with personal flexibility, autonomy, and career mobility can be complex. While intimate relationships might contribute to individual well-being, people seem less inclined to heavily invest in such partnerships, instead tending to minimize the constraints arising

from the commitments of intimate relationships. For many individuals up to mid-adulthood, living alone has become more appealing , potentially because societal norms have shifted and the practical benefits of cohabitation have diminished [69].

Although these competing goals and options might contribute to the decline of marriage and the instability of relationships, they don't influence all societal groups or individuals in the same manner. For instance, regional variations in marriage behavior suggest that cultural factors continue to influence the diversity in how unions are formed and evolve [59]. The landscape of marriage and partner markets has undergone social restructuring and now includes new avenues for meeting and forming relationships. Living conditions and expectations regarding fulfilling relationships have also changed [4], resulting in discrepancies between reality and expectations. For instance, the widespread acceptance of gender equality clashes with the enduringly stable gender roles in the division of household tasks, as observed in the daily lives of parents in Germany [69].

The Pairfam project aims to facilitate such analyses by taking a comprehensive approach. It examines various aspects of partnership quality (intimacy, autonomy, emotional security, etc.), subjective quality, dyadic coping, and conflict resolution behaviors experienced and enacted by both partners. The project also delves into personal expectations, personality traits, skills, social networks and the quality of other significant relationships, providing a holistic view of relationship dynamics.

3.1 Variables and Scales

The data collected in the Pairfam study is stored in a survey-like manner, using ordinal variables ($1-5$, $1-7$, $1-10\ldots$). Because some of the questions are directly adapted from other different questionnaires, the range of the possible responses varies in multiple ways and the measurement scales tend to change depending on the questions. The answers to all the questions are stored in two different SPSS files per wave: one for anchor respondents and another for their partners (if they have any). Pandas has been used to load these files into Python in the form of a DataFrame, for ease of exploration and data manipulation.

First of all, all variables have been scaled from their original scales to the $[0, 1]$ range. Once the anchor and partner data is merged, the created variables have a prefix denoting the gender of the member of the couple (`m` for male and `f` for female, as the study mainly centers around heterosexual couples) and a suffix if the question is asking about their opinion about their partner: for example `m_verb_aggr_p` is a variable that quantifies the verbal aggression towards the man in the couple coming from his partner, but and only from his point of view. The different kinds of variables used or created throughout this work are listed and described in the following subsections

3.1.1 Generated Variables

These are the variables automatically generated by the Pairfam study, with ease of use and convenience in mind. For example, there are unique identifiers for anchors and partners (`id` and `pid`), the sex of both members of the partnership (`sex` and `psex`), their age (`age` and `page`) and their level of education attained (`yeduc` and `pyeduc`). Factual characteristics of the relationship also fall into this category, like the marital status of both members (`marstat` and `pmarstat`), the duration of the relationship, cohabitation and marriage (`reldur`, `cohabdur` and `marrdur`), the number of previous

partners and marriages (`np` and `nmar`) or whether the couple is expecting a child (`pregnant`).

3.1.2 Relationship Value, Expectations and Satisfaction

The `tradmarr` variable measures how much of a traditional marriage idea someone has: things like considering marriage a lifelong union that must not be broken and the necessity of getting married to someone after things like cohabitation or having children occur. On the other hand, the value of partnership scale (codified as the variable `value_rel`) registers the value of the relationship for the respondents from an expectations point of view. It asks questions about perceived costs and benefits of a relationship in five different dimensions: stimulation, affection, status, comfort and autonomy. These dimensions have been defined according to self-determination theory [91].

In a very similar way, the quality of the relationship is measured in four scales: intimacy, admiration, dominance and conflict. These scales have been adapted from the Network of Relationships Inventory (NRI) [50]. The two positive scales, intimacy and admiration, measure honesty towards the partner and how often the partner shows appreciation towards the respondent, respectively. The negative scales, dominance and conflict, register the tendency of the respondent's partner to get their way when a consensus can't be reached and the frequency of quarrels and disagreements.

Lastly, the satisfaction of both members of the marriage with respect to different relationship domains is also measured in the following variables: `sat_rel` registers the overall satisfaction with the relationship, `sat_life` the general satisfaction with life and `sat_sex` the satisfaction with their sex life.

3.1.3 Personality Traits

The explosiveness and tendency to anger scale is stored in the `explo` variable, and it measures how quickly the respondent becomes angry, as its name suggests. Emotional autonomy (`emo_aut`) gathers information on the extent to which the respondent is willing to stand up for their own opinion and defend it against other opposing views. Shyness is also measured in the `shy` variable. Furthermore, personality traits belonging to the Big Five Inventory [87] are registered as well: extraversion (`extrav`), neuroticism (`neuro`), agreeableness (`agree`), conscientiousness (`consc`) and openness (`open`). The level of religiousness (`relig`) was also registered.

3.1.4 Emotional Insecurity, Autonomy and Competence

Emotional insecurity is another dimension of relationship quality explored in the Pairfam study, using three subscales to this end: ambivalence (`ambiv`), fear of love withdrawal (`love_with`) and engulfment anxiety (`engul_anx`). Ambivalence asks about the extent to which the respondent's need for commitment from the partner is connected with the fear that this need will not be met. Likewise, fear of love withdrawal registers the respondent's fear that their own bad behavior or failure might lead to rejection or loss of love from their partner. Lastly, engulfment anxiety collects information on the emotional defensive reactions of the respondent when their partner attempts to impose their will.

The autonomy scale (`auto`) measures the extent to which the respondents feel that they can pursue

their interests without restriction within their relationship. Another scale to measure relationship quality is competence in the partnership (`comp`), which registers information on the subjective feelings of competence that the respondent has in their own partnership, something that also serves as an indicator of the perceived quality of the partnership.

3.1.5 Instability, Conflict and Cognitive Disposition

The instability of partnership scales (`ins_rel`) collects information about the subjectively perceived instability of the relationship, putting the focus specifically on intentions of breaking up or divorcing. Serious problems related to use of drugs (`prob_drugs`), extra-marital affairs (`prob_cheat`) and the use of physical force (`prob_force`) are also registered in their own variables. Different scales on the conflict styles of both members of the marriage are also considered. Verbal aggression covers verbal, non-physical forms of aggression during conflicts between partners (`verb_aggr`). Manipulation (`manipu`) collects information on indirect and dysfunctional strategies for attempting to modify the partner's behavior during a conflict. A further instrument for measuring conflict styles is the withdrawal (`withdr`) scale, which consists of measuring behaviors like staying silent and refusing to talk about the subject. In contrast, constructive behavior (`constr`) measures the willingness to compromise in partnership conflicts.

Lastly, scales designed to capture cognitive-emotional behavioral dispositions like hostile attributions, commitment and reciprocity orientation are also used in this work. Hostile attributions (`host_attr`) gathers information on negative attribution tendencies. Commitment is measured on two different scales: future orientation (`fut_orient`) registers the expectation of a long-term future as a couple and tolerance of conflicts (`tol_confl`) addresses the tendency of respondents to consider separation when conflicts appear in the relationship. Lastly, the scale reciprocity orientation (`recipr`) collects information on the extent to which the respondent expects immediate reciprocity when they do something for their partner.

3.2 Correlations

The most important correlations between variables can be seen on the correlation matrix in Figure. The highest positive values in correlation correspond to the variables `sat_rel` and `sat_rel_p`: when a member of the couple is satisfied with the relationship, it tends to think that the other member is similarly satisfied. Curiously, the correlation values for the real satisfaction values of the partner and the estimations made by the other member are not that high, meaning that these estimations might have a tendency to underestimate or overestimate the real satisfaction of the partner.

Other significant correlations concern the use of verbal aggression and manipulation between partners in conflict, so when a member of the partnership admits to having used insults or manipulation techniques it usually means that their partner did too. The use of insults and manipulation is also positively correlated, so there exists some evidence that they tend to go hand in hand. The same thing happens in instances of use of physical force in a partnership conflict. Verbal aggression is also related to the conflict scale, signifying that the use of insults negatively impacts the relationship's quality, as it may be expected. However, the use of constructed behavior to solve conflicts is also highly correlated between both members of the marriage.

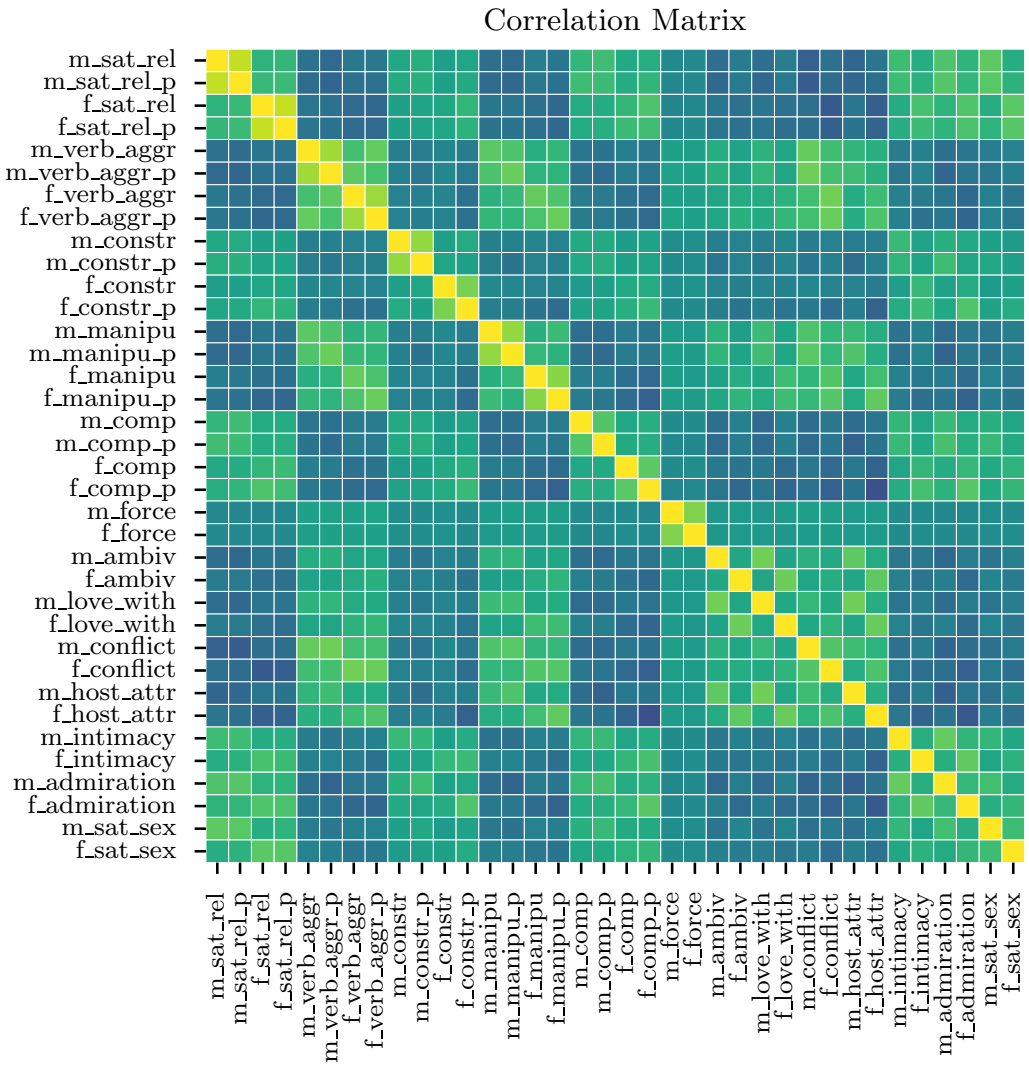

Figure 3.1: Correlation matrix of the most correlated variables.

Scales of emotional insecurity like ambivalence and love withdrawal are also positively correlated between them, and there is also a positive relation between these and cognitive-behavioral hostile attributions. On the other hand, positive scales like intimacy and admiration can also be seen to be positively related, and the overall quality of the relationship is correlated with the satisfaction with the partnership that each member has and the perceived satisfaction with the relationship of the other corresponding member of the marriage.

4 Better Agents with Machine Learning Models

4.1 Divorce Prediction and Relationship Problems

In the task of trying to predict if a couple is getting a divorce, the response variable (getting a divorce or not) can be considered a binary qualitative or categorical variable, so a classification problem has to be solved in order to get a good prediction. Classification can be defined as the process of predicting a qualitative response. Predicting a certain qualitative response for an observation is known as classifying that observation, because the observation can be assigned to a specific class or category. To this end, some methods predict the probability of belonging to each class (similar to regression models) and then use this probability to make the classification, while other methods directly assign a label to the observation without computing probabilities first.

There are many classification methods, also called classifiers, used to predict a categorical response, with some of the most well-known classifiers being logistic regression, linear discriminant analysis, quadratic discriminant analysis, naive Bayes and K-nearest neighbors. The main objective is for the classifier to not only perform well on the training data, but also on test observations that were not used to train it. In other words, the chosen method should be able to generalize well to new, previously unseen data.

In a binary classification problem, a dummy variable approach can be used to apply regression methods for classification purposes with the use of a threshold value, usually 0.5. The coefficients of linear regression in this case correspond to the conditional probability $Pr(Divorce|X)$. However, by using linear regression some of the predictions might be out of $[0, 1]$ bounds, making them difficult to interpret as simple probabilities in a straightforward way. Classifications obtained using this method can be shown to be equivalent to those obtained using linear discriminant analysis (LDA) [71]. There are at least two main reasons not to use regression models in a classification problem:

- A regression model can't be used in a problem with more than two classes.
- A regression model won't predict meaningful conditional probabilities, even in the particular case of binary classification.

In this work, predicting whether a certain couple will get a divorce or not is key to designing more realistic agents for the agent-based model. This classification problem is clearly an imbalanced one, as the majority of couples registered as respondents in the Pairfam study don't usually have serious relationship problems or get divorced. In fact, the divorce rate among the married couples in the dataset used to train the different classifiers is around 2%, in line with the EU average annual rate of divorce.

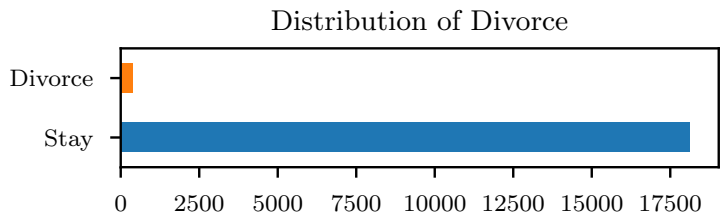

Figure 4.1: Instances of divorce in the data set.

Because predicting divorce straight-away can be a very complex classification task, the response variable finally chosen for the models in this work is actually whether the couple has serious relationship problems or not. Relationship problems are defined as one of the members of the partnership seriously thinking about separation or separation having been proposed and discussed in the couple. Separation and previous relationship troubles are linked and the problem is still an imbalance classification task, but the asymmetry between classes is less pronounced than before, having more observations from the minority class to learn from.

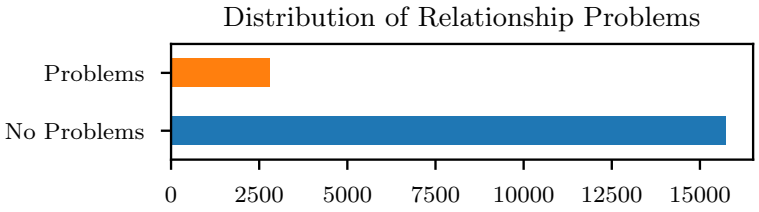

Figure 4.2: Instances of relationship instability in the data set.

Even though not all couples go through a phase of relationship trouble before getting a divorce, this variable is much easier to predict and correctly classify and can be more useful for the classification models. It has to be assumed that some romantic separations come suddenly and with little to no warning signs, and the modeling of that random nature will be left to the stochastic components of the agent-based simulation.

4.2 Imbalanced Classification

Classification predictive modeling involves assigning a class label to an instance, and it's one of the most extensively studied and practiced areas of ML. However, a significant portion of models and metrics used for learning from classification data assume an equal distribution of observations among class labels. This particular focus primarily addresses balanced classification problems, the simplest form of classification tasks.

On the other hand, when a clear imbalance in the distribution of instances across the different classes is present, many ML classification algorithms struggle and metrics like accuracy can become misleading. Many real-world problems like fraud detection, churn prediction, medical diagnosis and more often fall into the category of imbalanced classification and exhibit skewed class distributions. In fact, in many areas of research imbalanced classes are more prevalent than balanced ones [78]. Some

of the solutions proposed to solve this problem are adapting existing algorithms to suit imbalanced classification, making careful choices of performance metrics and developing new techniques for data preparation and modeling that specifically target this issue.

The degree of class distribution imbalance can vary across different problems in classification. A classification problem might exhibit a slight imbalance, where the distribution of instances is somewhat uneven in the training dataset (a $4:6$ ratio or similar), or it could suffer from severe imbalance, indicating a significant disparity in observation counts in the training dataset (a ratio of $1:100$ or more) [77]. In cases of a slight imbalance, concerns are often minimal, and the problem can be approached similarly to a regular classification task. However, severe class imbalance poses challenges for modeling and often needs the use of more specialized techniques.

The class that has many instances is called the major or majority class, while the class with a limited number of examples is known as the minor or minority class. When dealing with imbalanced classification, the minority class usually holds the greatest significance, so a model's accuracy in predicting the class label or probability for the minority class should carry more weight than the accuracy of the majority class. Predicting the minority class is often more difficult, because of its scarcity by definition [41]. A classification model usually faces difficulty in learning the unique traits of observations from this class and in distinguishing them from the majority class, as the sheer amount of instances in the majority class might sometimes overshadow those from the minority class.

Most ML algorithms designed for classification are tailored and demonstrated on problems that assume an even class distribution. Consequently, using such models in an unmodified manner might lead to a focus on learning from the abundant observations while neglecting the instances from the minority class. This is certainly problematic, since the minority class is often of greater interest and its predictions hold higher value than those of the majority class.

4.3 Model Evaluation Metrics

The effectiveness of a classifier is intrinsically tied to the metric used to evaluate its performance. Selecting an incorrect metric when evaluating models can result in the selection and use of subpar models or lead to misconceptions about the expected model performance [17]. The selection of an appropriate metric is a difficult task in applied ML, and this complexity is amplified when dealing with imbalanced classification problems. This challenge arises due to two main factors: most of the standard metrics that are commonly used assume a balanced distribution of classes, but not all classes hold equal significance in the context of imbalanced classification [101].

In contrast to conventional evaluation metrics that assign equal importance to all present classes, imbalanced classification tasks usually prioritize classification errors associated with the minority class over those involving the majority class, using performance metrics that specifically target said minority class. However, this task is complicated due to the lack of observations available for training a robust enough model within the minority class [17]. Usually, two different groups of metrics are taken into account: threshold metrics and ranking metrics [42].

4.3.1 Threshold Metrics

Threshold metrics are defined to capture the proportion of instances in which a predicted class does not align with the actual true class, using a separate validation dataset for testing. One of the most commonly used threshold metrics is classification accuracy, and its complementary counterpart known as classification error.

$$Accuracy = \frac{CorrPred}{TotalPred}$$

$$Error = \frac{IncorrPred}{TotalPred}$$

While accuracy is a commonly used evaluation metric in classification tasks, it is generally unsuitable for imbalanced classification scenarios. This is due to the fact that a high accuracy (or low error) can be achieved by a model that lacks predictive skill and merely predicts the majority class most or even all of the time. In the context of imbalanced classification problems, the majority class is often denoted as the negative outcome (0), while the minority class is typically identified as the positive outcome (1).

Most threshold metrics can be better understood using the concepts used to build a confusion matrix in the case of a binary classification task. The confusion matrix offers deeper insight into not just the effectiveness of a predictive model but also which classes are being accurately predicted, which ones are being predicted inaccurately and the nature of the errors being committed by the model.

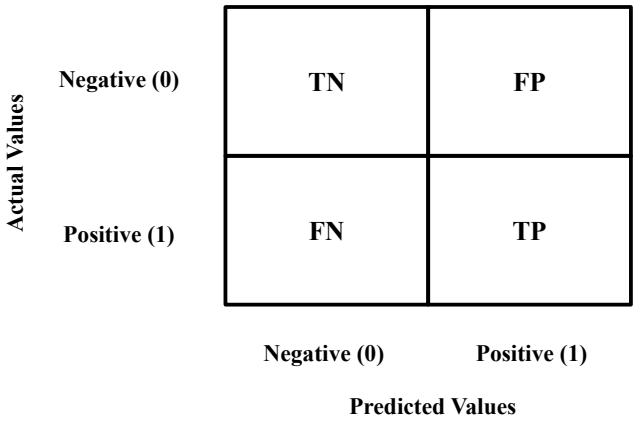

Figure 4.3: Definition of a confusion matrix for binary classification.

Two groups of evaluation metrics can prove useful for imbalanced classification due to their focus on a single class: sensitivity-specificity and precision-recall. Sensitivity, known as the true positive rate, captures the effectiveness of the model predicting the positive class. On the other hand, specificity, also known as the true negative rate, represents the accuracy of predicting the negative class. In imbalanced classification scenarios, sensitivity is often of greater interest than specificity, as it focuses naturally on the minority class and gives it more importance.

$$Sensitivity = \frac{TruePos}{TruePos + FalseNeg}$$

$$Specificity = \frac{TrueNeg}{FalsePos + TrueNeg}$$

On the other hand, precision quantifies the proportion of observations classified as the positive class that in reality belong to the positive class. Recall, which is equivalent to sensitivity, measures how accurately the positive class is predicted. These two metrics, precision and recall, can be merged into a single score that aims to strike a balance between both considerations. This composite score is known as the F-score, and it is a popular metric for imbalanced classification (Fernandez 2019).

$$Precision = \frac{TruePos}{TruePos + FalsePos}$$

$$Recall = \frac{TruePos}{TruePos + FalseNeg}$$

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

4.3.2 Ranking Metrics

Ranking metrics focus on evaluating classifiers based on their ability to effectively distinguish between classes. These metrics require the classifier to provide a prediction score or probability of class membership, and by applying various thresholds to this score the efficacy of the classifiers can be assessed. Models that maintain a strong score across a range of thresholds exhibit good class separation and are given higher ranks [41].

The ROC Curve, which stands for Receiver Operating Characteristic, is the most commonly used ranking metric and defines a methodology for analyzing binary classifiers based on their capacity to discriminate between classes. A ROC curve is a visual representation that summarizes a model's behavior by calculating the false positive rate and true positive rate for a set of predictions made by the model under different thresholds, with the true positive rate being equal to the recall or sensitivity of the model.

$$TruePositiveRate = \frac{TruePos}{TruePos + FalseNeg}$$

$$TrueNegativeRate = \frac{FalsePos}{FalsePos + TrueNeg}$$

In the ROC curve, every threshold corresponds to a point on the graph, and these points are then connected to create a curve. A classifier without any skill, such as one that predicts the majority

class for all thresholds, will be depicted by a diagonal line stretching from the bottom left corner to the top right corner. Points below this line indicate worse than no-skill performance, while a perfect model's point will be located at the top left corner of the plot.

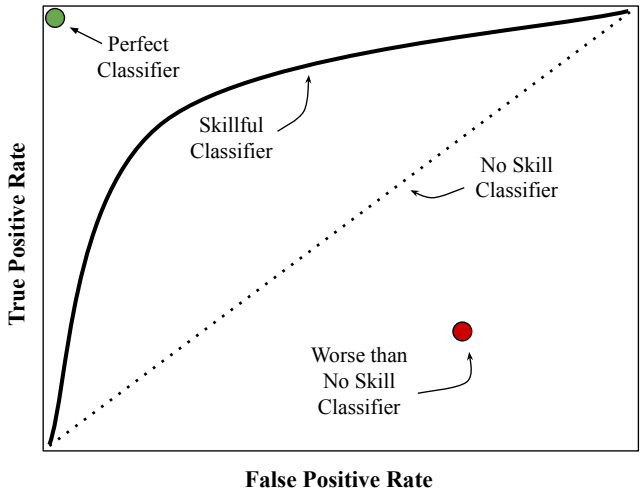

Figure 4.4: ROC curve.

The ROC Curve serves as a diagnostic tool for assessing a model's classification performance. By calculating the area under the ROC curve (ROC AUC), a single score is derived to summarize the curve, making model comparisons easier. An incompetent classifier will yield a score of 0.5, while a perfect classifier will achieve a score of 1.0. However, it's important to note that the ROC Curve and ROC AUC can tend to be overly optimistic in the presence of a significant class imbalance, particularly when the minority class comprises only a small number of observations [41].

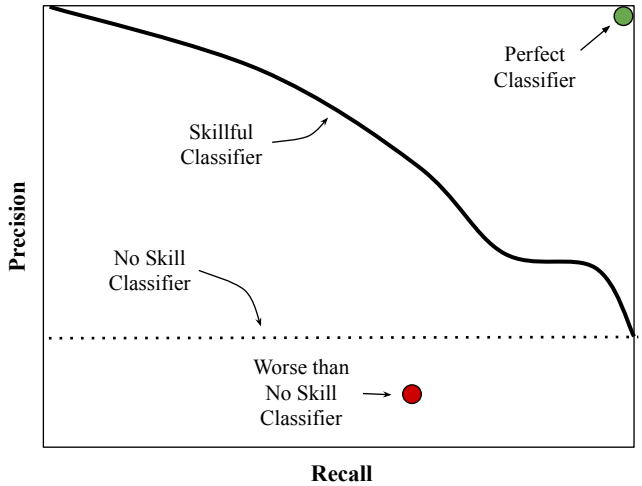

Figure 4.5: Precision-Recall curve definition.

The precision-recall curve offers an alternative to the ROC Curve, functioning in a similar way but emphasizing the classifier's performance on the minority class. By applying various thresholds to a set of model predictions, precision and recall are calculated, forming a curve. A no skill classifier

results in a horizontal line on the plot, with a precision score proportional to the number of positive observations in the data set, while a perfect classifier corresponds to a point at the top right.

Like the ROC Curve, the precision-recall curve serves as a useful diagnostic tool for assessing individual classifiers. The area under the precision-recall curve can be calculated as a score for comparison, equivalent to the metric also known as average precision, and its focus on the minority class renders it particularly valuable for imbalanced classification problems. In this work, the main metrics for model selection and comparison have been average precision or the area under the precision-recall curve and the F-score, both metrics with a special focus on the minority class.

4.4 Agent Models

Because the recorded characteristics and features of couples across the different waves of the Pairfam study are already known beforehand, predictions made by a model for the probability of relationship problems of each couple can be computed before the agent-based simulation even starts. Thus, the model can be trained at a population level to make these predictions, meaning that all agents share the same model, and these predictions can be accessed by the corresponding agents at each timestep of the simulation. This way, the agent-based simulation is computationally lighter, as it only needs to access a single probability number per agent every wave, instead of executing a model for each agent every time the probability of relationship problems is needed. In this section, some of the different population-level models proposed for this predictive classification task are presented, alongside their metric scores and feature importance evaluation.

4.4.1 Logistic Regression

Logistic regression is a classification method well-suited for the binary case, where only two possible classes are present in the response variable. Rather than modeling the corresponding class directly, logistic regression predicts the probability that an observation belongs to one of the two classes. This can be written as a conditional probability like $Pr(Probs|X)$, where X are the input variables for a specific couple. This probability is often written as $p(X)$, and will be in the range of $[0, 1]$. In this manner, a probability of divorce can be predicted for every set of recorded characteristics X. Classification in logistic regression is done using specific thresholds: divorce could be predicted when $pr(X) > 0.5$, for example. Other more conservative or aggressive classification models can also be implemented by changing the value of the threshold with which the classification is made.

In figure 4.6, the ROC and precision-recall curves corresponding to the logistic regression model can be seen, along their corresponding scores and the confusion matrix. Even though logistic regression can be considered a simple model in terms of complexity, the results obtained with little tuning are surprisingly decent. Because no weighting between classes has been done, it's clear that the model prioritizes classifying the observations belonging to the majority class (married couples that don't have relationship problems) correctly, as can be observed in the normalized confusion matrix.

Most ML algorithms assume that all errors in classification made by a model are of equal importance. However, this assumption does not hold true for imbalanced classification problems, where failing to detect a positive or minority class observation may carry a greater consequence than wrongly classifying an example from the negative or majority class.

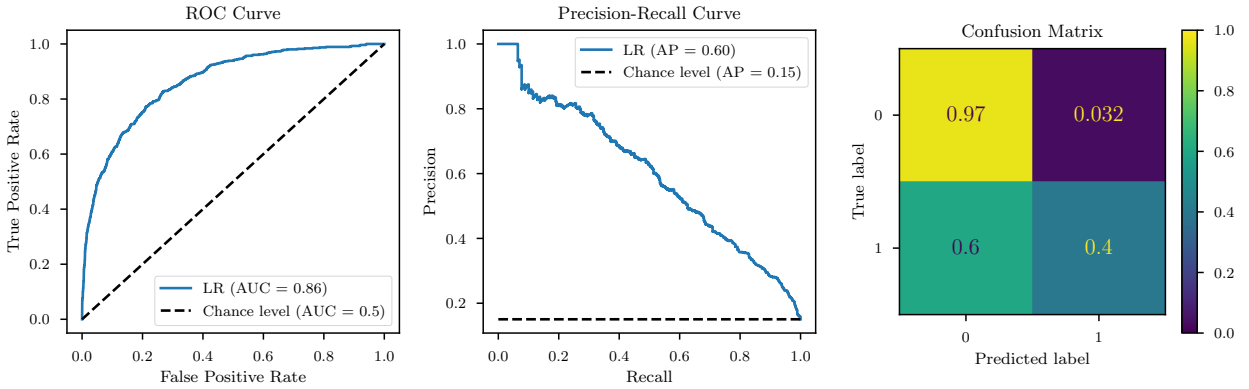

Figure 4.6: Logistic regression classification scores.

Cost-sensitive learning is a specialized area within ML that considers the associated costs of prediction errors during the training of a model. It falls within the broader context of imbalanced learning, which addresses classification tasks involving data sets with an uneven distribution of classes. As a result, many concepts and techniques developed in cost-sensitive learning are applicable to imbalanced classification issues.

Logistic regression doesn't directly handle imbalanced classification tasks. Rather, the training algorithm for fitting a logistic regression model must be adjusted to consider the imbalanced distribution of classes. This adjustment can be achieved by specifying a class weighting configuration, which affects the way logistic regression coefficients are updated during training.

The weighting assigned to each class can reduce the model's penalty for errors made on instances from the majority class and increase the penalty for errors on instances from the minority class. As a result, a modified version of logistic regression, often referred to as cost-sensitive or weighted logistic regression, is created. This adapted approach performs better on imbalanced classification problems, at least when it comes to predicting the minority class.

From a cost-sensitive learning point of view, logistic regression can be balanced by assigning proportional weights to the minority class, making it have the same importance as the majority class in the eyes of the classifier. While the ROC and precision-recall curve look fairly similar (with a slightly worse result in average precision in the balanced case), the most drastic change can immediately be seen in the confusion matrix. Instead of only guessing 5% of true divorces correctly, now the rate is around 73%, taking into account that a significant number of false positives has also been introduced (19% of non-divorcing couples are now predicted as divorcing). In this work, the results obtained for the logistic regression model will be considered as the baseline result when comparing the performance of different models.

Model Explainability

To measure the importance that the different features have when the model is making predictions, a method called permutation feature importance has been used. This method is a valuable technique used for model explainability that can be applied to any trained model when working with tabular

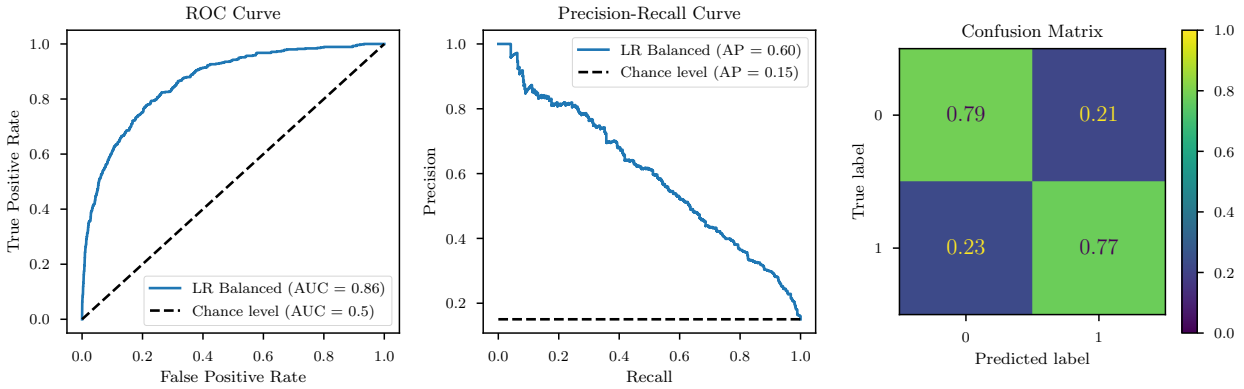

Figure 4.7: Balanced logistic regression classification scores.

data, which can prove particularly beneficial for non-linear or complex models that lack transparency.

Permutation feature importance involves measuring the change in a model's performance score when the value of a single feature is randomly shuffled. By doing so, the connection between the feature and the target is disrupted, making the reduction in the model's score a measure of the feature's importance to the model. This approach has the advantage of being independent of the specific model used and can be computed multiple times with different permutations of the feature.

Name	**Coeff**	**Mean**	**Std**
m_open	2.98	0.033	0.011
f_quality_rel	-2.00	0.030	0.011
f_sat_rel	-1.42	0.027	0.006
f_fut_orient	-1.10	0.025	0.003
m_fut_orient	-1.59	0.024	0.005
m_cheat	1.07	0.017	0.006
m_sat_rel	-1.17	0.016	0.006
probs_rel	1.00	0.014	0.012
f_comp_p	-0.71	0.013	0.005
m_age	-0.05	0.013	0.009
f_cheat	0.60	0.012	0.002

Table 1: Feature importance for balanced logistic regression.

In table 1, the most relevant features ordered by impact on model score when they are randomized are listed, alongside their logistic regression coefficients. If the coefficient is positive, an increase in that variable means a greater likelihood of divorce, and the contrary happens when the coefficient has a negative value. The absolute value of the coefficients is another way of measuring the impact of each specific feature, even though sometimes variables important for the model in the training set end up being less relevant when the model is evaluated in the test set (as is the case when permutation feature importance is done).

Among the most important variables for a healthy and stable marriage (at least according to this logistic regression model) we can find the satisfaction with the relationship and future orientation for both genders, the quality of the relationship and the ability of her partner to fulfill her needs as

perceived by the woman in the couple and, to a lesser extent, the age of the man in the couple. On the contrary, instances of cheating by both genders, serious problems in the relationship and, more surprisingly, how open the man in the couple considers himself play an important role in incentivising divorce.

4.4.2 Decision Trees

Tree-based methods can be used for classification tasks and they involve partitioning or segmenting the predictor space into more simple regions. Since the splitting rules that shape the final segmented predictor space can be compactly represented as a tree structure, these techniques are commonly referred to as decision tree methods [71]. In a classification tree, the prediction made for each observation is that it belongs to the class that is more represented among the training observations present in the region it falls into. When interpreting the outcomes of a classification tree, apart from the class predicted, understanding the distribution of classes in that specific terminal node is also of particular interest (the purity of the node).

The main objective of classification trees is to divide the dataset into smaller, more homogeneous groups. Homogeneity, in this context, implies that nodes resulting from splits are more distinct and contain a higher proportion of a single class within them. One way to measure purity in classification tasks is to maximize accuracy or, equivalently, minimize misclassification error. However, using accuracy as a measure of purity can be misleading, as it emphasizes partitioning the data to minimize misclassification errors rather than concentrating node samples primarily into one class [78]. To address this issue, two alternative measures, the Gini index and cross entropy, are usually employed to shift the focus from classification accuracy to purity. These measures are used to assess the quality of splits in classification trees and aim to create nodes that are more homogeneous or pure in terms of class proportions.

Trees constructed with maximum depth have a tendency to overfit the training data, meaning that they capture noise and details that do not generalize well to new observations. To create a more generalizable tree, pruning can be employed, a technique involving trimming back the initial tree to a more concise and shallow version. Another alternative is cost-complexity tuning, used to find an optimal balance between tree complexity and its performance on the training data. This tuning process penalizes the purity criterion by considering the total number of terminal nodes in the tree, resulting in a pruned tree. The complexity parameter, often called the complexity factor, plays a crucial role in cost-complexity tuning. By modifying this parameter, an optimal value that strikes the right balance between tree complexity and predictive performance can be estimated [78].

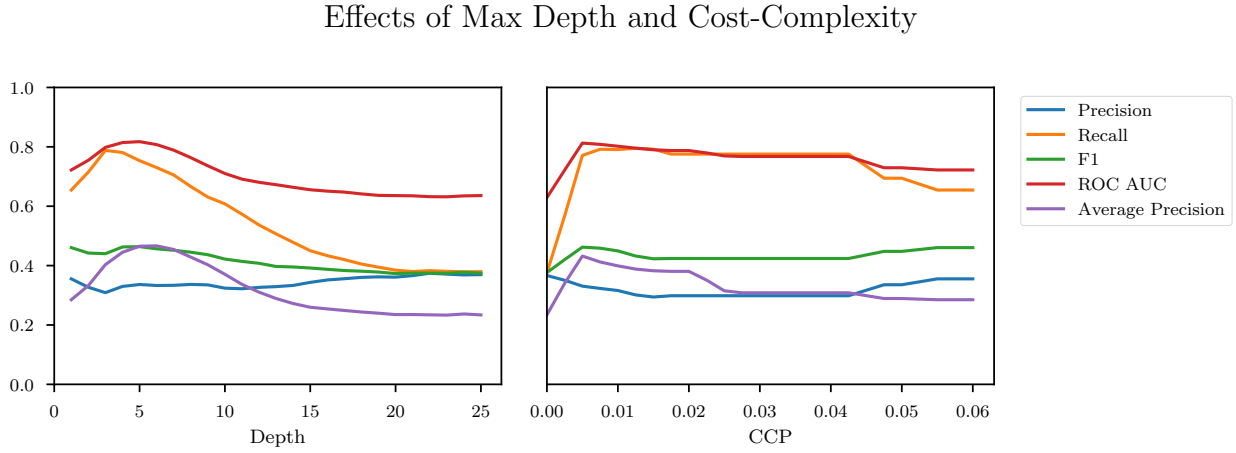

Figure 4.8: Metric scores for decision trees with different depth and cost-complexity parameter.

Looking at figure 4.8, it can be seen that the maximum scores in average precision are reached before the tree has a depth of 10 or with small cost-complexity parameter values. This shows that the generalization power of a decision tree greatly diminishes if the depth and complexity of the tree are left unchecked. Once the tree has been pruned, it can be used for making predictions. In classification tasks, each terminal node generates a vector of class probabilities based on the training set, and these probabilities are then used to make predictions for new observations. Selecting the optimal depth value of 6, the results obtained in the ROC and precision-recall curve are slightly worse than those obtained by logistic regression.

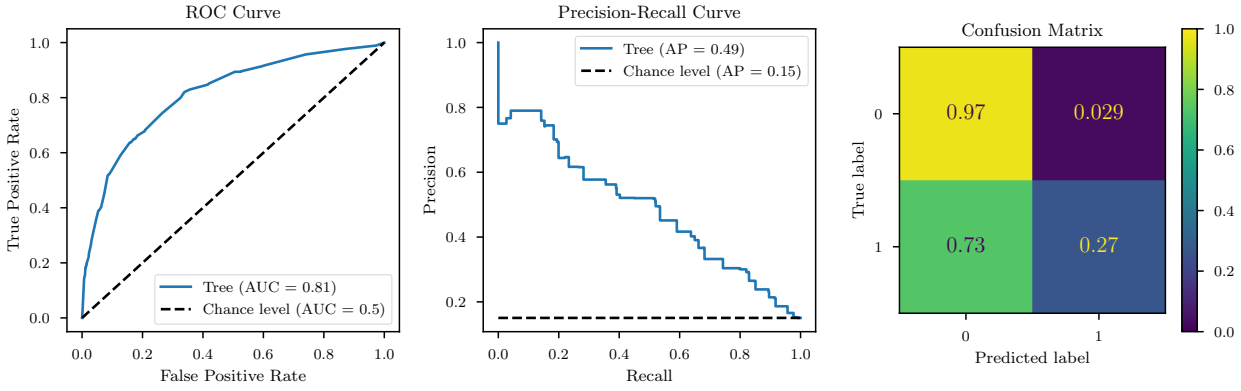

Figure 4.9: Decision tree classification scores.

Same as with logistic regression, decision trees can have their training samples weighted so both the minority and majority classes have the same importance when it comes to making predictions. Using these balanced weights, the results are still worse than those of balanced logistic regression.

Decision trees for classification offer several advantages over more traditional statistical models:

1. Explainability: Decision trees are highly intuitive and can be easily explained, often proving even easier to explain than linear regression models.

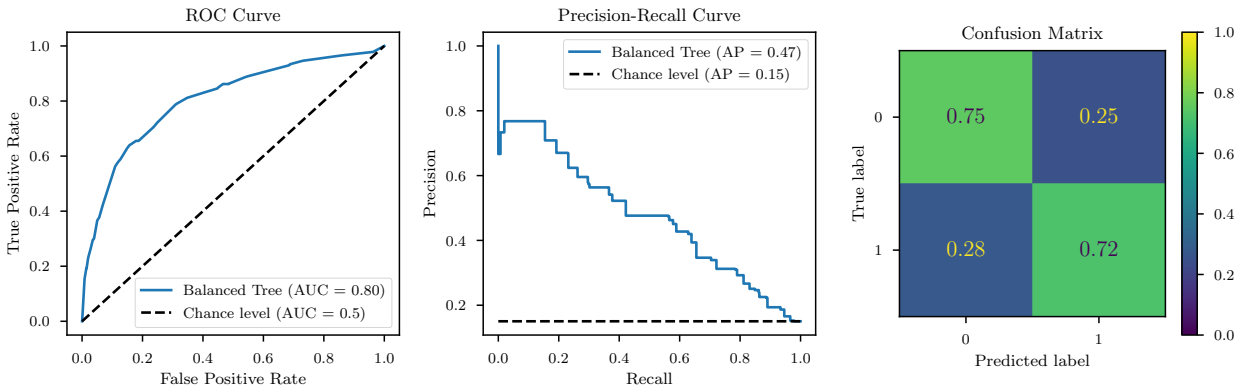

Figure 4.10: Balanced decision tree classification scores.

2. Resemblance to human decision-making: According to some, decision trees closely mimic human decision-making processes, compared to other more opaque classification methods.

3. Visual Interpretation: Decision trees can be visually represented, making them easily interpretable even by non-experts, particularly when the trees are small.

4. Qualitative Predictors: Decision trees can handle categorical response variables without requiring the creation of dummy variables.

However, decision trees also have certain limitations:

1. Predictive Accuracy: In general, decision trees may not achieve the same level of predictive accuracy as some other classification methods.

2. Non-Robustness: Decision trees can be sensitive to minor variations in the data, leading to significant changes in the final estimated tree and its structure if the training data changes.

Tree-based methods are straightforward and valuable for their interpretability. However, they often do not achieve the same prediction accuracy as some other supervised learning techniques. Therefore, additional techniques such as bagging, random forests and boosting are usually employed in order to make decision trees more effective. These methods involve generating multiple trees, which are then combined to produce a single consensus prediction. Thus, the fusion of a large number of trees can frequently lead to significant enhancements in prediction accuracy, thanks to a reduction in outcome variance, although this improvement may come at the cost of reduced interpretability of the model.

4.4.3 Bagging

Ensemble techniques involve combining predictions from multiple simpler models, and "bagging" (an abbreviation for "bootstrap aggregation") was one of the earliest examples of this methodology [19]. Bagging is a versatile approach that integrates bootstrapping with any regression or classification model to construct an ensemble. Each model within the ensemble contributes to generating predictions

for new observations, and these individual predictions are then averaged to produce the bagging model's ultimate prediction.

Bagging models have several advantages over non-bagging models. First and foremost, bagging effectively mitigates prediction variance through its aggregation and averaging process. For models prone to producing unstable predictions, such as decision trees, the act of aggregating over many versions of the training data leads to a significant reduction in result variance, therefore enhancing prediction stability. This implies that generating a different sequence of bootstrap samples, building models for each sample, and averaging predictions across models not only enhances the predictive performance of a bagging model but also improves its stability when compared to a non-bagging model.

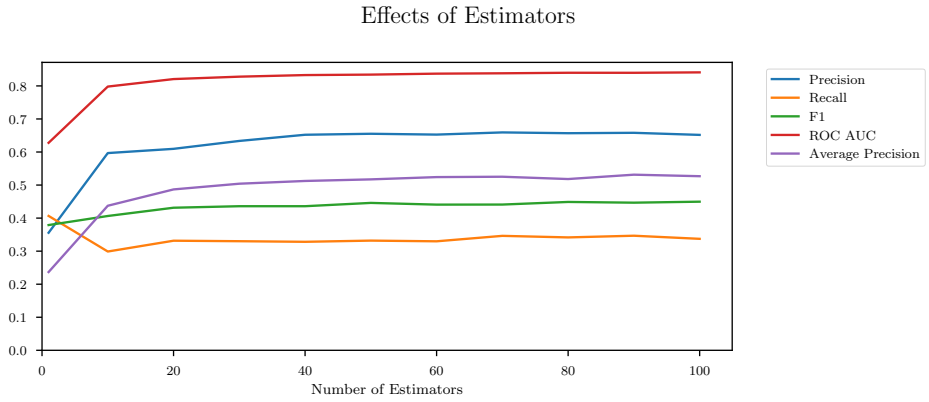

Figure 4.11: Metric scores for bagging ensembles with different number of estimators.

In the ensemble, each individual decision tree model is used to predict the class of a new observation. As all models carry the same weight in the ensemble, they can be viewed as casting votes for the class they believe the new sample belongs to. The total number of votes per class is then divided by the total number of models in the ensemble. This division results in a predicted probability vector for the sample, with the new observation being classified into the group with the most votes and highest probability. Thus, one of the most important hyperparameters is the number of estimators that are present in the bagging ensemble,

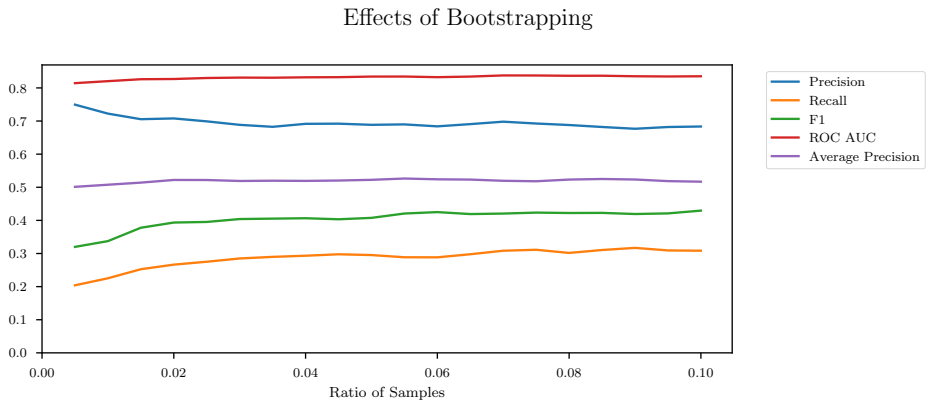

Figure 4.12: Metric scores for bagging ensembles with different sampling ratio.

As can be seen in figure 4.11, the average precision score of the bagging ensemble starts to plateau around 40 estimators. Another important parameter of a bagging classifier is the percentage of the original training samples to be used when training the individual weaker decision trees, as bootstrapping is a crucial concept in bagging. In this particular classification task, it looks like the effect of sample percentage is not very impactful in the final results, the optimal value being around 5.5% of the original number of samples (figure 4.12).

The results obtained for the ROC and precision-recall curve are slightly better than those of logistic regression, making it clear that bagging is an effective method to turn weaker learners like decision trees into much more effective classification models (figure 4.13).

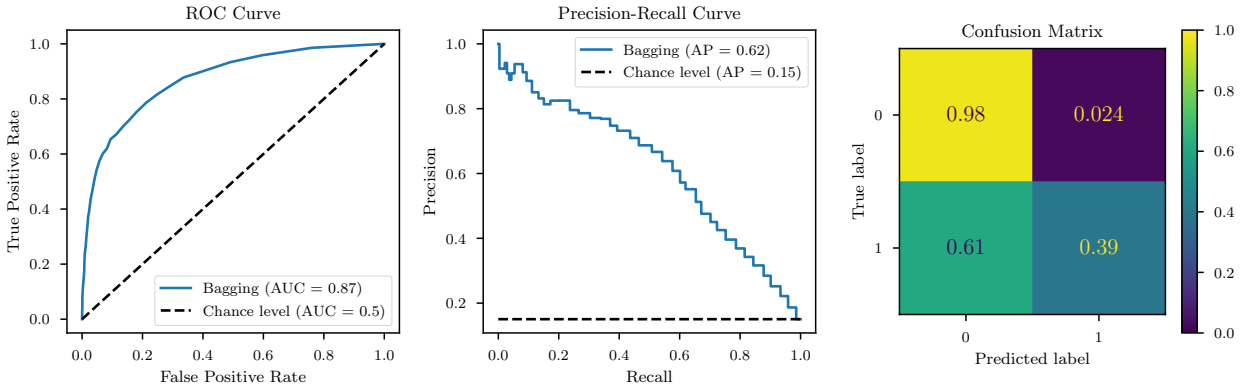

Figure 4.13: Bagging ensemble classification scores.

There's also the possibility to get a balanced version of the bagging model by weighting the classes accordingly. The results in the balanced case are also better than those of the balanced baseline model, obtaining a significant reduction in the percentage of false positives (figure 4.14).

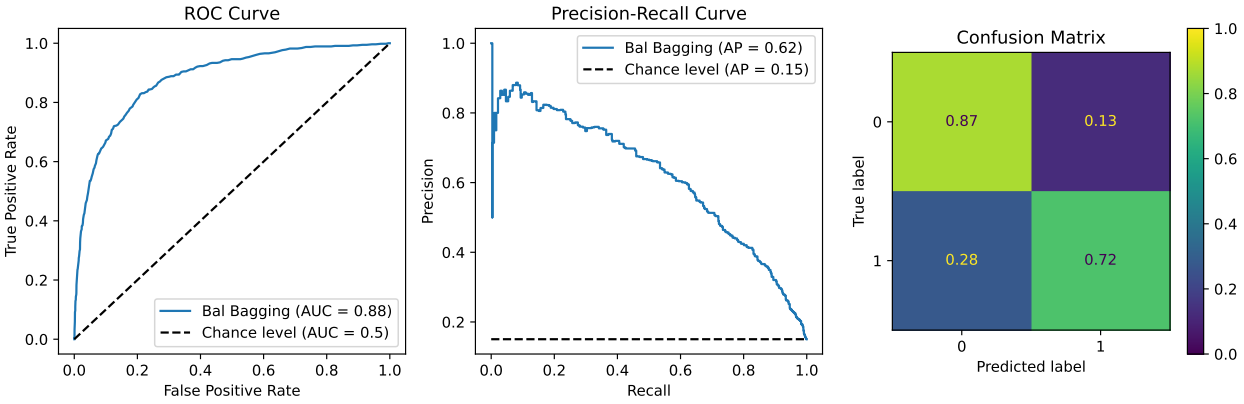

Figure 4.14: Balanced bagging ensemble classification scores.

Model Explainability

In the case of the bagging ensemble, the most important variable is clearly the satisfaction with the relationship that the woman in the marriage has, with a drop in average precision close to the

7.7% of the original score if this single variable is randomized. Other important variables for married women are the quality of the relationship and whether there are conflicts in the partnership or not. In the case of men, the perceived satisfaction of the woman in the couple is the most significant variable, closely followed by the quality of the relationship.

Feature	Mean	Std
f_sat_rel	0.077	0.009
f_quality_rel	0.027	0.006
m_sat_rel_p	0.020	0.005
m_quality_rel	0.016	0.005
f_conflict	0.016	0.005

Table 2: Feature importance for balanced bagging ensemble.

4.4.4 Random Forest

Bagging considers all original features at every split of each tree, so the decision trees generated are not entirely independent. This phenomenon, known as tree correlation, arises due to the underlying relationship between features and the responses variable. As a result, trees from different bootstrap samples may share similar structures, particularly at the top of the trees. This correlation prevents bagging from fully optimizing the reduction of prediction variance. Addressing this limitation involves reducing correlation among trees, a process referred to as de-correlating trees, and represents the next step for improving bagging's performance.

To achieve reduced correlation among trees, randomness is incorporated into the tree construction process. Various authors proposed modifications to the original bagging algorithm, introducing randomness into the learning process. For example, random split selection, where trees are built using a random subset of the top k predictors at each split or perturbing the tree structure by adding noise to the response variable. After thorough evaluation of these modifications, the unified random forests algorithm was developed, encompassing all these new ideas [20].

Random forests, by introducing randomness during split selection, effectively reduce tree correlation and improve predictive performance. The selection of strong learners with low bias, and the combination of many independent, strong learners, results in a variance reduction of the ensemble as a whole, leading to enhanced error rates. Due to the independence of learners in random forests, they are also robust to noisy responses. However, excessive independence can cause underfitting when the response is not really noisy.

Compared to bagging, random forests offer computational efficiency on a per-tree basis by evaluating only a fraction of the original features at each split. While random forests usually require more trees, their efficiency, coupled with the potential for parallel tree construction, makes them more computationally efficient than boosting.

It seems like the number of estimators reaches a plateau around 75 estimators in figure 4.15, so that

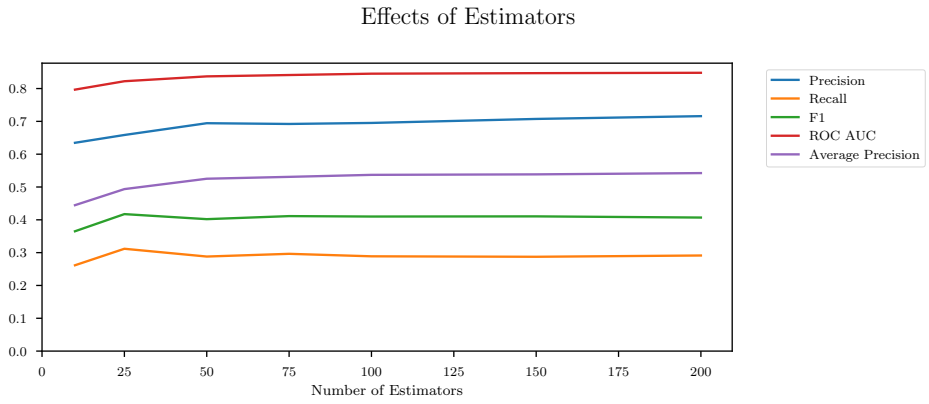

Figure 4.15: Metric scores for random forests with different number of estimators.

is the value used for the random forest algorithm in this work. A common value for the number of features considered in each split of the individual decision trees is the square root of the total number of features, and that number has also been chosen here. The results obtained in both curves are similar to those of normal bagging, so it seems that in this particular task the improvements introduced in the random forests algorithm did not have that big of an impact, although the final scores are slightly better in average precision (figure 4.16).

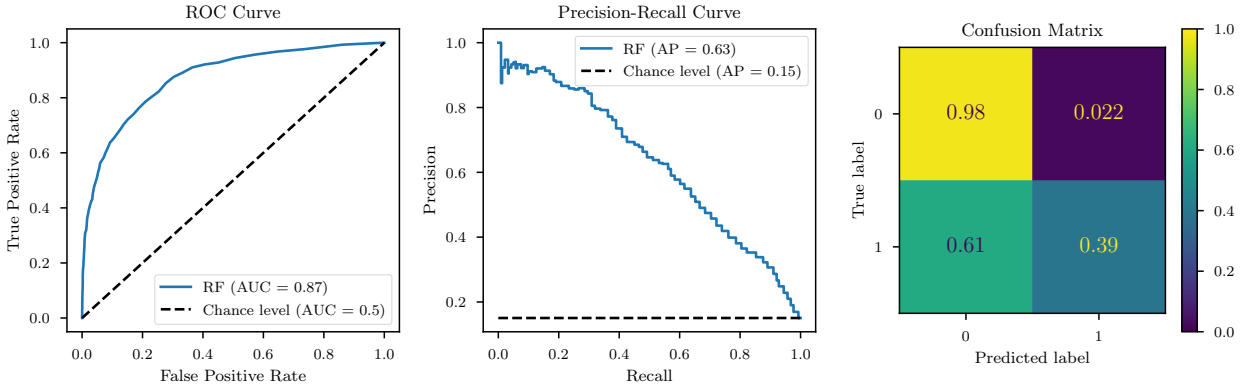

Figure 4.16: Random forests classification scores.

Random forests can also be balanced with weighted classes and, once more, the obtained metric results are similar to those of balanced bagging. In this case, the model tends to more aggressively predict the minority class, improving the ratio of true positives at the cost of more false positives (figure 4.17).

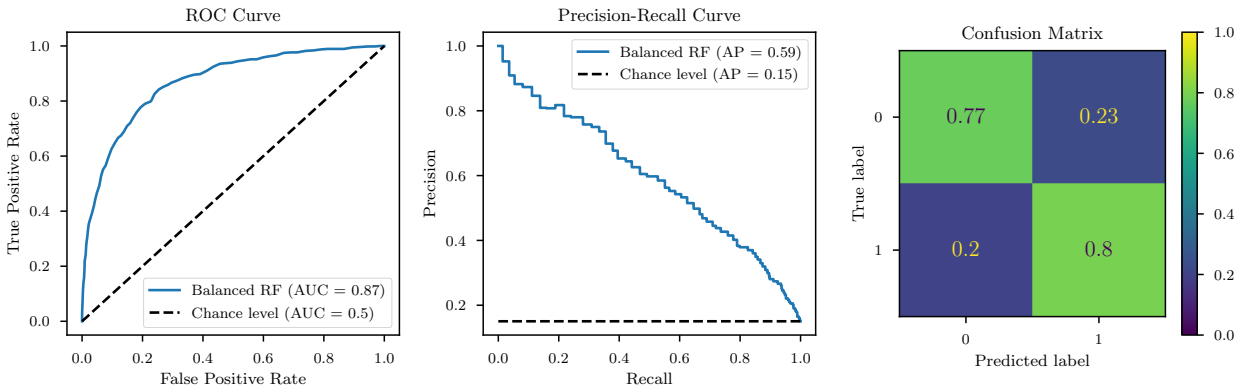

Figure 4.17: Balanced Random Forests classification scores.

Model Explainability

In the case of random forests, the crucial variable is once again the satisfaction of the married woman with her marriage, closely followed by the frequency of quarrels and conflicts from the woman's point of view. In this case, the perceived satisfaction by the woman of her partner is also an important feature for classification. In the case of married men, the overall quality of the relationship and their perceived frequency of conflicts are also important, but less important than their female counterparts.

Feature	Mean	Std
f_sat_rel	0.035	0.005
f_conflict	0.023	0.004
f_sat_rel_p	0.018	0.005
m_quality_rel	0.011	0.006
m_conflict	0.009	0.004

Table 3: Feature importance for balanced random forests.

4.4.5 Gradient Boosting Methods

The history of boosting begins with the AdaBoost algorithm [47] and evolves to the stochastic gradient boosting machine [48], which is now widely regarded as the boosting algorithm of choice. The core idea is to combine multiple weak classifiers (classifiers that predict marginally better than random) to create an ensemble classifier with a significantly improved generalized misclassification error rate.

Boosting ensemble algorithms work by generating a series of models that aim to rectify the errors made by preceding models in the sequence. These models then produce predictions, which can be

assigned weights based on their demonstrated accuracy. The outcomes of these predictions are then aggregated in a weighted manner to produce a final combined prediction.

AdaBoost proved to be a powerful prediction tool, often outperforming individual classification models. Researchers later connected the AdaBoost algorithm to statistical concepts such as loss functions, additive modeling, or logistic regression and showed that boosting can be interpreted as a forward stage-wise additive model that minimizes exponential loss. Eventually, this insight into the statistical framework of boosting resulted in the development of gradient boosting machines.

The fundamental principles of gradient boosting involve minimizing a loss function using a weak learner to create an additive model. The algorithm begins with an initial guess of the response, calculates the gradient, fits a model to the residuals to minimize the loss function and adds the current model to the previous one. This process then continues for a specified number of iterations. Since boosting requires a weak learner, nearly any technique with tuning parameters can be turned into a weak learner. Trees are particularly well-suited as base learners for boosting for several reasons: they can be made weak by restricting their depth, they can be easily combined and they can be generated quickly. When using trees as the base learner, simple gradient boosting for classification involves tuning parameters like tree depth and the number of iterations.

Boosting shares some similarities with random forests, but there are also significant differences in how the ensembles are built:

1. Independence vs. Dependence: In random forests, each tree is constructed independently of the others. They are created to have maximum depth, and each tree contributes equally to the final model. In contrast, boosting builds trees sequentially, and each tree depends on the previous ones. Boosting trees are typically shallow (weak learners), and they contribute unequally to the final model.

2. Over-fitting Considerations: Random forests tend to be more resistant to over-fitting because each tree is independent and has maximum depth, which helps reduce variance. In boosting, there is a risk of over-fitting because the optimal learner is being selected at each stage, even though these learners are weak. This greedy strategy can lead to over-fitting.

3. Regularization: To address over-fitting, boosting can employ regularization, similar to techniques used in linear regression. By introducing a learning rate, only a fraction of the current predicted value is added to the previous iteration's predicted value. This learning rate becomes a tuning parameter, and small values are often recommended to prevent over-fitting, but this requires more iterations and memory.

4. Stochastic Gradient Boosting: To further improve boosting's prediction accuracy and reduce computational resources, a modification known as stochastic gradient boosting was introduced. This approach randomly selects a fraction of the training data for each iteration. The residuals and models in that iteration are based only on this sample of data. The fraction of training data used, called the bagging fraction, becomes another tuning parameter for the model, with a common value being around 0.5.

In summary, while both random forests and boosting can offer competitive predictive performance, they differ in terms of independence of trees, tree depth, and contribution to the final model.

Boosting, when combined with regularization and stochastic sampling, can help mitigate over-fitting and improve prediction accuracy, but it often requires more computational resources than random forests. The choice between these two ensemble techniques often depends on the specific problem and the trade-offs between computational efficiency and predictive performance.

Three different implementations of the boosting ensemble method have been considered in this work: XGBoost, LightGBM and CatBoost. Due to the imbalanced nature of the classification task at hand, different weights for the minority class have been considered in each of the methods. The results of trying different weights can be seen in figure 4.18.

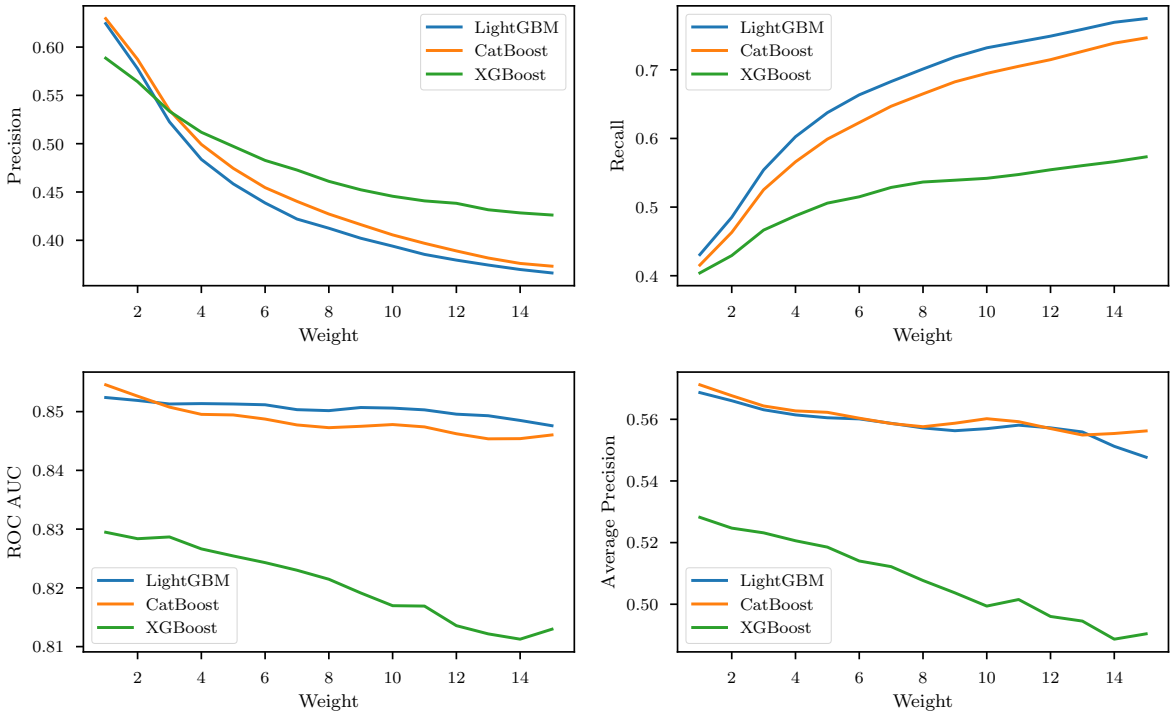

Figure 4.18: Metric scores for boosting implementations with different weights.

Looking at the metric scores for average precision, it looks like the best weight for the minority class is 1, just leaving the problem in its imbalanced state. But taking a closer look at the F-Score (figure 4.19), another of the useful metrics for imbalanced classification tasks, we can see that the optimal value ranges around 4 to 6, depending on the implementation chosen. All in all, it looks like XGBoost falls a little behind in all metric scores compared to LightGBM and CatBoost.

XGBoost

The XGBoost algorithm, short for Extreme Gradient Boosting, is a powerful algorithm for classification predictive tasks. It belongs to the ensemble learning family and is known for its exceptional predictive performance and efficiency and efficiently implements the stochastic gradient boosting algorithm, providing a variety of hyperparameters that allow for precise control over the training process. It implements L1 and L2 regularization techniques in order to prevent overfitting, making it more robust and less prone to high outcome variance.

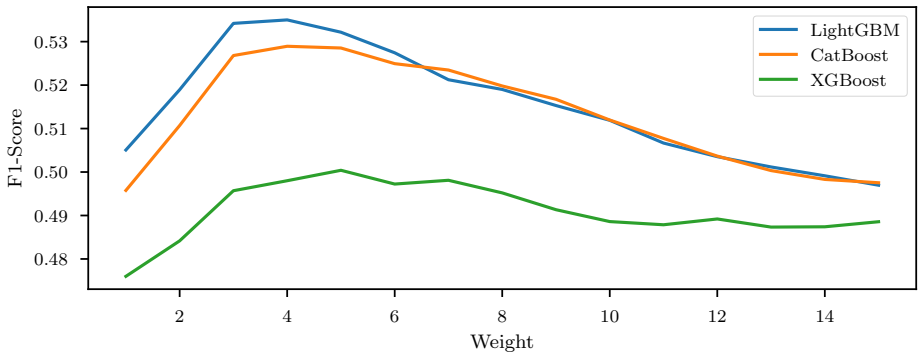

Figure 4.19: F-scores for boosting implementations with different weights.

While it generally performs effectively, even on data sets with imbalanced class distributions, XGBoost also provides an avenue to fine-tune the training process to prioritize the correct classification of the minority class in situations where class imbalances exist, using the weight parameter discussed in the previous section. The results obtained using XGBoost are clearly better than those of the baseline model and of the other ensemble methods considered before (bagging and random forest), showcasing the power of boosting as an ensemble method in this classification task (figure 4.20).

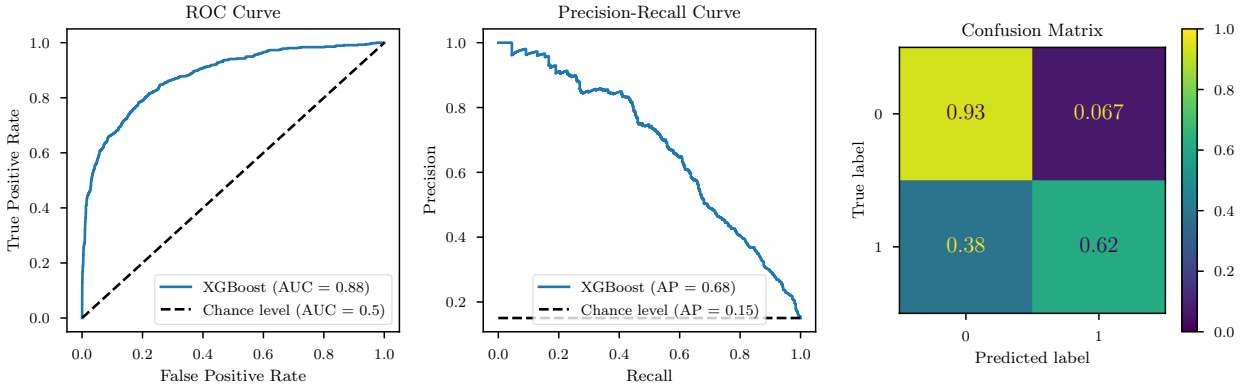

Figure 4.20: Balanced XGBoost classification scores.

LightGBM

LightGBM (Light Gradient Boosting Machine) is an open-source, distributed, high-performance gradient boosting framework for machine learning tasks. It is specifically designed to be efficient, fast, and memory-friendly, making it well-suited for handling large data sets and complex problems. LightGBM is written in C++ and supports multiple programming languages, including Python, R, and more.

LightGBM uses a histogram-based learning approach to optimize the decision tree building process. It bins data points into histograms and calculates gradients within these bins, resulting in a faster and more memory-efficient training process compared to traditional tree-based methods. It also uses a leaf-wise tree growth strategy, which differs from the level-wise strategy used in some other

gradient boosting implementations. This approach aims to reduce the loss more aggressively by growing deeper trees and then pruning them, which can lead to better model performance.The results obtained using LightGBM with the optimal weight are slightly better than those of XGBoost, but the difference is minimal (figure 4.21).

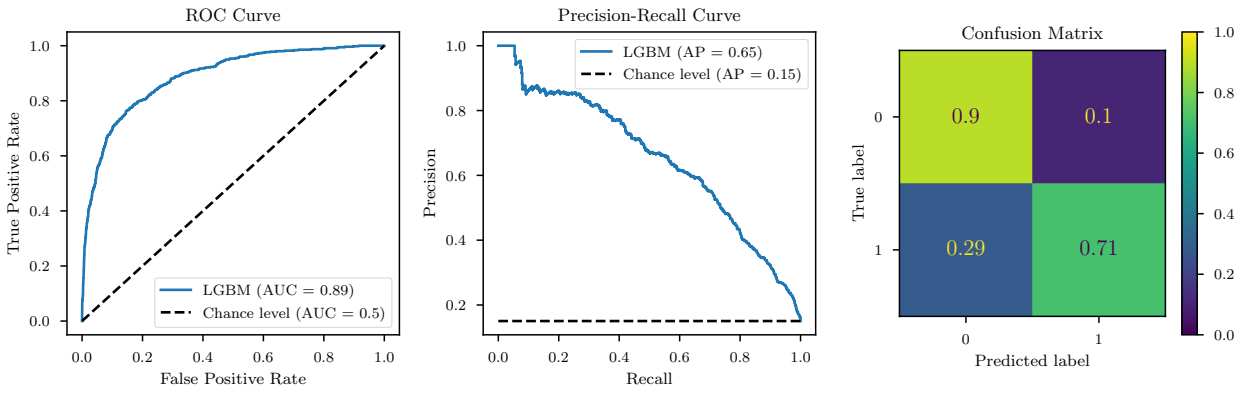

Figure 4.21: Balanced LightGBM classification scores.

CatBoost

CatBoost, short for Categorical Boosting, is a high-performance gradient boosting library for machine learning developed by Yandex, a Russian multinational IT company. CatBoost is designed to handle categorical features naturally and efficiently, making it particularly useful for tasks where you have a mix of numerical and categorical data. CatBoost supports multiple programming languages, including Python, R, and others, making it accessible to a wide range of users.

CatBoost includes regularization techniques like depth regularization and learning rate annealing to prevent overfitting. These techniques make it more robust when dealing with complex data sets. The results obtained by CatBoost are the best ones obtained yet, being even better than XGBoost and LightGBM in classifying the observations of the test dataset, so for the rest of this work this has been the chosen model for predictive classification tasks (figure 4.22).

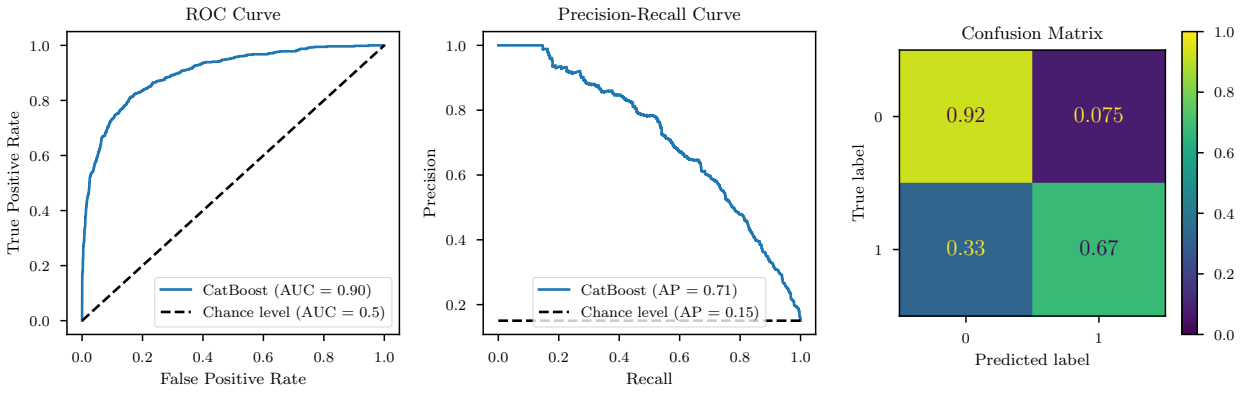

Figure 4.22: Balanced CatBoost classification scores.

Model Explainability

In the case of CatBoost, the most important variable when predicting if a relationship is stable or insatiable is the satisfaction of the woman in the couple, once again (table 4). The perceived satisfaction of the woman by the man in the marriage also plays a role, but in a less significant way. Other new variables that appear as important in this boosting ensemble are the duration of the relationship, the frequency of sex and instances of extra-marital affairs by the man.

Feature	Mean	Std
f_sat_rel	0.037	0.005
reldur	0.009	0.002
m_sat_rel_p	0.009	0.003
sex_freq	0.007	0.002
m_cheat	0.007	0.002

Table 4: Feature importance for balanced CatBoost.

4.5 Comparison of Models

A comparison of the ROC and precision-recall curves obtained by the different models is shown in figure 4.23. As said above, the gradient boosting ensemble best all other models in a very significant way, in particular the CatBoost implementation.

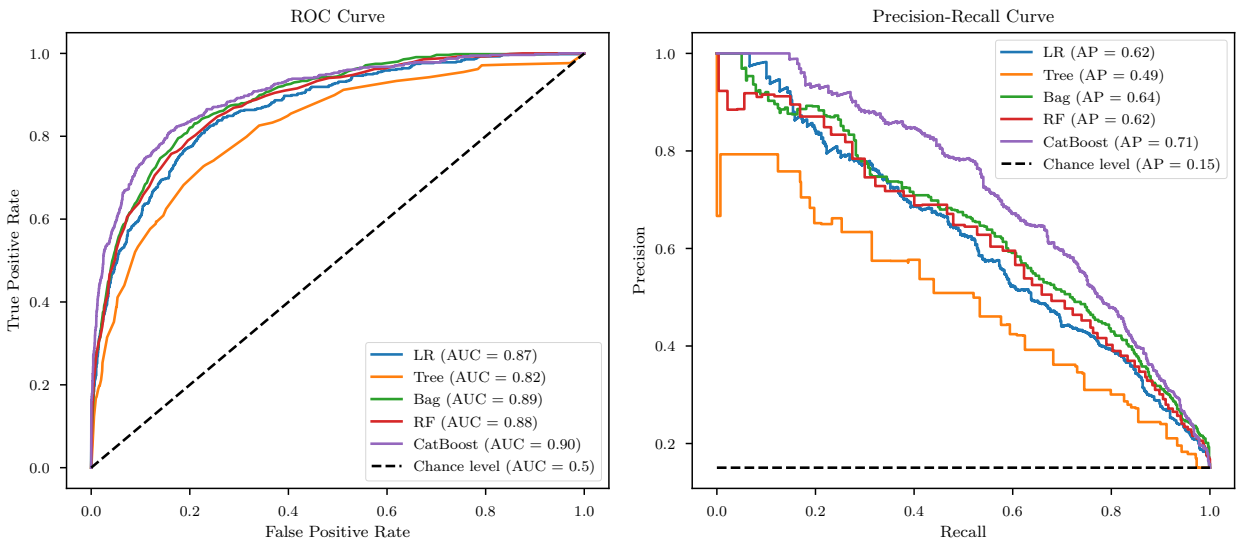

Figure 4.23: Balanced CatBoost classification scores.

5 Agent-Based Simulation

In this section, the descriptions and main results of the implemented agent-based models for the social contagion of divorce are discussed. First, the characteristics of the chosen Python library (AgentPy) used to design and run the simulations are presented, followed by the general simulation configuration that is common to all the different models and an explanation of the theoretical aspects of the employed sensitivity analysis. Then, the models are described and their results analyzed, using said sensitivity analysis and visualizing simulation outcomes. Lastly, all three models are compared against each other.

5.1 AgentPy

AgentPy is an open-source library designed for the development and analysis of agent-based models [43], and it is the modeling framework that has been used in this work. Some key features and characteristics of AgentPy that stand out compared to other alternatives are the following:

1. Python 3: AgentPy is written in Python 3, which is a widely-used and versatile programming language for scientific computing with Object-Oriented Programming capabilities that make it a good fit for agent-based modeling.
2. Interactive Computing: The library is optimized for interactive computing with IPython and Jupyter, making it convenient for researchers to work with agent-based models in an interactive and exploratory manner.
3. Comprehensive Environment: AgentPy provides a comprehensive environment for agent-based modeling tasks, including creating custom agent and model types, running interactive simulations, conducting numeric experiments across multiple runs and analyzing the resulting data. These tasks can all be performed within a Jupyter Notebook.
4. Scientific Applications: AgentPy is designed with scientific applications in mind. It offers tools for parameter sampling, performing Monte Carlo experiments, generating random numbers, using parallel computing, and conducting sensitivity analysis.
5. Integration with Python Libraries: The library is built to work seamlessly with established Python libraries such as EMA Workbench, NetworkX, NumPy, pandas, SALib, SciPy and seaborn. This compatibility allows researchers to leverage a wide range of tools and resources for their modeling and analysis needs.
6. User-Friendly Syntax: AgentPy provides an intuitive syntax for creating agent-based models, making it accessible to both newcomers and experienced practitioners in the field of agent-based modeling.

Overall, AgentPy offers a user-friendly and powerful environment for developing, simulating, and analyzing agent-based models, particularly within the context of scientific research and analysis. It streamlines the process of creating and working with agent-based models, making it a valuable tool for researchers. The AgentPy framework is structured in a nested manner, and the core components include agents, environments, models, experiments, and data analysis. Here's an overview of this nested structure:

- Agents: Agents are the fundamental building blocks in AgentPy. They represent individual entities or actors within an agent-based model. Agents can have behaviors, attributes and can interact with their environment and other agents.
- Environments: Agents can be placed within one or more environments, defining the spatial or network context in which agents interact. AgentPy supports various types of environments, including networks, spatial grids, and continuous spaces. These environments provide the stage for agent interactions.
- Models: Models are responsible for initializing agents, defining agent behaviors and attributes and running simulations. They serve as the main control structure for the agent-based model and set the initial conditions, update agents over time, and record data during simulations.
- Experiments: Experiments involve running the model over multiple iterations and different parameter combinations. This allows for sensitivity analysis, Monte Carlo simulations and exploring the model's behavior under different conditions. Experiments help understand how the model behaves across a range of scenarios.
- Data Analysis: The output data generated from experiments can be saved and then analyzed. This analysis may involve processing and visualizing the data to extract insights, patterns and trends from the simulation results.

In summary, AgentPy provides a structured framework for developing agent-based models. Agents interact within environments, and models coordinate these interactions to simulate complex systems. Experiments allow for systematic exploration of model behavior and data analysis tools assist in interpreting and visualizing the simulation outcomes.

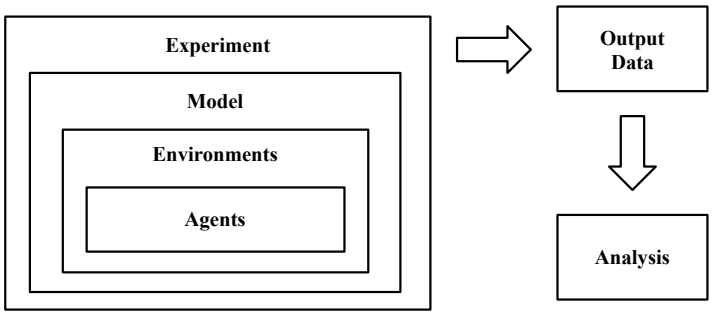

Figure 5.1: AgentPy structure.

5.2 Simulation Configuration

In all model configurations 8 time steps are simulated, corresponding to waves 2 to 9 of the Pairfam longitudinal study, while the first wave 1 is reserved for initialization purposes. This way, the couples that get a divorce in wave 1 of the study are used as the first infected agents in the simulation and, thus, all runs start with the same couples getting divorced in timestep 0, instead of randomly infecting a percentage of the total population like other epidemiological models tend to do. The total number of agents used is 4295 and the distribution of divorce across the time steps can be seen in figure 5.2.

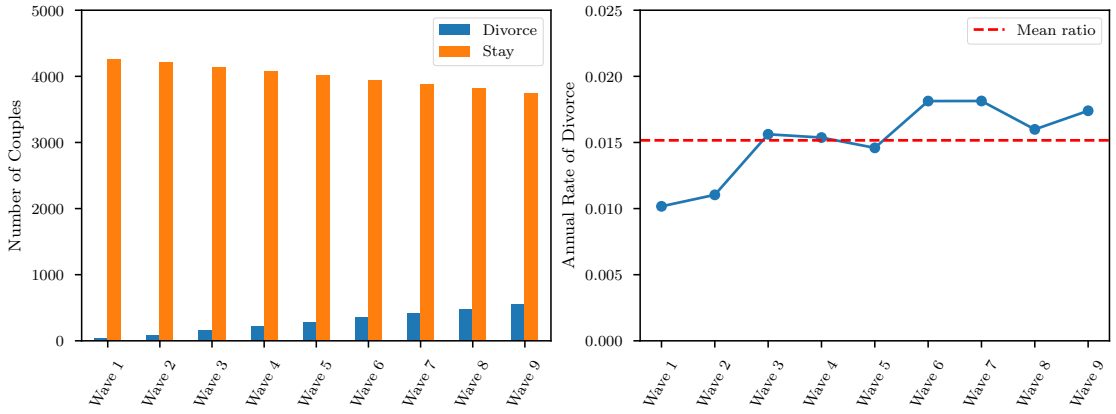

Figure 5.2: Divorce rates across Parifam waves.

Agents are then initialized, randomly shuffled and placed in a small-world Watts-Strogatz network graph as its nodes, with a network randomness value of 0.05. This means that every timestep each couple has a 5% chance of developing new connections with other agents in the graph, and mainly serves as a way of introducing some dynamism into the different runs and simulating the creation of new friendship links between couples.

The number of connections each couple has at the same time is 3, as this value has shown the best results in initial runs. Statically fixing these parameters (network randomness and number of neighbors) greatly reduces the dimensionality of the simulation experiments required for conducting sensitivity analysis. After every timestep, the total number of agents that are divorced is registered as a reporting variable and at the end of each simulation run the $L2$ norm is computed against the true empirical number of divorced couples in each time step.

5.3 Sensitivity Analysis

Sobol sensitivity analysis is a valuable technique for understanding how the uncertainty in the output of a mathematical or computational model is influenced by variations or uncertainties in its input parameters. It aims to quantify and link this uncertainty to sources of variation in the model inputs, using different types of sensitivity indices to that end:

1. First-Order Indices: These indices quantify the contribution of a single model input to the overall variance in the model output. In other words, they measure how sensitive the output is to variations in a single input parameter while keeping the others fixed.

2. Second-Order Indices: Second-order indices measure the impact of interactions between pairs of model inputs on the output variance. They measure the extent to which variations in the output result from the joint effects of two specific input parameters.

3. Total-Order Index: Total-order indices provide a comprehensive measure of the contribution of a model input to the output variance. They consider not only the first-order effects (individual input variations) but also all higher-order interactions, including second-order, third-order and

so on. Total-order indices give a more complete picture of the influence of an input parameter on model output, accounting for all possible interactions.

High first-order sensitivity indices indicate that a parameter has a strong individual influence on the output, while high total-order sensitivity indices suggest that a parameter has a significant influence both individually and through interactions with other parameters. The key idea behind Sobol sensitivity analysis is to decompose the total variance in the model output into different components, each corresponding to the contribution of a specific input parameter or a combination of parameters.

Sensitivity analysis is a process with a few steps. First, the parameters to be tested have to be defined, with their range of possible values, and n sets of randomized input parameters are generated. Then, the model is run n times and the results are captured. These results can be analyzed to identify the most/least sensitive parameters. As the measurements of second-order indices are somewhat encapsulated in the results of total-order indices, they have not been computed in this work with the goal of saving computation time. Thus, Sobol sensitivity analysis is a valuable tool for gaining insights into the behavior of complex systems and improving the understanding of which parameters are most influential. It is particularly useful in situations where uncertainty in model inputs needs to be quantified effectively.

5.4 Simple SIR Model

In the most basic of infectious disease models [6], individuals are categorized into two main states: susceptible, denoting the absence of the disease, and infected, signifying the presence of the disease. Transmission of the disease can occur only when a susceptible person interacts with an infected individual. The rate of transmission from an infected individual to a susceptible one can be quantified and is commonly called the transmission rate. Following infection, an individual recovers from the disease at a constant rate, known as the recovery rate, regardless of their interactions with either susceptible or infected individuals.

In a specific category of disease models named Susceptible-Infected-Recovered (SIR), those who recover gain immunity and shift to a recovered state. Another common configuration is known as the Susceptible-Infected-Susceptible (SIS) model, used to simulate infectious diseases that don't confer immunity, such as numerous sexually transmitted diseases. In this configuration, recovered individuals can revert to being susceptible and might be infected again after that reversion.

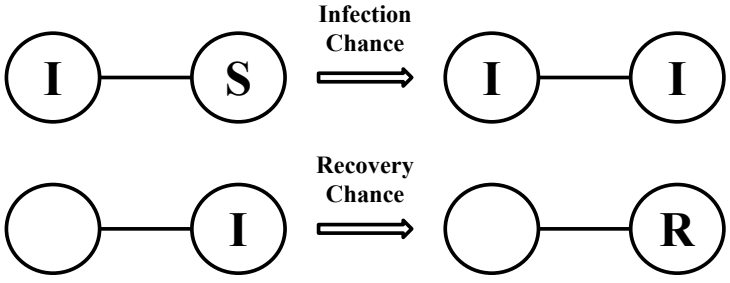

Figure 5.3: Mechanisms of infection and recovery in a SIR model.

In this work, separations are treated as non-reversible and the recovery rate is used to measure the length of the social impact of such marital dissolutions in their social circles. This way, when a separated couple is recovered, they can no longer impact their social network neighboring couples in any way. Every time step, the agent-based model checks all agents and selects those that are infected / divorced. Every infected agent has then a chance of spreading their infection onto all their susceptible neighbors, defined by the simulation parameter *infection chance*. If the infection is successful, the newly infected couple changes their condition to infected and get divorced.

All infected agents can also recover, effectively erasing the effect of the couple's divorce from their social circle. This recovery rate is defined by the simulation parameter recovery chance and only affects agents that are already infected. Because recovered agents can act as bottlenecks in the transmission of infection (they can not get infected or infect others), the ability to create new social links thanks to the network randomness parameter (5%) can be helpful in avoiding premature ends to the social contagion of divorce.

5.4.1 Sensitivity Analysis

In the case of the SIR model, it is clear that the parameter with the most significant impact in the final error of the simulation runs is the infection chance (figure 5.4). This parameter controls the transmission rate of divorce between couples in the simulation, and it has a bigger impact than the recovery chance does. Even though both parameters have high first-order values, they also display high total-order indices, meaning that the interaction between them is also significant.

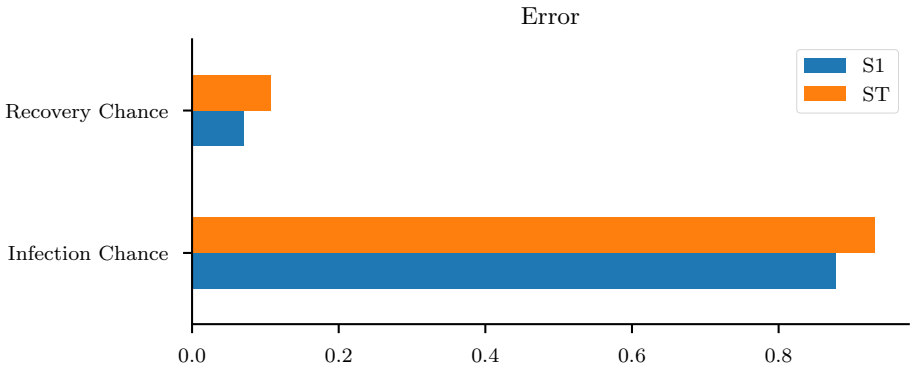

Figure 5.4: Sensitivity indices for the SIR model.

Another way of displaying the results of the sensitivity analysis consists of plotting the average values of the error metric across the parameter variations. This way, we can identify possible parameter values that could help minimize the final error of the simulation run. Figure 5.5 shows that the optimal parameter values for minimizing the final error are around 0.8 - 0.9 for infection chance and practically 0 for recovery chance. In other words, for the SIR model to be close to empirical results, the infectivity of divorce in the social circle of couples must be high and the impact of such separations must stay present for many years, due to an almost null recovery rate for infected agents.

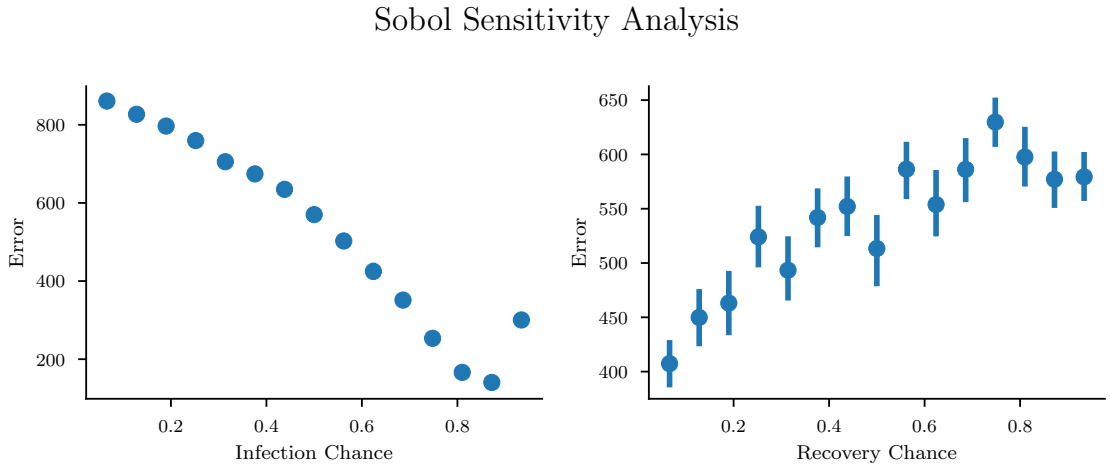

Figure 5.5: Sensitivity indices for the SIR model.

5.4.2 Results

The optimal values for the two parameters have been obtained using an implementation of the Global Best Particle Swarm stochastic optimization algorithm, using a total of 15 particles and 100 iterations. These configuration of the algorithm has been found to be effective after some testing with the available hyperparameters. Stochastic optimization methods such as particle swarms can be very useful when a high-dimensional space, such as all the possible simulation parameter combinations, has to be effectively explored with the aim of minimizing or maximizing a function.

Parameter	Infection	Recovery
Optimal Value	0.730	0.019

Table 5: Optimal parameter values for the SIR model.

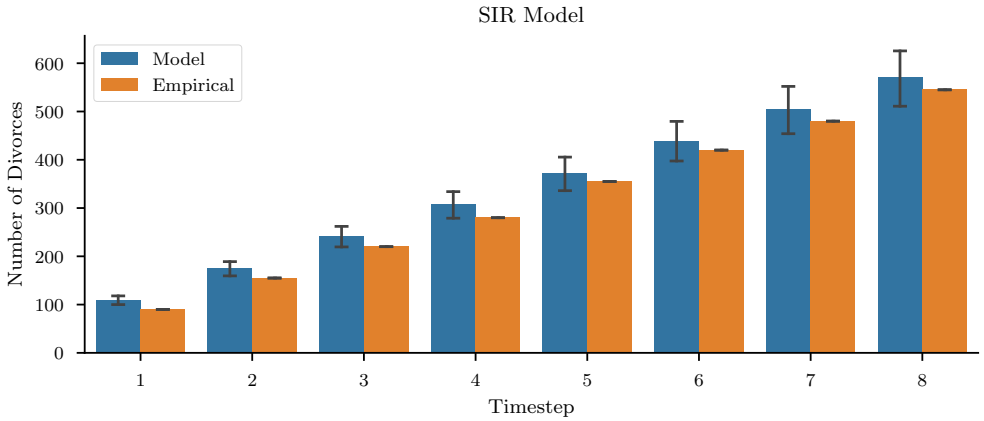

Figure 5.6: Divorce numbers for the SIR model.

Using the resulting optimal values for the parameters infection chance and recovery chance, the agent-based model has been simulated for 500 different runs, as the number of divorces per wave / timestep changes every simulation run due to the stochastic nature of some of the components of the agent-based model. The average of the obtained results is surprisingly close to the real empirical data of the Pairfam study (figure 5.6), if the simplicity of this epidemiological model and the very small number of parameters used are taken into account.

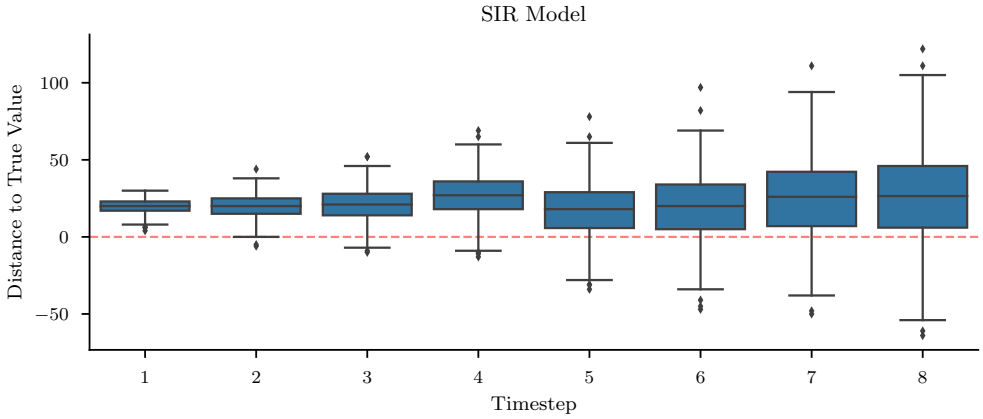

Figure 5.7: Distance to true value for the SIR model.

The difference with respect to the true empirical number of divorces in each wave can also be plotted as a boxplot graph, so that the real distribution of outcomes can be more clearly seen (figure 5.7). It looks like the SIR model tends to overestimate the number of divorces in every single wave, with this effect being more pronounced in the first few waves. The dispersion of results tends to grow significantly with each timestep, but this is an expected result of measuring the total number of divorces each wave as a cumulative sum. This means that the final result can differ from the true number of divorces in quite a significant way, as the effects of the stochastic elements of the model ripple through the timesteps.

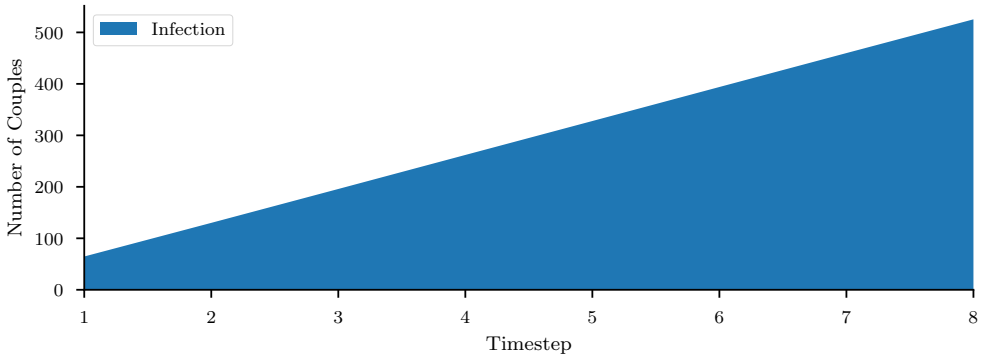

Figure 5.8: Importance of infection mechanisms.

As the only social mechanism present in the simple SIR model is infection by a neighboring couple, the graph showing the number of infections each wave shows a fairly constant rate and nothing more. When more social mechanisms are introduced and the models get more complex, the different sources of divorce will be plotted in a similar way.

5.5 Simple SIRa Model

The SIRa model can be considered a natural extension of the SIR model of the previous section. It includes a new mechanism of infection for the agents: the possibility of autoinfection or spontaneous infection without any contact with an infected agent. As divorces can obviously also happen due to the personal circumstances and characteristics of each couple, and without any input from their social circle, the inclusion of the autoinfection mechanism seems like a logical fit for the social contagion model being developed in this work.

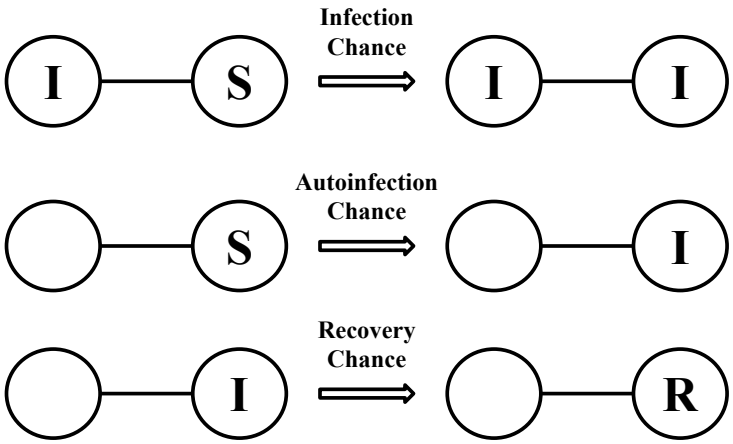

Figure 5.9: Mechanisms of infection and recovery in a SIRa model.

5.5.1 Sensitivity Analysis

In the SIRa model, it is clear that the impact that the parameter *autoinfection chance* has on the final error of the simulation runs is even more significant than the effect of the paremeter *infection chance* (figure 5.10). It also looks like *recovery chance* is not that important for final error results compared to the other two parameters being considered, as the impact its values have on the final error may not be as noticeable as with the other two parameters.

Figure 5.10: Sensitivity indices for the SIRa model.

The inclusion of the autoinfection mechanism has completely altered the optimal values of the different simulation parameters of the SIRa model, compared to those obtained for the SIR model. Now, the best values for both infection and autoinfection chance are quite low, so it seems that the addition of a second independent infection mechanism has lowered the need for higher social infectivity between agents. This might mean that the importance of social contagion between couples gets significantly reduced when the possibility of autoinfection is introduced. The recovery chance has higher optimal values in this case, thus reducing the length of the impact that divorce has in the social circle of the due to the addition of spontaneous infection.

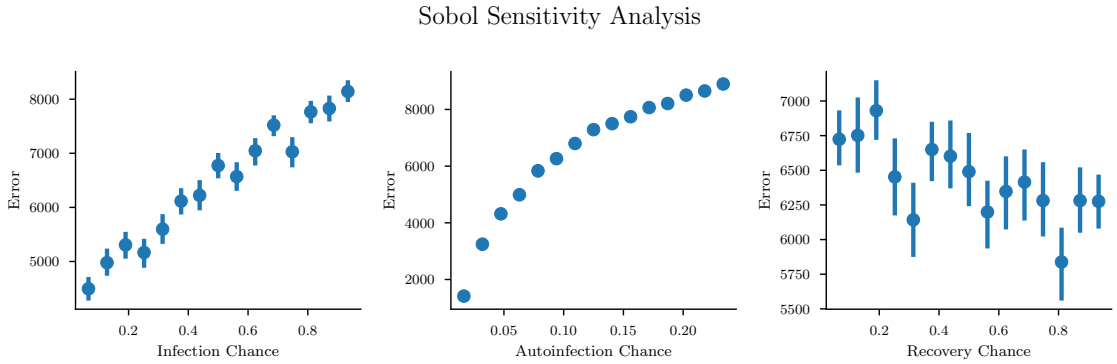

Figure 5.11: Sensitivity indices for the SIRa model.

5.5.2 Results

Once again, the Global Best Particle Swarm optimization has been used to get the optimal combination of the numerical values of the simulation parameters. 15 particles have been used for a total of 100 iterations, and the obtained results can be seen in table 6.

Parameter	Infection	Autoinfection	Recovery
Optimal Value	0.062	0.013	0.402

Table 6: Optimal parameter values for the SIRa model.

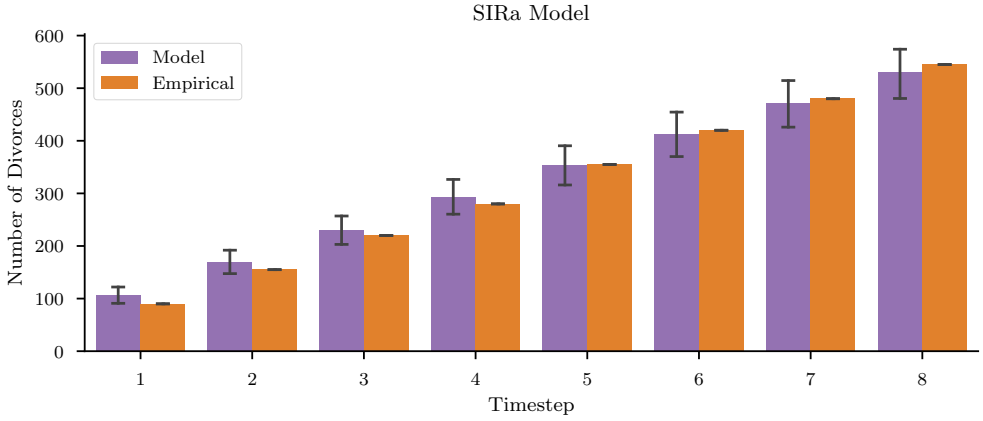

Figure 5.12: Divorce numbers for the SIRa model.

The results obtained in the SIRa model are also quite close to the true empirical values, even closer than those obtained with the SIR model (figure 5.12). It looks like this particular model has the tendency to overestimate divorces in the first few waves, but in contrast to the SIR model, it tries to correct this tendency and ends up underestimating the number of separations in the last waves. All in all, the average error value in each wave is lower than in the first SIR model, even though the dispersion shown in the outcomes is still quite significant.

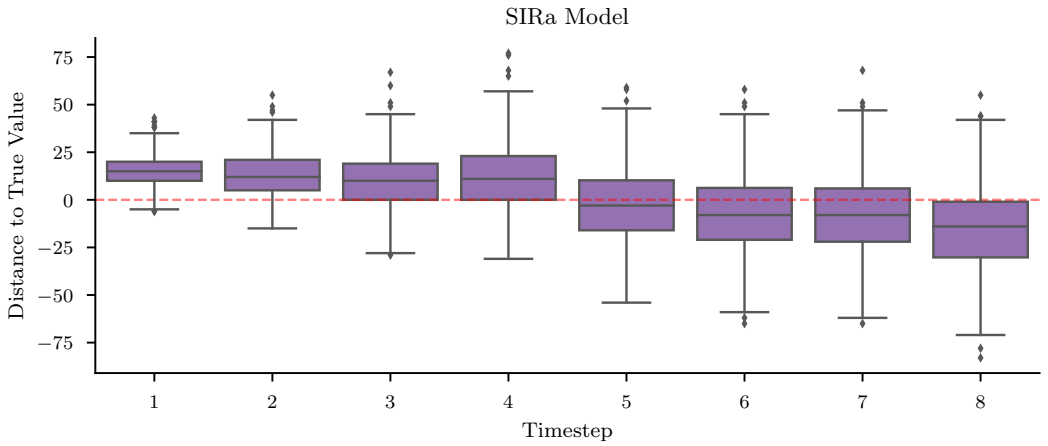

Figure 5.13: Distance to true value for the SIRa model.

Looking at the distribution of autoinfections and social contagion infections, it is clear that at least in the case of the SIRa model the best values for the different parameters prioritize the autoinfection mechanism over the social infection one. The numerical ratio is around 5.7:1 in favor of autoinfections, so in this model getting divorce due to the autoinfection mechanism is much more likely than getting socially infected by the divorce of another couple. This result could be expected, as separations are usually a very personal decision taken by the couple.

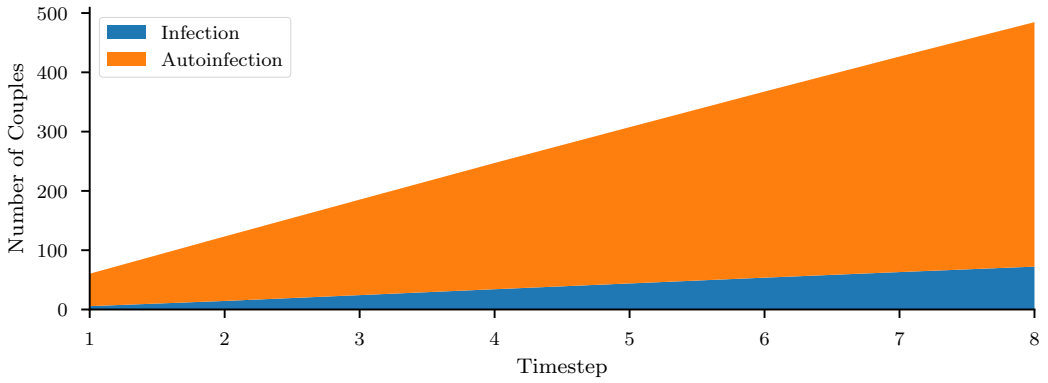

Figure 5.14: Importance of infection mechanisms.

5.6 Informed SIRa Model

The main contribution of this work is the introduction of a better version of the SIRa epidemiological model presented in the previous section, with the primary aim of making that type of infectious disease model better suited for the specific case of social contagion. The objective of this new model is to introduce a source of heterogeneity in the agents that form the agent-based model, because in the last two models every single agent was defined as having no personal characteristics at all. Using the CatBoost classification model implemented in the previous section, trained at a population level, each one of the couples that make up the agents of the model is given a probability of having relationship problems according to their real and changing personal characteristics in every one of the waves.

The probability of having relationship problems is then introduced in the model in two ways. That probability is scaled linearly to define the grade of susceptibility for infection each couple has when contacted by a divorced couple. This way, when a couple with a high probability of relationship instability is contacted by an infected agent, the probability of being infected and getting a divorce should be higher, and vice versa.

$$P_{Infection} = I_{Weight} \cdot P_{Problems} + I_{Base}$$

The probability of marriage instability also takes part in defining the rate of autoinfection in this model, in a very similar way to the infection mechanism explained above. This probability of autoinfection is also scaled linearly and higher probabilities of marital problems also translate to higher chances of autoinfection. Both linear equations (infection and autoinfection probabilities) have their own independent parameters (Weight and Base), to be optimized by the stochastic optimization algorithm.

$$P_{Autoinfection} = A_{Weight} \cdot P_{Problems} + A_{Base}$$

Thus, having more realistic and heterogeneous agents might help in getting better outcome results and, above all, in reducing the uncertainty and outcome dispersion that plagued the previous two simpler models. This way, having information about which couples are facing marriage instability should, in theory, help in getting more exact divorce numbers.

5.6.1 Sensitivity Analysis

The parameter with the most direct impact in error rates is clearly the base chance of autoinfection, with the weight of autoinfection and the respective parameters for the infection mechanisms also being important in higher-order interactions. That being said, it looks like the effect of the infection mechanism in the final error rate of the simulation is much less than that of autoinfection.

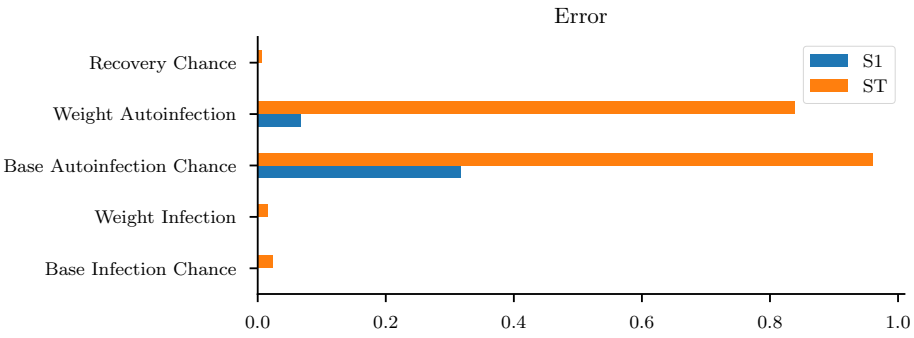

Figure 5.15: Sensitivity indices for the Informed SIRa model.

Looking at the average values of the error results across the parameter variations, the effects of even slightly modifying the optimal values of base chance of autoinfection and the weight of autoinfection can be clearly seen. Other parameters are more random in their impacts on error, but all values could be optimized simultaneously using an stochastic optimization technique like the particle swarms method used for the other models.

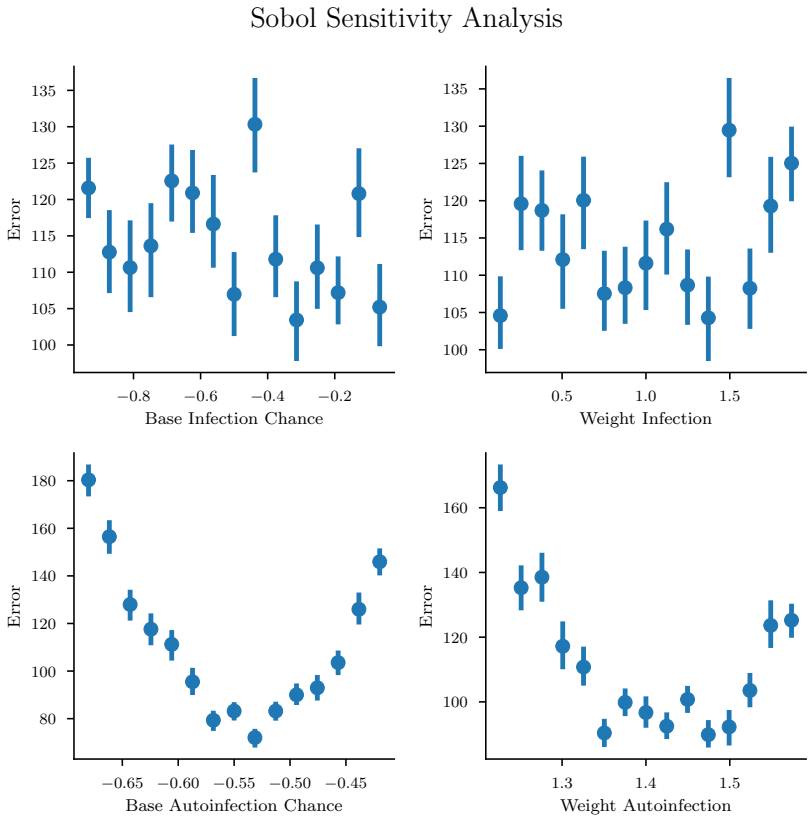

Figure 5.16: Sensitivity indices for the Informed SIRa model.

5.6.2 Results

Same as with all the other models before, Global Best Particle Swarm optimization has been used to obtain the final parameter values. 15 particles have been used for a total of 100 iterations, and the results can be seen in table 7.

Parameter	Optimal Value
Base Infection	-0.312
Weight Infection	1.465
Base Autoinfection	-0.484
Weight Autoinfection	1.304
Recovery	0.767

Table 7: Optimal parameter values for the Informed SIRa model.

One of the additions of this new model is the fact that it models the probabilities of infection and autoinfection as linear equations. Plotting this linear equations for all possible values of relationship problems might help in getting some deeper insights into the behavior of this informed SIRa model (figure 5.17). Looking at the lines, it looks like there's a clear threshold for autoinfection to happen at around 0.4, meaning that couples with probabilities of marital problems smaller than that value won't get spontaneously divorced. Looking at the other side of the graph, even if a couple has marriage trouble with absolute certainty, the probability of getting a divorce is around 0.8, making it likely but not certain. Focusing on the infection probability, its line is always above that of autoinfection, signifying that even with smaller probabilities of relationship problems, the likelihood of getting divorce is higher when the number of separations occurring in the social circle of the couple is high.

$$P_{Infection} = 1.465 \cdot P_{Problems} - 0.312$$

$$P_{Autoinfection} = 1.304 \cdot P_{Problems} - 0.484$$

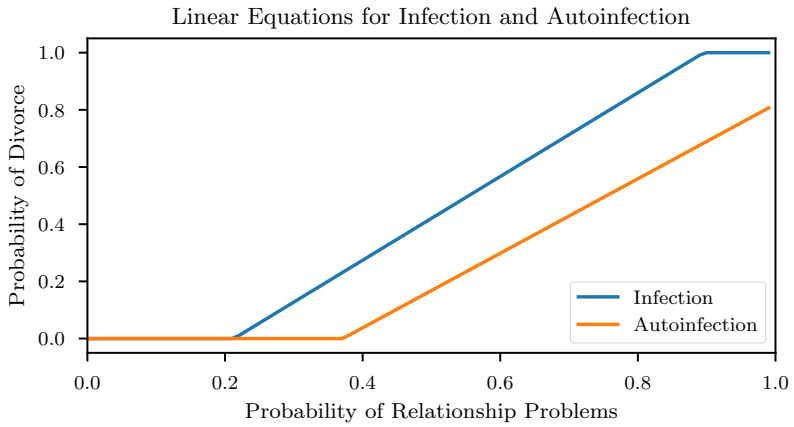

Figure 5.17: Linear equations for infection and autoinfection probabilities.

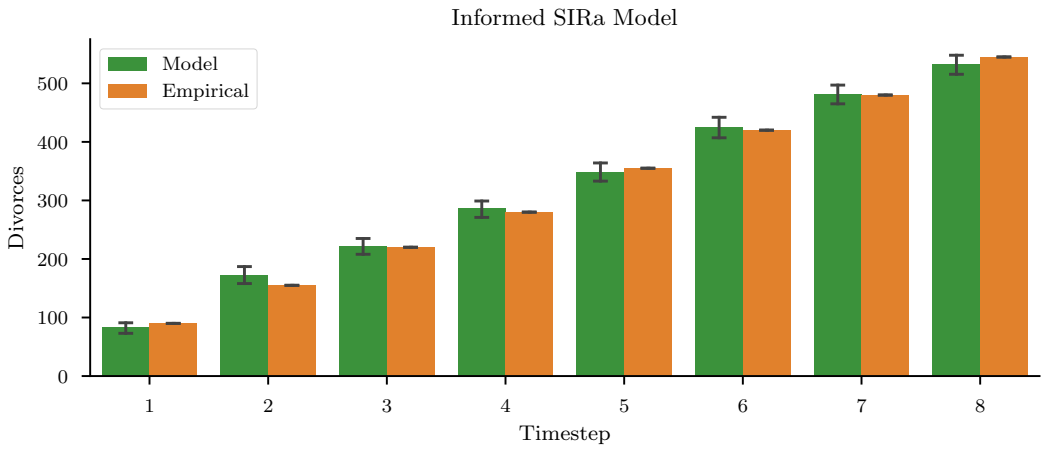

Figure 5.18: Divorce numbers for the Informed SIRa model.

Looking at the simulation results, it can be seen that the simulated number of divorces is also quite close to the empirical value (except a tendency to overestimate how many couples get divorced in Wave 2) (figure 5.19). Even then, the previous models also got a very good approximation of the real empirical values, if only the aggregated mean values of all the different simulation runs are taken into account.

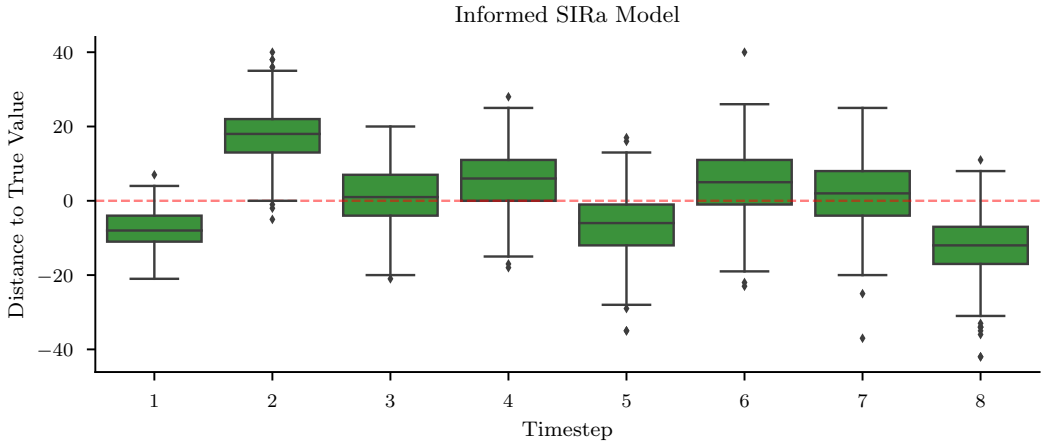

Figure 5.19: Distance to true value for the Informed SIRa model.

The most important characteristic of the informed SIRa model, then, is the significant reduction in outcome dispersion that can be better seen in figure 5.19. Thanks to the information about the probability of relationship instability that each of the couples face, introduced by the population level classification model, the range of values that the simulated number of divorces can take in each of the waves is greatly reduced, mainly in the last waves.

As can be seen in figure 5.20, the distribution of divorces between the infection and autoinfection mechanisms remains similar to the results obtain with the non-informed SIRa model, with a ratio of 4.66:1 in favour of autoinfection. This means that even though couple separations are a very personal choice, the impact of the social circle in these decisions is still significant.

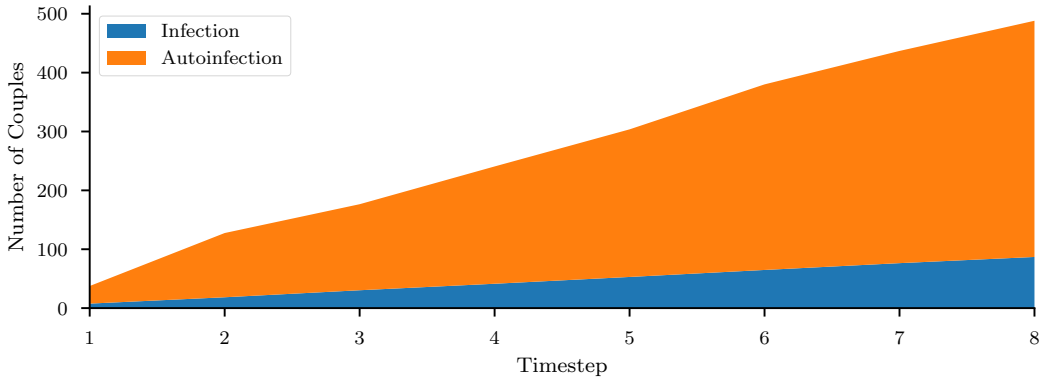

Figure 5.20: Importance of infection mechanisms.

5.7 Agent-Based Model Comparison

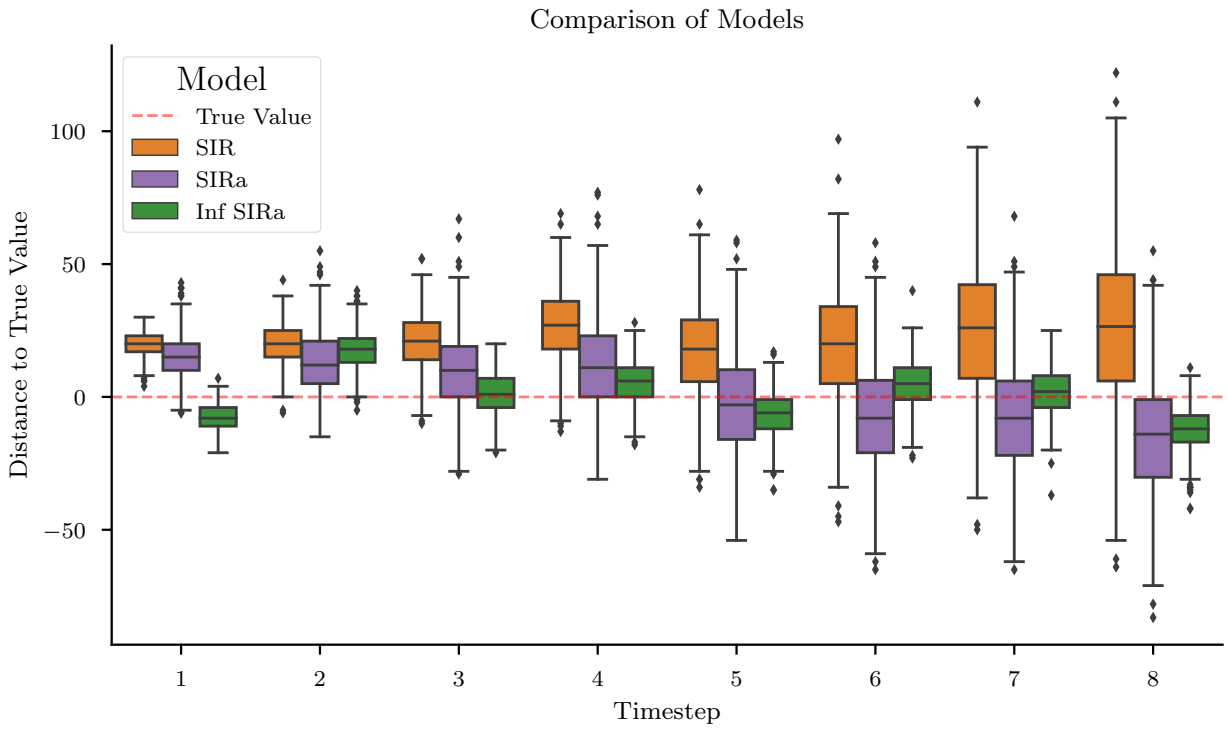

Figure 5.21: Distance to true value for all models.

The outcomes of the different models in each wave are better compared when they are all plotted in the same graph (figure 5.21). This way, the differences between the models can be better analyzed and the reduction in outcome dispersion that the informed SIRa model achieves can be appreciated in a much more clear way.

The average results of the informed model with the added heterogeneity are usually better than the other models, but the main advantage is the aforementioned reduced dispersion. This means that the impact of the uncertainty introduced by the stochastic elements of the agent-based model has

been somewhat mitigated by designing more realistic agents using ML models trained on realistic empirical domain data. The average distance to the true value has been plot as a line pot with confidence intervals in figure 5.22, and with the exceptions of wave 2 and 5 (where the non-informed SIRa model performs slightly better) the results of this new model are better than the other two models, with a much more controlled dispersion in outcomes.

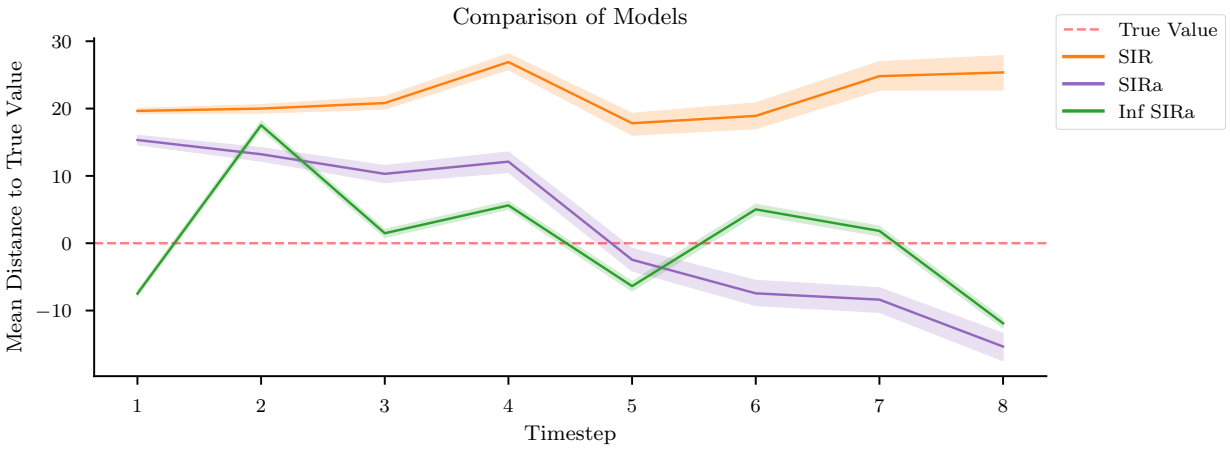

Figure 5.22: Average distance to true value for all models.

6 Conclusions

Even though divorce is such a complex social phenomenon, the results obtained in this work have shown that even the outcomes of simple epidemiological models can effectively approximate empirical values of separations in a social network context. Average results can be close to those encountered in reality, but this does not mean that other more intricate social mechanisms are not at play. In fact, going from the simple SIR model to the SIRa model completely changed the values of the *infection chance* parameter as the autoinfection mechanism that was not present before took a clearly central role in the new model. Thus, more complex models might include more advanced social mechanisms and the value of the parameters may most certainly change in order to accommodate the new distribution of importance between the different infection mechanisms.

Instead of relying entirely on chance and probability, and treating every couple as effectively the same type of agent without any identifying characteristics, a way to introduce significant heterogeneity into the agent population has been presented in this work. The probability of relationship instability of every single couple, computed taking into account personal features of the couple like personality traits of both members, their satisfaction with the relationship and their ways of dealing with intramarriage conflicts, has been modeled using a gradient boosting ensemble called CatBoost. This model works as a population-level model, because it has been trained with data from all available marriages in the Pairfam study and is not tailored for each specific agent. Having such a model to predict probabilities of marriage troubles allows computing them even before the simulation run starts, as the characteristics of each couple are known beforehand for every simulated wave.

The introduction of this new kind of agents into the SIRa model, along with a stochastically optimized linear model of infection and autoinfection probabilities, helped reduce the average error rate in the simulation runs with a really significant drop in outcome dispersion. These results show how the introduction of more complex and realistic agents can help mitigate the effects of compounding uncertainty in agent-based models that deal with social contagion.

Moreover, the use of machine learning techniques to create the new agent rules proved a fruitful endeavor. Without this approach, researchers from outside the field of sociology could not have approximated an agent rule-set as efficient as this. Not having manually defined rules can have some limitation, though. Machine learning models tend to behave in a black box manner, and the importance of using model explainability techniques like feature permutations is crucial when trying to understand which variables have the most impact in a human decision-making process.

Even though better results have been obtained, the limitation of this work are clear. Agents in the model only have a single numerical characteristic: the probability of the couple facing relationship instability. Adding other numerical parameters that help differentiate the agents even more is a logical next step. Moreover, the relationship between the probability of divorce and the probability of relationship instability was modeled using a linear equation, when the true relationship might have a non-linear (exponential, logarithmic, quadratic...) nature. Further investigation into the use of different types of models is warranted.

References

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, 2002.

[2] Joe Alexander, Jr, Roger A Edwards, Luigi Manca, Roberto Grugni, Gianluca Bonfanti, Birol Emir, Ed Whalen, Steve Watt, Marina Brodsky, and Bruce Parsons. Integrating machine learning with microsimulation to classify hypothetical, novel patients for predicting pregabalin treatment response based on observational and randomized data in patients with painful diabetic peripheral neuropathy. *Pragmat. Obs. Res.*, 10:67–76, 2019.

[3] Ethem Alpaydin. *Machine Learning, revised and updated edition*. MIT Press, London, England, 2021.

[4] Paul R Amato, Alan Booth, David R Johnson, and Stacy J Rogers. *Alone together: How marriage in America is changing*. Harvard University Press, London, England, 2009.

[5] Jane Anderson. The impact of family structure on the health of children: Effects of divorce. *Linacre Q.*, 81(4):378–387, 2014.

[6] Roy M Anderson and Robert M May. *Infectious diseases of humans: Dynamics and control*. Oxford University Press, London, England, 1992.

[7] Oliver Arránz Becker. Effects of similarity of life goals, values, and personality on relationship satisfaction and stability: Findings from a two-wave panel study. *Pers. Relatsh.*, 20(3):443–461, 2013.

[8] Ellen-Wien Augustijn, Shaheen A Abdulkareem, Mohammed Hikmat Sadiq, and Ali A Al-babawat. Machine learning to derive complex behaviour in agent-based modellzing. In *2020 International Conference on Computer Science and Software Engineering (CSASE)*. IEEE, 2020.

[9] Sigal G Barsade. The ripple effect: Emotional contagion and its influence on group behavior. *Adm. Sci. Q.*, 47(4):644–675, 2002.

[10] Peter S Bearman and James Moody. Suicide and friendships among american adolescents. *Am. J. Public Health*, 94(1):89–95, 2004.

[11] Gary S Becker. *Treatise on the family*. Harvard University Press, London, England, 1991.

[12] Hans-Peter Blossfeld and Rolf Müller. Guest editors' introduction: Union disruption in comparative perspective: The role of assortative partner choice and careers of couples. *Int. J. Sociol.*, 32(4):3–35, 2002.

[13] Riccardo Boero. *Behavioral Computational Social Science: Boero/behavioral computational social science*. John Wiley & Sons, Nashville, TN, 2015.

[14] Joyce E Bono and Remus Ilies. Charisma, positive emotions and mood contagion. *Leadersh. Q.*, 17(4):317–334, 2006.

[15] Raymond Boudon. Social mechanisms without black boxes. In Peter Hedstrom and Richard Swedberg, editors, *Social Mechanisms*, pages 172–203. Cambridge University Press, Cambridge, 1998.

[16] Matthew D Bramlett and William D Mosher. Cohabitation, marriage, divorce, and remarriage in the united states. *Vital Health Stat. 23*, (22):1–93, 2002.

[17] Paula Branco, Luis Torgo, and Rita Ribeiro. A survey of predictive modelling under imbalanced distributions. 2015. arXiv:1505.01658.

[18] Dale K Brearcliffe and Andrew Crooks. Creating intelligent agents: Combining agent-based modeling with machine learning. In *Proceedings of the 2020 Conference of The Computational Social Science Society of the Americas*, pages 31–58. Springer International Publishing, Cham, 2021.

[19] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[20] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[21] Jonathan Brogaard, Joseph Engelberg, and Christopher A Parsons. Networks and productivity: Causal evidence from editor rotations. *J. Financ. Econ.*, 111(1):251–270, 2014.

[22] L L Bumpass, T C Martin, and J A Sweet. The impact of family background and early marital factors on marital disruption. *J. Fam. Issues*, 12(1):22–42, 1991.

[23] Antoni Calvó-Armengol and Matthew O Jackson. The effects of social networks on employment and inequality. *Am. Econ. Rev.*, 94(3):426–454, 2004.

[24] Damon Centola. Failure in complex social networks. *J. Math. Sociol.*, 33(1):64–68, 2008.

[25] Damon Centola. The spread of behavior in an online social network experiment. *Science*, 329(5996):1194–1197, 2010.

[26] Damon Centola and Michael Macy. Complex contagions and the weakness of long ties. *Am. J. Sociol.*, 113(3):702–734, 2007.

[27] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *N. Engl. J. Med.*, 357(4):370–379, 2007.

[28] Nicholas A Christakis and James H Fowler. The collective dynamics of smoking in a large social network. *N. Engl. J. Med.*, 358(21):2249–2258, 2008.

[29] Claudio Cioffi-Revilla. *Introduction to computational social science: Principles and applications*. Springer International Publishing, Cham, Switzerland, 2018.

[30] Danielle Cohen. It's a great summer to break up. https://www.thecut.com/article/celebrities-break-up-summer-2023.html, August 2023. Accessed: 2023-8-28.

[31] Lauren Cohen, Andrea Frazzini, and Christopher Malloy. The small world of investing: Board connections and mutual fund returns. *J. Polit. Econ.*, 116(5):951–979, 2008.

[32] James Coleman, Elihu Katz, and Herbert Menzel. The diffusion of an innovation among physicians. In *Social Networks*, pages 107–124. Elsevier, 1977.

[33] James S Coleman. Social theory, social research, and a theory of action. *Am. J. Sociol.*, 91(6):1309–1335, 1986.

[34] Antonio Criminisi. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Found. Trends® Comput. Graph. Vis.*, 7(2-3):81–227, 2011.

[35] Erik Cuevas. An agent-based model to evaluate the COVID-19 transmission risks in facilities. *Comput. Biol. Med.*, 121(103827):103827, 2020.

[36] D J Daley and D G Kendall. Epidemics and rumours. *Nature*, 204(4963):1118, 1964.

[37] Theodore Eugene Day, Nathan Ravi, Hong Xian, and Ann Brugh. An agent-based modeling template for a cohort of veterans with diabetic retinopathy. *PLoS One*, 8(6):e66812, 2013.

[38] P S Dodds and D J Watts. A generalized model of social and biological contagion. *J. Theor. Biol.*, 232(4):587–604, 2005.

[39] Scott W Duxbury and Dana L Haynie. Criminal network security: An agent-based approach to evaluating network resilience. *Criminology*, 57(2):314–342, 2019.

[40] Du Feng, Roseann Giarrusso, Vern L Bengtson, and Nancy Frye. Intergenerational transmission of marital quality and marital instability. *J. Marriage Fam.*, 61(2):451, 1999.

[41] Alberto Fernandez, Salvador Garcia, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*. Springer Nature, Cham, Switzerland, 2019.

[42] C Ferri, J Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognit. Lett.*, 30(1):27–38, 2009.

[43] Joël Foramitti. AgentPy: A package for agent-based modeling in python. *J. Open Source Softw.*, 6(62):3065, 2021.

[44] John Foster. Why is economics not a complex systems science? *J. Econ. Issues*, 40(4):1069–1091, 2006.

[45] James H Fowler and Nicholas A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *BMJ*, 337(dec04 2):a2338, 2008.

[46] James H Fowler and Nicholas A Christakis. Cooperative behavior cascades in human social networks. *Proc. Natl. Acad. Sci. U. S. A.*, 107(12):5334–5338, 2010.

[47] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[48] Jerome H Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, 2002.

[49] Feng Fu, Xiaojie Chen, Lianghuan Liu, and Long Wang. Social dilemmas in an online social network: The structure and evolution of cooperation. *Phys. Lett. A*, 371(1-2):58–64, 2007.

[50] Wyndol Furman and Duane Buhrmester. The network of relationships inventory: Behavioral systems version. *Int. J. Behav. Dev.*, 33(5):470–478, 2009.

[51] Aman Garg, Samson Yuen, Nuttiiya Seekhao, Grace Yu, Jeannie A C Karwowski, Michael Powell, Jon T Sakata, Luc Mongeau, Joseph JaJa, and Nicole Y K Li-Jessen. Towards a physiological scale of vocal fold agent-based models of surgical injury and repair: Sensitivity analysis, calibration and verification. *Appl. Sci. (Basel)*, 9(15):2974, 2019.

[52] R Gentleman and V J Carey. Unsupervised machine learning. In *Bioconductor Case Studies*, pages 137–157. Springer New York, New York, NY, 2008.

[53] Nigel Gilbert and Pietro Terna. How to build and use agent-based models in social science. *Mind Soc.*, 1(1):57–72, 2000.

[54] Nigel Gilbert and Klaus G Troitzsch. *Simulation for the social scientist*. Open University Press, Buckingham, England, 2 edition, 2005.

[55] Erving Goffman. *Encounters: Two studies in the sociology of interaction*. Literary Licensing, 2011.

[56] Frances K Goldscheider and Linda J Waite. *New families, no families?: The transformation of the American home*. University of California Press, Berkeley, CA, 1993.

[57] Mark Granovetter and Roland Soong. Threshold models of diffusion and collective behavior. *J. Math. Sociol.*, 9(3):165–179, 1983.

[58] Volker Grimm, Gary Polhill, and Julia Touza. Documenting social simulation models: The ODD protocol as a standard. In *Understanding Complex Systems*, pages 349–365. Springer International Publishing, Cham, 2017.

[59] Karsten Hank and Michaela Kreyenfeld. A multilevel analysis of child care and women's fertility decisions in western germany. *J. Marriage Fam.*, 65(3):584–596, 2003.

[60] Rom Harre, David Clarke, and Nicola De Carlo. *Motives and mechanisms: An introduction to the psychology of action*. Routledge, London, England, 2017.

[61] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: Data mining, inference, and prediction, second edition*. Springer, New York, NY, 2 edition, 2017.

[62] Elaine Hatfield, John T Cacioppo, and Richard L Rapson. *Studies in emotion and social interaction: Emotional contagion*. Cambridge University Press, Cambridge, England, 1993.

[63] Peter Hedström and Richard Swedberg. Social mechanisms: An introductory essay. In Peter Hedstrom and Richard Swedberg, editors, *Social Mechanisms*, pages 1–31. Cambridge University Press, Cambridge, 1998.

[64] Peter Hedström and Petri Ylikoski. Causal mechanisms in the social sciences. *Annu. Rev. Sociol.*, 36(1):49–67, 2010.

[65] Nathan O Hodas and Kristina Lerman. The simple rules of social contagion. *Sci. Rep.*, 4(1):4343, 2014.

[66] John H Holland. Genetic algorithms. *Sci. Am.*, 267(1):66–72, 1992.

[67] John H Holland. Complex adaptive systems and spontaneous emergence. In *Contributions to Economics*, pages 25–34. Physica-Verlag HD, Heidelberg, 2002.

[68] M J Howes, J E Hokanson, and D A Loewenstein. Induction of depressive affect after prolonged exposure to a mildly depressed individual. *J. Pers. Soc. Psychol.*, 49(4):1110–1113, 1985.

[69] Johannes Huinink, Josef Brüderl, Bernhard Nauck, Sabine Walper, Laura Castiglioni, and Michael Feldhaus. Panel analysis of intimate relationships and family dynamics (pairfam): Conceptual framework and design. *JFamRes*, 23(1):77–101, 2011.

[70] Iacopo Iacopini, Giovanni Petri, Alain Barrat, and Vito Latora. Simplicial models of social contagion. *Nat. Commun.*, 10(1):2485, 2019.

[71] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An introduction to statistical learning: With applications in python*. Springer, New York, NY, 2023 edition, 2023.

[72] Thomas E Joiner and Jennifer Katz. Contagion of depressive symptoms and mood: Meta-analytic review and explanations from cognitive, behavioral, and interpersonal viewpoints. *Clin. Psychol. (New York)*, 6(2):149–164, 1999.

[73] Elihu Katz, Paul F Lazarsfeld, and Elmo Roper. *Personal influence: The part played by people in the flow of mass communications*. Routledge, London, England, 2017.

[74] Matt J Keeling and Ken T D Eames. Networks and epidemic models. *J. R. Soc. Interface*, 2(4):295–307, 2005.

[75] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2003. ACM.

[76] Dr Rafiqul Zaman Khan and Haider Allamy. Training algorithms for supervised machine learning: Comparative study. *Int. J. Manag. Inf. Technol.*, 4(3):354–360, 2013.

[77] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.*, 5(4):221–232, 2016.

[78] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. Springer, New York, NY, 2019.

[79] Reed W Larson and David M Almeida. Emotional transmission in the daily lives of families: A new paradigm for studying family process. *J. Marriage Fam.*, 61(1):5, 1999.

[80] Robert K Merton. *Social Theory and Social Structure*. Macmillan, London, England, 1968 edition, 1968.

[81] Thomas Mitchell. *Machine Learning*. McGraw-Hill Professional, New York, NY, 1997.

[82] M E J Newman. Spread of epidemic disease on networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 66(1 Pt 2):016128, 2002.

[83] Esteban Ortiz-Ospina and Max Roser. Marriages and divorces. *Our World in Data*, 2020.

[84] R Pastor-Satorras and A Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86(14):3200–3203, 2001.

[85] Ilya Prigogine and Isabelle Stengers. *Order out of chaos: Man's new dialogue with nature*. Verso Books, London, England, 2017.

[86] S Douglas Pugh, Joerg Dietz, Jack W Wiley, and Scott M Brooks. Driving service effectiveness through employee-customer linkages. *Acad. Manag. Perspect.*, 16(4):73–84, 2002.

[87] Beatrice Rammstedt and Oliver P John. Big five inventory. In *Encyclopedia of Personality and Individual Differences*, pages 469–471. Springer International Publishing, Cham, 2020.

[88] M Richards, R Hardy, and M Wadsworth. The effects of divorce and separation on mental health in a national UK birth cohort. *Psychol. Med.*, 27(5):1121–1128, 1997.

[89] Kat Rock, Sam Brand, Jo Moir, and Matt J Keeling. Dynamics of infectious diseases. *Rep. Prog. Phys.*, 77(2):026602, 2014.

[90] James B Rule, Peter Hedstrom, and Richard Swedberg. Social mechanisms: An analytical approach to social theory. *Soc. Forces*, 77(4):1697, 1999.

[91] Richard M Ryan, Edward L Deci, Maarten Vansteenkiste, and Bart Soenens. Building a science of motivated persons: Self-determination theory's empirical approach to human experience and the regulation of behavior. *Motiv. Sci.*, 7(2):97–110, 2021.

[92] Bruce Sacerdote. Peer effects in education: How might they work, how big are they and how much do we know thus far? In *Handbook of the Economics of Education*, pages 249–277. Elsevier, 2011.

[93] Laura Sanchez and Constance T Gager. Hard living, perceived entitlement to a great marriage, and marital dissolution. *J. Marriage Fam.*, 62(3):708–722, 2000.

[94] Liana C Sayer and Suzanne M Bianchi. Women's economic independence and the probability of divorce: A review and reexamination. *J. Fam. Issues*, 21(7):906–943, 2000.

[95] Jaime S Sichman, Rosaria Conte, and Nigel Gilbert, editors. *Multi-Agent Systems and Agent-Based Simulation*. Springer, New York, NY, 2014.

[96] Adam Slowik and Halina Kwasnicka. Evolutionary algorithms and their applications to engineering problems. *Neural Comput. Appl.*, 32(16):12363–12379, 2020.

[97] Kirsten P Smith and Nicholas A Christakis. Social networks and health. *Annu. Rev. Sociol.*, 34(1):405–429, 2008.

[98] Scott J South and Kim M Lloyd. Spousal alternatives and marital dissolution. *Am. Sociol. Rev.*, 60(1):21, 1995.

[99] Forrest Stonedahl and Uri Wilensky. Finding forms of flocking: Evolutionary search in ABM parameter-spaces. In *Lecture Notes in Computer Science*, pages 61–75. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[100] Martin Summer. Financial contagion and network analysis. *Annu. Rev. Fin. Econ.*, 5(1):277–297, 2013.

[101] Yanmin Sun, Andrew K C Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *Intern. J. Pattern Recognit. Artif. Intell.*, 23(04):687–719, 2009.

[102] Richard S Sutton. *Reinforcement Learning*. Springer My Copy UK, 1992.

[103] Johan Ugander, Lars Backstrom, Cameron Marlow, and Jon Kleinberg. Structural diversity in social contagion. *Proc. Natl. Acad. Sci. U. S. A.*, 109(16):5962–5966, 2012.

[104] Axel van den Berg. Is sociological theory too grand for social mechanisms? In Peter Hedstrom and Richard Swedberg, editors, *Social Mechanisms*, pages 204–237. Cambridge University Press, Cambridge, 1998.

[105] Gerben A van Kleef, Carsten K W De Dreu, and Antony S R Manstead. The interpersonal effects of anger and happiness in negotiations. *J. Pers. Soc. Psychol.*, 86(1):57–76, 2004.

[106] Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Inf. Fusion*, 76:89–106, 2021.

[107] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Michal Walczak, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Trans. Knowl. Data Eng.*, pages 1–1, 2021.

[108] L Weng, A Flammini, A Vespignani, and F Menczer. Competition among memes in a world with limited attention. *Sci. Rep.*, 2(1):335, 2012.

[109] Jinghui Zhong, Xiaomin Hu, Jun Zhang, and Min Gu. Comparison of performance between different selection strategies on simple genetic algorithms. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. IEEE, 2006.