Miniproject 2

Dante Godolja (11929150)
Lukas Burtscher (11925939)

Introduction: The Rising Trend of Machine Learning

Artificial intelligence (AI) and machine learning (ML) have been making headlines in recent years. As these technologies continue to advance, they're transforming industries and occupations in significant ways. One community particularly impacted by this shift is the developer community. Developers adept in AI and ML techniques, referred to as ML developers, are increasingly influential. This effect is intensified with the advent of transformative tools such as OpenAI's ChatGPT.

In a fascinating turn of events, we're witnessing an imaginative scenario where ML developers are inspiring their peers, primarily web developers, to explore the burgeoning field of Machine Learning. This shift from web development to ML is the primary focus of our study. We model this transformation as a contagion process and perform a simulation to understand the dynamics of this transition. For both cases (simple and complex contagion) we will consider a starting number of 9739 (out of 37700) “infected nodes” since these developers identified as Machine Learning developers. Thus we stay true to the dataset and to Miniproject 1.

Simple Contagion: Modeling the Shift to ML

The shift towards ML within the developer community can be likened to a contagion process, where the "infection" in this case is the interest and knowledge of ML. Under this model, ML developers are the "infected" nodes, and web developers are the "susceptible" nodes.

This contagion process is governed by the rules of the Susceptible-Infected-Recovered (SIR) model:

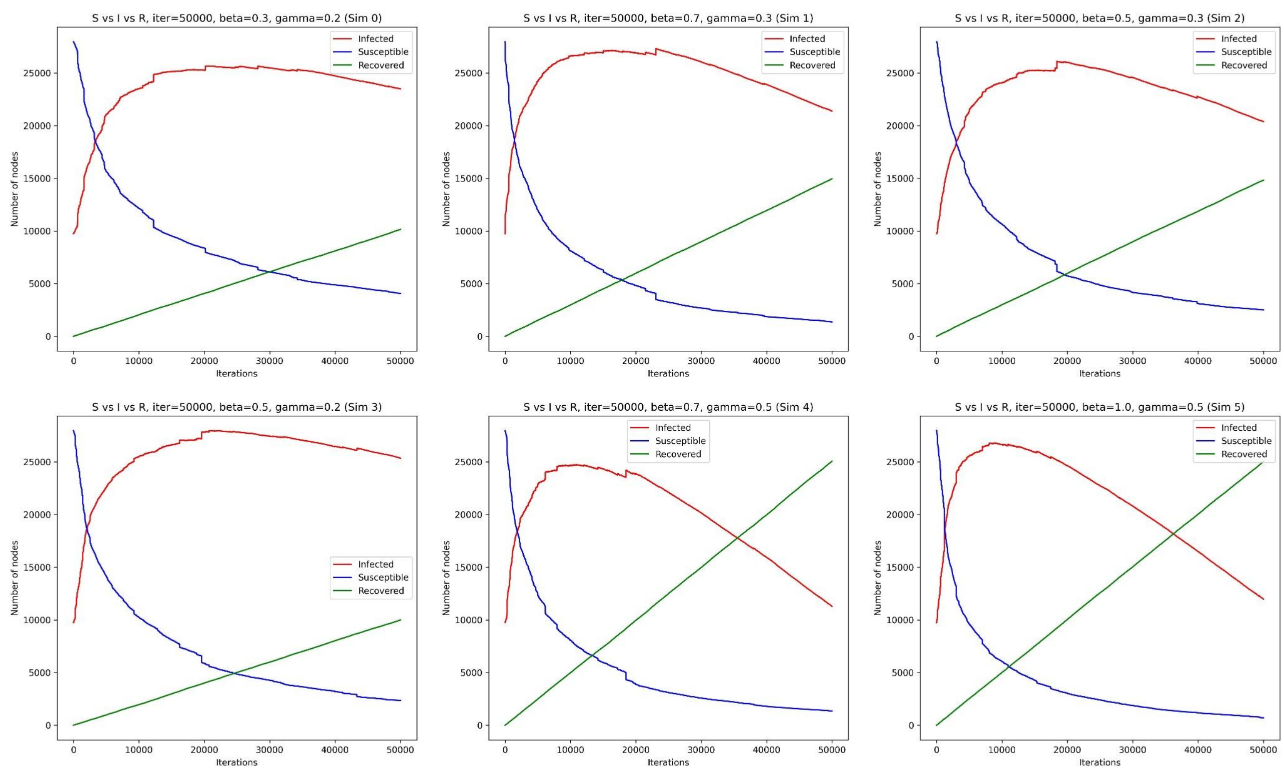
1. **Selection of Infected Node:** The process begins by randomly selecting an infected node, an ML developer, from the network. This node is also marked as ‘processed’ which is necessary for the calculation of the reproduction rate R which is defined as the average number of infections a secondary infected node has managed to infect.
2. **Identifying Susceptible Nodes:** We then identify the susceptible neighbors of the infected node. These are the web developers who are susceptible to the influence of ML developers.
3. **Infection Process:** Each susceptible node has a chance (defined by the parameter beta) of being "infected" or convinced to shift their interest towards ML.
4. **Recovery Process:** The infected node then has a probability (defined by the parameter gamma) of recovery, indicating that it has exhausted its influence.

Simulation Parameters and Limitations

Running a complete simulation for such a shift can be computationally demanding due to the extensive size of developer communities. To mitigate this, we have capped each run of the simulation to 50,000 iterations. Despite this constraint, the simulation captures enough dynamics to provide meaningful insights into the contagion process.

Simulation Scenarios and Outcomes

We simulated six scenarios using different combinations of the beta (infection rate) and gamma (recovery rate) parameters: (0.3, 0.2), (0.5, 0.2), (0.5, 0.3), (0.7, 0.3), (0.7, 0.5), and (1.0, 0.5). We could interpret the betas in our case as the ability of the ML developers to convince web devs to switch and the gammas as the chance that an ML dev gets bored of his field and tries something else. For example a beta of 0.3 means that an ML dev has a 30% chance to convince a web dev friend of his to make the switch and a gamma of 0.2 means that there is a 20% chance that the ML dev gets bored of ML and switches. Among these, the simulation with beta 0.5 and gamma 0.2 resulted in the highest number of infected nodes, implying that more web developers transitioned to machine learning under these conditions. Moreover, this scenario also showed the 2nd highest rate of reproduction, indicating an effective spread of the contagion. Another notable run is simulation 4 which shows the lowest number of infected nodes and incidentally the lowest reproduction rate.



Simulation	Reproduction rate R0
Simulation 0	1.1555352241537054

Simulation 1	1.1305287896592244
Simulation 2	1.0763075722092115
Simulation 3	1.1533204121187393
Simulation 4	1.0041311754684838
Simulation 5	1.0025930572982016

Influential and Vulnerable Nodes

Our network analysis revealed nodes with the highest degree of centrality—nodes with the most connections—as the most influential and vulnerable. The majority of these influential nodes were originally web developers, which points to their vulnerability to the shift towards machine learning. In contrast, nodes with degree centrality around the median did not manage to infect a lot of other nodes. This could be because such nodes have a small amount of connections or by pure chance because of the low betas. The figure below shows the most influential nodes from all simulations and their original alignment (web-developers).

id	name	ml_target
31890	dalinhuang99	0
19222	Bunlong	0
13638	gabrielpconceicao	0
27803	nfultz	0
35008	mdo	0
36652	rfthusn	0
35773	addyosmani	0
3712	isaacs	0
30002	kauegimenes	0
36628	SuriyaaKudoIsc	0
73	kentcdodds	0
9051	getify	0
10001	ronenhamias	0

Conclusion: Insights and Implications

Through the lens of a simple contagion model, our study provides intriguing insights into the shifting landscape of developer communities. The simulations suggest that highly-connected web developers play a significant role in spreading the 'infection'—the interest in machine learning since they themselves might be connected to originally ML developers.

Complex Conatgion

The Scenario

To understand complex contagions, let's continue our imaginative scenario. Machine Learning (ML) has become a pivotal aspect of software development, and ML developers are persuading their peers, who are primarily Web Developers, to switch their expertise. This shift is not as straightforward as a simple contagion like in our previous model. It's instead driven by the intensity of influence a web developer experiences from their circle of friends.

Understanding Complex Contagion

In a complex contagion model, an individual doesn't switch (or, in our case, doesn't become an ML developer) just by coming into contact with a single 'infected' individual. Instead, the likelihood of a node (Web Developer) getting 'infected' (becoming an ML developer) is tied to the proportion of its neighbors who are already infected. This proportion is denoted by a threshold. If, say, the threshold is 0.1, a web developer only switches to become an ML developer if more than 10% of their connected friends are already ML developers. They are 'convinced' by the collective influence of their peer group.

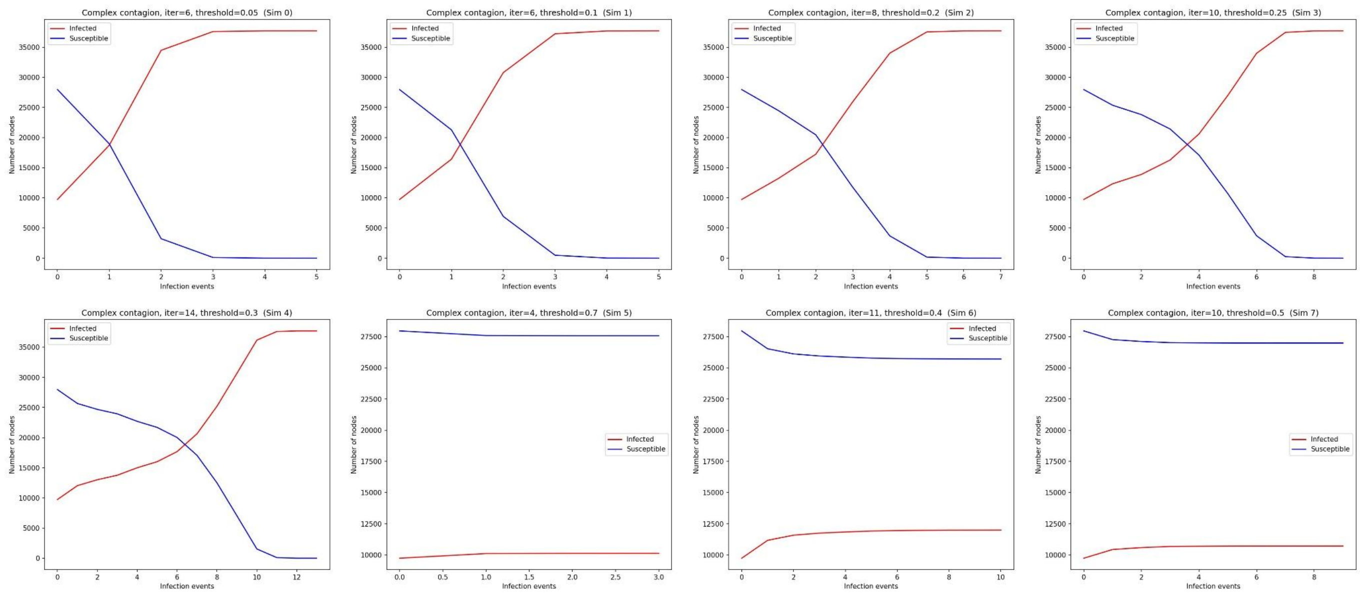
The Simulation Process

In our complex contagion simulation, we iterate through each susceptible (web developer) node and calculate the ratio of its neighbors who are infected (ML developers). If this ratio is greater than a predefined threshold, we change the node's status to infected (become an ML developer). The iteration proceeds until no more nodes can be infected, or the maximum number of iterations (defined as infection events) is reached. For this scenario we decided to process all susceptible nodes at once and not per time-step, in order to make the simulation run faster and terminate when no more nodes can be infected and not introduce an actual maximum number of iterations.

It's crucial to note that these iterations do not signify actual time passing. Instead, each iteration is an event of possible infection, a moment of decision for each web developer node based on the current status of their network neighbors.

Simulation Findings

We ran the simulation on eight different thresholds: 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5, and 0.7. For thresholds of 0.05 to 0.3, we observed a generalized infection - practically all web developers were convinced to become ML developers. As the threshold increased to 0.4, 0.5, and 0.7, fewer nodes were infected, meaning fewer web developers decided to transition.



At thresholds leading to generalized infections (0.05 to 0.3), the most influential nodes, in other words, the nodes that infected the most neighbors, were those with the highest degree of centrality. In our scenario, these were originally web developers with many connections, which initially (before getting infected) are also the most vulnerable. As they transitioned, their influence on their network significantly accelerated the contagion.

For the cases where the threshold was higher, and not all nodes got infected, the initially infected nodes (original ML developers) were the most influential. Despite their influence, the higher threshold prevented them from convincing the majority of their web developer friends.
In these cases (0.4, 0.5, 0.7) we observe that the number of infected converges to ~12500, ~11000 and ~10000 respectively.

The least influential nodes ended up being those who are connected to developers that don't have a ratio of ML friends higher than the threshold. These nodes represent ML developers who are connected to Web devs who are yet to be convinced about the transition to ML development.

In conclusion, this complex contagion simulation gives us a more nuanced view of influence dynamics. Our findings underscore the role of network structure and influential nodes in shaping the adoption of new ideas or practices within the network. This analysis can be valuable for planning strategic interventions or understanding resistance to change in similar social contexts.