CS112 LBA

Jacob Puthipiroj

Hello Vivian, Katka, as per our meeting on Tuesday, I'd like to go into further detail about the matching algorithm I proposed.

Executive Summary

Instead of relying solely on human judgment for the matching of startups to investors on Investors' Day, Betahaus can use a simple matching algorithm to compute the level of 'fit' of each startup to each investor, and order each startup by said fitness score. This a simple algorithm for which data can easily be implemented in currently existing customer-facing interfaces, such as EventBrite signup sheets. Having an algorithmically sorted list for each startup will significantly reduce the amount of human hours to be spent on this task.

Problem Framing

Betahaus, as a community of entrepreneurs, often organizes multiple events year-round in order to grow the startup ecosystem in Berlin. One of the largest of these events is the Investors' Day, held annually in November. In recent years, the number of attendees has grown exponentially; the attendance in 2018 is guaranteed to include at least 300 startup representatives and 30 investors. With it being impossible for every investor to spend time individually with each startup, each investor aims to spend time only with startups that are most relevant to their own interests. To date, this has been done manually - Betahaus staff will spend hours finding attendees in the startup list which they believe have the closest matches to the investor, based on their own understanding of investor preferences. However, this process can be significantly automated by way of a matching algorithm.

Data Required

The matching algorithm requires relatively small amounts of data that can be easily (and is currently) gathered in the EventBrite signup pages. This includes, from the startups:

- Field of interest (The question is posed as a 'check all that apply' decision)
- Necessity of investors (A binary yes/no variable)
- Startup stage (An categorical variable of Idea/Prototype/Product Launched)
- Funding round (An ordinal variable of Pre-seed/Seed/Series A and beyond)
- Location of the startup (A City name that is looked up for its coordinates on Google Maps)

Furthermore, the following preferences are required from the investors:

- Field of interest (The question is posed as a 'check all that apply')
- Startup stage preference ('Check all that apply' for Idea/Prototype/Product Launched)
- Funding Round preference ('Check all that apply' for Pre-Seed/Seed/Series A and beyond)
- Location preferences (If there are preference, input City name and the radius for which to consider startups, if none, leave blank).

Methodology

1. Start with the first investor.
2. Filter startups on 'Startup stage' and 'Funding round' variables.
    - For example, if the investor does not want to consider startups in the idea stage or in the Pre-Seed stage, then temporarily remove them from the list of startups
3. Create a matrix of matches:
    - The columns indicate the investors' interests, while rows indicate startups' field of interests.
    - For example, for an investor interested in 'SaaS', 'Enterprise', and 'Developer Tools', and with Startup A interested in 'SaaS', 'Enterprise' and 'Smart Mobility', and with Startup B interested in 'Smart Mobility', 'FashionTech' and 'SaaS', the matrix is

Investor Fields of Interest	SaaS	Enterprise	Developer Tools
Startup A	1	1	0
Startup B	1	0	0

    - 1 indicates a shared interest in a field, while 0 indicates no shared interest.
    - By looking at the matrix, we can see that Startup A is at least as relevant to the investor than Startup B is, because it shares all of Startup B's interests, plus one more. All of the startups' other interests are ignored.
4. Scale the matrix.
    - This is done by dividing each cell in the matrix by the *standard deviation* for that column. However, if the standard deviation for that column is 0, then set the entire column to 0.
    - The reasoning behind this is that the rarity of each interest should be taken into account. In the above example, the fact that both Startups are interested in SaaS makes the column relatively unimportant. Similarly, the fact that neither is interested in Developer Tools makes that column unimportant. The only important column here, is actually 'Enterprise', since there is a mix of both 1s and 0s. Standard deviation takes this rarity of the columns into account.
    - In the above example, the matrix obtained would be:

Investor Fields of Interest	Saas	Entreprise	Developer Tools
Startup A	0	0.7071	0
Startup B	0	0	0

5. Add up the sums of each column to obtain the final score, and sort by said score
    - In the example, this would be 0.7071 for Startup A, and 0 for Startup B.
    - Hence, Startup A should be ranked higher than Startup B.
6. Repeat steps 2-5 for each new investor.

Conclusion and Next Steps

With an exponentially growing number of attendees to Betahaus events, it is quickly becoming infeasible to rely solely on manpower to ensure satisfaction of investors when it comes to presenting portfolio startups. By using an algorithm that sorts startups by relevancy, Betahaus staff can more effectively choose matches for each investor.

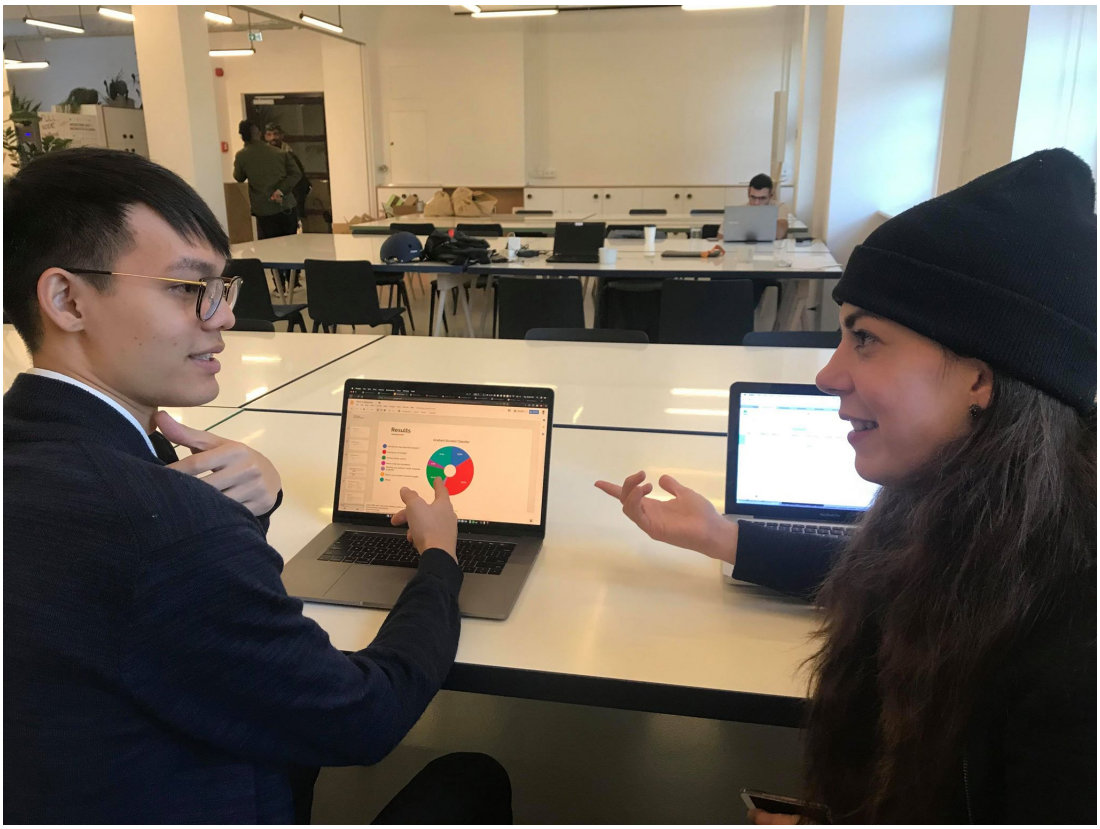

Figure 1: A meeting with Katka on November 6$^{\text{th}}$, 2018.