Q1. Could you describe your process for developing an ML system?

It is a really huge discussion. First of all, the data set that you usually have in the industry is no comparison to the standard datasets, which are basically preprocessed and prearranged with correct sampling, so completely different. So usually, the raw data needs a lot of cleaning and identifying data quality issues. This is the first step, understanding your data. Then you also have some talks with domain experts that know the data better. And you define the business requirements of your project. Then the goal of the project is clear. And then the prototyping starts with Jupyter Notebooks for Data Scientists, I think it's very friendly, very easy, you can get results very easy like visualizations. Also I like to use Tableau a lot lately in order to get some fast analytics. The process is not really standardized because it also depends on the problem. So if you want to model a specific feature then you would start exploring and engineering this feature. It's very important that you understand what your data means. And then gradually proceed with feature engineering and as soon as you have all your features then you move to starting modeling some models. Especially for the data analysis, there is no standard process. And then after modelling you calculate some metrics in case it's a supervised problem and then see "ok, I reached the threshold for the accuracy that I wanted" and then you move on with the deployment. Of course accuracy is not enough, you also have other metrics like precision, recall. If all satisfy some threshold then you would move to production. But if it's unsupervised it's harder. You need a lot of feedback from people who are evaluating because you don't have a ground truth. I didn't have experienced this amazing thing of going to production with Jupyter Notebooks like Netflix does. So all the time it is scripts that you transfer like after the prototyping is done and you know what needs to be done and everything is working. Then you take the components that are needed, transformed to a script. Then you basically go with python scripts and usually you are generating the model which is stored in some artifactory like jfrog. And then you build a pipeline out of it.

Q2. Do you continue experiments after deploying the ML system? If so, how do you go about doing this?

"If some of the requirements change, like if you have some new additional requirements that your client would like to explore, then you would have to go back to the prototyping phase and start developing from scratch. So, you start a new prototype but incorporate the knowledge and code snippets from the first POC. Then you explore some new ideas on how to deal with the problem, maybe a new approach, a new algorithm or so on. This also happens when your data contains new information, like some new meta data. I also had a case where we worked on a POC but suddenly they came and tell us that it's not the correct data set. We got data from a completely different source, so different preprocessing and different way of identifying the specific fields and so on. We adapted our original POC to the new situation. Actually, you need to document everything and a lot of times it is also really hard to maintain, like I think this is a challenge because you need to maintain your prototypes which are in Jupyter now and at the same time you have the same thing translated into scripts and deployed to a server into production. And at the same time, like you are experimenting with the new prototype, you make changes and then you still have to adapt the scripts. This is a challenging part. And people somehow make it like you keep track of what happens, but it requires a lot of effort. Like a lot of capacity, a lot of resources. From one side you develop something new, but you cannot have like an automatic way of translating this directly to scripts. Maybe you also redesign and change the architecture by introducing something in the prototype. This affects the whole architecture of the system which is in production. This is like 2 paces basically. You have one pace which is

experimenting and developing new things and the other has to somehow keep up with your pace of experimentation. You have to maintain two different things. What I noticed in like bigger companies: usually those decisions are not taken very fast like there is a moment where the client will ask you to work on something new. Until this is deployed into production, there is some time like it's not happening from one day to another, so I think this is what makes it a little bit smoother. Because you don't have a continuous flow, that new thing. deploying, new thing, deploying. Usually there is some space between those two phases and so this is what helps.

Q3. Which tools do you use?

already answered in Q1 [Interviewer asks if plugins are used] No, can you recommend some?

Q4. What are some common challenges you face in this process form a tooling perspective?

I think debugging is. This is what I really miss in Jupyter Notebooks because it is sequential, and you basically have to run from the beginning to the end. I think there are some work arounds in Jupyter so that you can put some breakpoints but it’s not that user friendly I would say.

Q5. Which issues do you encounter when working with Jupyter Notebooks?

Reusability of code is an issue in a Jupyter Notebook. Also testing is really hard to do and parallel experiments as well. Scalability is also very important. Experiment tracking is also very important but you can find work arounds but I haven’t come up with a workaround for testing.