1:41

Alexey

**This week, we'll talk about MLOps. We'll talk about pragmatic and standardized MLOps. Actually, we just finished our MLOps course. For the students who just graduated, this will be a very interesting and relevant video, in addition to what we learned and we covered in the course. We will see how to actually implement some of the things we talked about. We have a very special guest today, Maria. Maria is a machine learning engineer. She is bridging the gap between data scientists, infrastructure and IT teams at different brands. She is focused on the standardization of machine learning Ops. It's a pleasure to have you here. Welcome to the show!**

2:30

Maria

Thank you. I'm very happy to be here. And I love your course, by the way. It's amazing. I never fully did it, but I looked through it. I recommend it to everyone. I'm planning to do it too, at some point.

2:43

Alexey

**Well, I'm not sure how much you will learn. But hopefully, maybe you will learn something. Please let me know how it goes.**

2:49

Maria

I think it's always interesting to see how others do the courses. That's what I like about it.

2:58

Alexey

**Thanks. The questions for today's interview were prepared by Johanna Bayer, as always. Thanks, Johanna, for your help. Now let's start.**

# Maria's background

3:07

Alexey

**Before we go into the main topic of our interview today, which is MLOps, let's start with your background. Can you tell us about your career journey so far?**

3:17

Maria

Yes, it's been a while already. I'm almost 14 years in data – in the AI field. I started as a data analyst and I studied economics and econometrics. So doing something in data was logical, I guess. Data science was not there yet. There were some data analyst jobs and I got one. I did a lot of things in R – built some models (churn and acquisition) models for a telecom company, and did some automation from some cron jobs running on some server standing in the room. It was funny. [chuckles]

3:54

Alexey

**It was an actual physical machine – a physical computer?**

3:59

Maria

Yeah, in the room, with a key. Yes.

4:03

Alexey

**So you needed to go to that room, open a screen, and then you would use a USB stick with your program?**

4:13

Maria

No, no, no, there was a shared drive. We could access and, indeed, schedule things. It was fun. I really liked automation already back then. And then I moved to another team. My position, formerly, was data scientist. I learned Python back then and fully switched to Python. I did some models for fraud detection for mobile subscriptions and was also busy with NLP. The main project I was working on was model factory, which is basically an MLOps project. It was seven years ago – MLOps was not a thing – but we already standardized the process of machine model deployment across different departments in that company.

This was at KPM, a telecom company in the Netherlands. That was really thinking very forward back then – no one was busy doing this. We had a governance on top of that with Teradata. Yeah, it was interesting. We rebuilt that system multiple times because the tool stack was changing over time. The last version of it was using Kubernetes and ELK stack monitoring, and we had Bitbucket and Jenkins – we had orchestration. We also did this iteration with AWS-native tools, with Sagemaker, step functions, and all of that.

5:45

Alexey

**Do you remember what the first stack was? What did it look like?**

5:48

Maria

Yeah, so KPM bought Aster. I don't know whether anyone knows what Aster is.

5:57

Alexey

**No.**

5:59

Maria

Good. It's horrible. [chuckles] But it was a product from Teradata. It was distributed computing – something kind of like Spark, but worse. You could do machine learning in SQL – you could do random forests in SQL and stuff like that. It had...

6:17

Alexey

**It doesn't sound terrible, though.**

6:20

Maria

Well, it didn't work very well. It wasn't very user-friendly. So we built some repositories that basically triggered execution on those servers and we used Bitbucket. We had Teradata and some... I'm not sure what kind of dashboard it was back then... we had some Python dashboards with the history about the runs, so you could do interactive search across multiple model runs. All metadata about the models (about the experimentation) we were storing in the Teradata database. If you look at Mlflow – what MLflow does now is pretty much what we did, but the backend was different. And we had our own wrappers, like “store metadata”, “store RC results”, “store experimentation results” – I don't remember the function names anymore. It's probably still somewhere on GitHub. [chuckles] It is funny. Now I'm kind of doing the same MLOps framework, but the tools are different. In every company the tools are different and I think you have to go with what you've got because fighting for getting new tools is not always a good idea. It will take you a long time and I'm not sure you will gain anything from it. Here we have GitHub, GitHub Actions – we use Databricks for pretty much everything, and we have Kubernetes. Well, I think I have quite a good experience building with this tool stack.

8:03

Alexey

**Interesting. So you started doing MLOps before it was a thing, when you were a data scientist. And now, your title was ML Engineer, right?**

8:14

Maria

Well, I guess I'm a MLOps Tech Lead. I don't have a title per se. My official title is Manager of Machine Learning Engineering or something, but I guess I combined three different roles in myself. It's more like a product manager, also advertising what we do and talking to different departments across the organization – and trying to promote what we do, get budgets, and also developing and creating the roadmap.

8:46

Alexey

**That sounds like a lot. Do you still get to do hands-on stuff?**

8:53

Maria

Not that much as I used to before, but I still try to do some things myself.

9:02

Alexey

**I think I was doing something quite similar to what you're doing right now. And I did not have time to do hands-on stuff at all.**

9:10

Maria

In the evenings, yeah.

9:11

Alexey

**In the evenings. [chuckles] Work-related, or not?**

9:16

Maria

Well, I guess now I spend a lot of evenings on Marvelous MLOps. Everything we write about is very much work-related, because we write a lot about specific tool stacks we also use, and the best practices we implement in our organization. I also code work-related in the evenings and the weekends, occasionally. [chuckles] I like it too much, I guess.

# Marvelous MLOps

9:45

Alexey

**What's this Marvelous MLOps that you mentioned?**

9:48

Maria

Marvelous MLOps is a blog that we started in April of this year, together with my colleague, Başak. We work together at Ahold Delhaize at this moment and we want to share our knowledge with the world. We started with a Medium blog and we created the company page on LinkedIn. Later, Rafael joined us and the three of us now work on creating the articles. We post an article every week, and we create content three times per week. We have a meme, and we have a Post-it or “cheat sheet,” something like that. We also have a newsletter. I really like doing that. It's really fun. I also now post every day on LinkedIn. I really like it. I enjoy doing that.

10:41

Alexey

**Yeah, I think I come across your posts quite often. When I do I like them, of course, so you can see the likes. If somebody wants to find out more about that, they should follow you, right?**

10:57

Maria

Yeah, of course. You should follow Marvelous MLOps. We do post a lot of useful information about how to become better at MLOps and how to think pragmatically about it.

# Maria's definition of MLOps

11:10

Alexey

**Which is the topic of today's interview, right? Maybe if we take a step back – I'm curious to know your definition of MLOps. What is MLOps in your opinion?**

11:24

Maria

MLOps is a set of practices that allows data scientists to bring models to production in an efficient way. I really see MLOps as an enablement tool, where you don't have another team that helps you to deploy things, but the data scientists need to be given the freedom to do it themselves. But the goal of the MLOps team is not just to enable but also to teach data scientists how to use it in the correct way. Because if you just give a tool and don't explain how it works and how it should be used, it goes wrong. That's how I see it.

12:03

Alexey

**It's almost the same definition that I use in the course, I think. For me, I think I used something like, “It's a set of practices and tools to bring data science machine learning to production. I don't remember the exact definition. But I think it's almost the same one. You mentioned the MLOps team – I'm wondering, what kind of setup do you have in mind? Is there a central MLOps team that helps other teams with the deployment of their machine learning models?**

12:42

Maria

Yeah, so how I see the MLOps team – it is a team that provides MLOps infrastructure. It deploys the tools that are being used for MLOps. They also provide things like maybe reusable CI/CD pipelines, the authentication mechanisms – it's all done in a standardized way so that the product teams don't have to figure it out on their own. Because pretty much everything that is done within the organization is kind of everyone repeating themselves. So to avoid that, most of this workload should be on a central team. But you still need data scientists, you still need machine learning engineers. Machine learning engineers, who will be working on optimizing the code that the data scientists write and working really closely with them. For example, if there are certain requirements from the latency point of view, it is the machine learning engineers that are helping with it. But MLOps is really more like an infrastructure team, I would say, rather than a machine learning engineering team. But it's very related.

13:48

Alexey

**So would you say that there are teams (we can call them feature teams or product teams) that work on specific parts of the product. They have data analysts, data scientists, ML engineers – and then there is a central MLOps team that provides the infrastructure that enables other teams to deploy machine learning to production. The central team also teaches the ML engineers and data scientists from other teams how to use this set of tools (this set of practices) to be able to achieve what they want. Right?**

14:23

Maria

Yeah, that's how I see it. And monitoring is also one of the important things that the central team also needs to provide. I think it works well in this setup. I don't know whether other people have different experiences, but from our experience, it works quite well.

# Alternate team setups without a central MLOps team

14:45

Alexey

**I'm also curious to know if there is any other kind of setup where there is no centralized team for MLOps and the teams are kind of still doing this. Because in my case, it was a very similar setup – there was a centralized MLOps team. Then there were other teams who would follow the best practices and use the tools provided by the team. Interesting. Okay. Today we wanted to talk about “pragmatic MLOps”. I'm curious to know why this is even a thing. Is there a non-pragmatic MLOps? And what's the difference between non-pragmatic and pragmatic MLOps?**

15:32

Maria

Yeah. I think everyone knows about this landscape called the MAD (Machine Learning, Artificial Intelligence & Data) Landscape, which grows every year. If you compare the MAD Landscape from what it was two years ago to what it is now – it's just blown up. It's crazy. Basically, it's a map that shows what kind of tools there are for different parts of MLOps – what there is for machine learning model deployments, for feature stores, for monitoring, you name it. It's literally all aspects of AI. And it just grows.

16:10

Alexey

**This is the picture with a lot of logos, right? [Maria agrees] Recently, it has become tiny logos, where you really need to zoom in to see any individual ones. It's just a huge set of logos, right?**

16:22

Maria

Yes, exactly.

16:23

Alexey

**That's the thing you're referring to? I think I saw this picture on the internet.**

16:27

Maria

It's madness. It's really madness. And it's not helping anyone. I think it creates imposter syndrome, where people think they don't know anything and that there are so many tools. But the fact is that – buying any tool won't really solve your problems. Typically, the main problem that I see is an organizational problem. The teams are organized in the wrong way and the tools are already there, actually. If you look at the tools within any large organization, you already have Kubernetes, you already have something for orchestration, something for version control, something for CI/CD pipelines – pretty much something for everything you need for MLOps.

And it doesn't make any sense to buy any new tools, because it will be, first of all, hard to convince people why you need that in the first place. But secondly, if you even buy it, you will spend possibly years on integrating it with the rest of the organization. It may work very well, I believe, for small companies. There are many tools that claim to do end-to-end – and I guess, if you don't have anything (if you're a startup, for example) then maybe, for the time being, it's a good idea to have something like that. But in a large organization, it's not gonna help. That's what I really believe in.

# Pragmatic vs non-pragmatic MLOps

17:55

Alexey

**So that's the pragmatic part. The non-pragmatic part is trying to look at this landscape and have this fear of missing out like, “Oh, I need all these tools!”**

18:06

Maria

Yeah, and “I need to buy it all”.

18:08

Alexey

**Yeah, buy, learn, integrate, spend five years trying to do all that, instead of using the tools that are already there and focusing on thinking, “How can we use these tools that we already have to deploy our projects?” Right?**

18:26

Maria

Yeah. Because tools are changing all the time. By the time that you integrate it, everything's already outdated because every two years, as there is in this new cycle, I believe, in tooling.

# Must-have ML tools (categories)

18:41

Alexey

**So what is a “must-have” set of tools that we need to have? What are the categories? I imagine maybe there are some categories that we can introduce later – but at the beginning, we just need just the basic ones that cover 80% of cases.**

18:56

Maria

Yeah, we have this article on the Marvelous MLOps Substack. It's a featured article at this moment. It covers really minimal setup. So you really need version control, you really need the CI/CD pipeline (CI/CD tool like Jenkins or GitHub Actions or GitLab pipelines), you need to have something for Docker registry, you need to have something for model registry, you need to have something for deploying the models (it's probably Kubernetes or maybe some tools like AzureML or Databricks because they provide managed services for deploying things – that may work as well). Feature stores, I wouldn't consider as minimal because you could, I guess, have some workarounds. And monitoring, monitoring is crucial. You must have monitoring. That's also a part of the absolute minimum. I guess I mentioned them all. Maybe I forgot something but I think that's it. Yeah.

20:03

Alexey

**Well, I imagine any software company already has permission control systems like Git, GitLab, or GitHub or whatever. Then CI/CD, too. And then probably some sort of Docker registry. It could be from GitLab, or Amazon – whatever. And then there is probably a way to deploy things – maybe there is Kubernetes, or some other container orchestration platform. But when it comes to model registry and monitoring, maybe there is nothing – or maybe there is something, but... [cross-talk]**

20:39

Maria

At KPM, where we worked before we had, we basically used Artifactory from JFrog.

20:47

Alexey

**What is Artifactory?**

20:49

Maria

Artifactory, it's like a package registry or any object registry, really. You can have pipelines hosted there even. You can also upload any files there and you can assign attributes to those files so they become searchable. They also have nice Python integration. You can just store files in S3 buckets. You can also assign attributes to them and search through them. You have to build something around it but, I mean, it wouldn't be a total nightmare, right? If you don't have MLFlow specifically. I love MLFlow, though. But for minimal setup, something like that would also be okay if you don't really don't have anything – you don't have a team to support anything like that, then it's also going to be okay. The idea is just that things are traceable and reproducible.

That's the main key idea. We also have this list of MLOps Maturity Assessment. I think I had a post about it two days ago, I will also put it in “featured” on my LinkedIn so that people can see. That's an Excel sheet with 60 questions that goes through all aspects of MLOps – or at least main aspects that we see as more important. That's how you can see how mature you are in what you do. I think that's a great start – to see what things are structurally missing, so that you can act on them in a strategic way.

# Maturity assessment

22:23

Alexey

**You said there are 60 questions in this spreadsheet, right? Do you remember what kind of... Well, maybe we will not go into all 60, but maybe you remember what kind of categories of questions there are?**

22:36

Maria

Yeah, of course. There is a documentation piece, which I find very important. Documentation is always missing, but we really need to pay attention to documentation. Then there are aspects regarding reproducibility. Reproducibility means that for every round of the machine learning project, you need to be able to find what code was responsible for the run, what computer was responsible for the run, what model was responsible for certain deployment, where it was stored – all of that. And it's important that you can always roll back easily. Because if you don't have this in place, then the rolling back process will be very tedious. You really don't want to go there [chuckles].

Then there's also code quality, which is an important piece of this assessment. You need to ensure that if things are changed, there are pull requests created, and that there are multiple people looking at your pull request. Then there are also things like the merge is blocked unless you have certain tests to be run. There is also a test coverage, maybe, assessment on how well your testing is covered – integration testing all that stuff. Those are important pieces that I guess are mentioned there. There are many more – feature stores, monitoring, as well. These are some examples.

# What to start with in MLOps

24:01

Alexey

**Well, I Imagine that if you work at a startup and let's say you already рave your first model or maybe their first multiple models – at least in my experience, you often deploy them in “you only live once” mode, like “Okay, let's just deploy. Throw them out there and keep our fingers crossed that nothing will break.” But then at some point, the startup (this organization) becomes more mature and they actually realize that they need these things.**

**In which order should they introduce this? Should they start with documentation? Should they start with reproducibility? Should they start with code quality? Should we start with feature stores? Because there are 60 questions, right? As a startup, you cannot just stop everything you're doing and say “Let's cover all the 60 questions and then two years later, we'll come back and continue working.”**

24:57

Maria

No. Well, documentation is, of course, important, but I guess it's less critical than having proper version control and code quality guidelines and traceability and reproducibility. So if that is covered, at least that will save you from a lot of headaches later. Other things are, of course, also important, but that is really crucial. Because if you don't have that, it's gonna be a big mess.

25:26

Alexey

**All these things, (at least, this is how it sounded) are more about processes, right? So it's like, “Okay, how do you deploy in such a way that we have traceability and reproducibility and we can easily roll back?” So it's about having these guidelines and teaching people like machine learning engineers from other teams, “If this happens, what do you do? If there is a bug, how do you roll back?”**

26:00

Maria

Yeah, but it depends on the tools, right? This assessment is a very high level one – it doesn't go into tools. But what we're doing right now in our blog, we have some articles on traceability and reproducibility, and how we use it with Databricks, specifically. But you can think about a similar setup for Kubernetes deployments. You have to really look at the tools you have to implement that. But for us, I think it's important to say this definition of “done,” which is not really present. That's my feeling for data science projects. If the model is just deployed, then it's good. We don't really look at all those things that must be there before you can consider that something is in production. I've seen scheduled notebooks in Databricks without any version control and they claim to be in production. I wouldn't call it in production – almost 0% of the whole list is covered, right?

27:06

Alexey

**So if we come back to our discussion about pragmatic MLOps, maybe we can somehow summarize what exactly is pragmatic and how we can be pragmatic about MLOps?**

27:20

Maria

I think it depends whether you're in a large organization, or maybe in a startup. From my experience, (I only worked in large corporate companies, so this is coming from corporate companies perspective) you should not look into buying tools in the first place, but more into looking what tools you already have and how you can use them so that you score high in this MLOps Maturity-based Assessment, as an example. Also, how do you structure the team so that data scientists can actually deploy themselves with proper guardrails in place, but not blocking data scientists from doing the deployments?

What I've often seen is that there are data science teams that are totally separate from the IT department, and then the IT department has their own DevOps engineers and cloud engineers and other engineers and data scientists just have no permission to do anything. It's just a “throwing over the wall” kind of situation happening. So you really want to avoid that and see how you construct your team in a way that is sustainable for everyone. I think those two aspects – the teams in the organization and the choice of the tooling are the main components, I guess, in pragmatic MLOps.

# Standardized MLOps

28:42

Alexey

**So you probably already have all the tools you need – so start with those. Then think about the structure of your teams, how exactly they're organized, “Are you helping data scientists or you're just blocking them and you're annoying them?” I remember, when I was a data scientist, I needed to deploy something but then somebody from the Ops team needed to do something and they're, of course, almost always busy. So I'm just sitting there waiting, “Okay, should I go check out YouTube and watch cat videos? What should I do now?” [chuckles]. Okay, so that's pragmatic MLOps. I think we've covered that a little bit.**

**We also wanted to talk about standardized MLOps. From what I understood from our discussion, we already talked about the maturity assessment questions and some things we covered there like, “Okay, how do you go about reproducibility? How do you go about rolling back?” and so on. Is this something that is related to standardization?**

29:55

Maria

Yeah, I guess so. I think the choice of the tooling is related to standardization. We work in a large corporate organization with 19 brands all over the world – a lot in Europe: Greece, Serbia, Romania, Czech Republic – we also have brands in the US. But pretty much everyone has the same tool stack with slight variations. We basically bring it to the same structure everywhere and we provide reusable CI/CD pipelines for everyone – that everyone can use. Those CI/CD pipelines go beyond just CI/CD pipelines. We introduced a cookie cutter repository. Data scientists typically hate cookie cutter templates, because it's not always clear how to use them, so we made it simpler. It's just a simple action in the repository, where people can trigger, and your name of your repository should follow some conventions, otherwise, it wouldn't be able to deploy. It checks whether you have the permissions to create this repository because you need to belong to a certain GitHub group to be able to deploy it – create a certain repository, certain naming convention. Then, that person triggers the workflow and it also applies.

We are on Azure, so we use service principles – it applies service principles, credentials, which are organizational secrets, to those repositories. This means that after this cookie cutter template runs, the data scientist has a working repository with main.pi that can already be deployed in Databricks, which is tagged for the cost management point of view. Basically, everything is covered. They don't have to think about it at all. And we don't have proper guardrails yet, but we're working on it. For example, say that certain branches are blocked and you can just push there – so they now have to implement it by hand. But we're working on it. Actually, this also allowed me to push the master, to create default branches and all that stuff. It's also possible to automate all of that as part of this cookie cutter. Yeah. So that's how we do it.

32:09

Alexey

**That's quite impressive. So if I'm a data scientist, let's say I need to create a model for fraud detection or customer scoring (whatever). This is a model that needs to be served online – I want to serve it as a web service. So there is a cookie cutter template that is specifically for that. As understood, all I need to do is click some sort of button and then fill in some information, or maybe run something in the command line that will ask me, “What's the project? The team, (because you mentioned DAGs).” Then I do that, execute, and at the end, I have a working project in my repo with a CI/CD pipeline for deployment. There's already a main.py file that I can deploy using this CI/CD pipeline and all the tags are assigned. Right now with this project, what I can do is already get the model from a model registry and start serving. Right?**

33:24

Maria

So how we see this is that main.py needs to be replaced with your actual logic. You still need to create a package. We utilize Databricks through data scientists by using notebooks, but likely the Databricks notebooks are not just Jupyter Notebooks – they're quite nice, for version control at least. But we also have an article for data scientists to move from a notebook to actual working production code. So we encourage data scientists to create functions, classes, and modules and move the logic outside of the notebooks – to keep notebooks really short and clean.

That's the main execution file. Then you also create the paths and packaging logic, so if you need to have either Poetry or setup.py. With that – our pipelines take care of that. They know that there is a package, that package has to be moved to dBFS, and it's all gonna work. So they don't even have to think about that.

34:29

Alexey

**Amazing. How long did it take to implement this?**

34:32

Maria

Well, the implementation wasn't long. I think it took less than half a year to implement that, but one year to tell everyone that we are doing that and agree with the DevOps engineers to give us permission to do things.

34:48

Alexey

**Ah. Because DevOps folks do not like when data scientists can mess up their Kubernetes clusters, right?**

34:56

Maria

Yeah, well.... We have no dedicated machine learning environment which only we use. They basically don't care that much, I guess.

35:07

Alexey

**I see. So it took six months to implement but before that you needed to do all this preparation.**

35:19

Maria

Convincing.

# Convincing DevOps to implement

35:21

Alexey

**Did you do this yourself as a tech lead? [Maria agrees] Do you have any tips on how to address that if somebody is also facing some hesitation from the DevOps team?**

35:34

Maria

Yeah. Three years ago, the first thing we did was go through all the brands and give them the questionnaire. The questionnaire that I was talking about has already existed for a while. That's how we started. We gave it to everyone, we did this assessment for all the machine learning models we could find, and we showed them that we are pretty much at zero. So you can say you're doing things the right way. We were also in the situation where DevOps engineers that deploy machine learning models have zero understanding of machine learning, but they were responsible for the deployment. And if they were errors, they would send the errors over email to the data scientists that would try it out in a different environment than their production is running, and couldn't reproduce there.

So this loop can go on and on and on. No one was happy, so there was already pain. There was pain – we made the pain visible by doing the assessment and showing that there's clearly something wrong going on. We wrote a whole document and how to do a data science project – “Data Science Framework” we called it – and we created a set of MLOps standards. It actually now has become an official document within our organization that the internal audit can check to see whether you're actually following those rules. So these three things helped a lot. I think there was also trust already between our departments which allowed us to move further.

37:17

Alexey

**I guess my main takeaway from that is you wanted them to feel the pain – make the pain visible. Then they realize, “Okay, we actually have this problem and maybe there is a way to solve this problem.” Then you say, “Yeah, we actually want to solve this problem.”**

37:36

Maria

“We know how to solve this problem.” [chuckles] Yeah.

37:40

Alexey

**And these people then help you convince the DevOps engineers or it was the DevOps engineers who...?**

37:46

Maria

No, I was convincing the lead of the IT platform and also the DevOps engineers. That's how we at least got our own service principles with the right permission to deploy things ourselves.

38:01

Alexey

**Okay. How large is the team? Who was on that team? In order to implement something like this cookie cutter template, reusable CI/CD pipelines – you need engineers, somebody who implements that. So who and how many people did you have on that team?**

38:21

Maria

Yeah, it's crazy. We basically did it with three people.

38:26

Alexey

**Three people. Okay. Impressive. Including you, right?**

38:32

Maria

Including me, yeah.

38:35

Alexey

**So who are these people?**

38:38

Maria

So there was me, Başak, who is my colleague with whom I work on Marvelous MLOps, and also our colleague from another department, but he kind of works in MLOps. [inaudible] He doesn't have LinkedIn so we can advertise him. Hopefully he creates it. [chuckles]

38:55

Alexey

**He should. I'm also more interested in their profiles (in their roles). Are they more engineers? Are they data scientists? What kind of background do they have? Do they already know Kubernetes and all these tools?**

39:10

Maria

ML engineers, we all know Kubernetes. We all know something about Databricks. So it's not like we started from zero. We didn't have any juniors, it was all upper-medium-senior profiles.

# Understanding what the tools are used for instead of knowing all the tools

39:29

Alexey

**What is also interesting, and the reason I'm asking that, is because in our MLOps course, we try to cover the fundamentals. We break down what we think MLOps is into multiple areas, which is something like experiment tracking, machine learning pipelines, deployment, monitoring, and then best practices. I think this is what we focus on. Then, instead of exploring all the possible tools, we just pick one and try to learn how to use that one.**

**Of course, when the graduates join a company, the tools will probably be different. Instead of Prefect, it will be Airflow or Jenkins or whatever. Instead of Evidently, maybe there will be Elasticsearch and Kibana. Instead of MLflow, there will be nothing. What I'm trying to ask you is – what kind of profile do people need to have, or what kind of things do they need to know to be able to join a company and start implementing things like this?**

40:42

Maria

Yeah. That's what I think too. Of course, it's nice to know some of the tools, but I think the most important thing is understanding the idea behind it. There are tools that fall within different pieces. There are different tools for CI/CD – it doesn't matter what you use. I really believe that it doesn't matter that much. For version control, what you use doesn't matter. For orchestration, for registry – it doesn't matter what you use, it just has to be tied together in a way that follows the principles that are written in the assessment. If you understand that and have already tried to do it at least once with some tools (doesn't matter which one), then you are good. If you just try to do it once – to combine all these pieces together using whatever tools, you're fine. I think you're prepared to do it with anything else.

41:36

Alexey

**So if somebody did a project in our course, then they're probably ready. Because the purpose of this project is to actually take all these tools and stitch them together into something that doesn't fall apart and works. [Maria agrees] Which is probably the most difficult part, right?**

41:52

Maria

It is. It is super difficult indeed. It is super difficult.

41:58

Alexey

**Also what I heard from you is that it's a good idea to actually go ahead and check the assessment we talked about [Maria agrees] and understand how much you can make sense of things there. If some things are not clear, then this is something that they can read about. For example, we don't really talk about documentation on the course. We don't talk about feature stores. Because it's not possible to put every possible thing into six weeks. So then going to this maturity assessment when, for example, they already joined the workplace and are going through this thing and understanding “Okay, this is what we should focus on.” That will help, right?**

42:48

Maria

Yeah, definitely. Definitely.

# Maria's next project plans

42:53

Alexey

**What are your plans? What do you want to do next? You already told us that you have this amazing cookie cutter template for standardization, which makes it super easy for the feature teams, for the product teams, to deploy the models. You already have cookie cutter, reusable CI/CD pipelines, automatic deployment – what do you want to work on next?**

43:24

Maria

The next things that are important for us – we can do things like A/B testing and monitoring. But those are quite ad hoc. This means that we don't have to standardize monitoring, standardized A/B testing yet. Those are extremely important to get standardized. The main thing here is that we are basically relying on decisions that other teams are making for monitoring. Because monitoring goes beyond MLOps – monitoring goes for anything. I think it's a bad idea for us to choose a tool that's different from the one DevOps engineers use. Because they are still in the process of deciding on the tool, we are waiting for it.

44:07

Alexey

**I suppose it's a tool for “traditional monitoring,” like New Relic, Datadog, Elk – this kind of stuff?**

44:19

Maria

Yeah, indeed. I think it's good enough for pretty much everything. Looking at our use cases, our use cases and demand forecast – various recommendation engines. For those specific use cases, we can fit whatever needs to be fit for monitoring for those use cases and these tools. If you need to talk about evaluating/monitoring LLMs and all of that stuff, you probably need to have something else. [chuckles] But for what we have, for like 99% of use cases, that's probably enough.

44:59

Alexey

**What about are use cases that do not fit into this traditional demand for customers. I imagine that maybe one of the teams decides to use an LLM.**

45:11

Maria

Yeah, we are now also working on it for internal use cases – internal storage base. That's the first project we are trying out with LLMs. Because it's not customer facing, I guess it's not that important. We are still in the process of figuring it out. I guess that by the end of the year, I will be able to tell you more about this specific use case. But right now, I'm too focused on other things.

# Is LLM Ops a thing?

45:44

Alexey

**We'll have to interview you again, I guess. [Maria chuckles] For others, I'm pretty sure you will probably cover it to some extent somewhere on LinkedIn or maybe in a newsletter. [Maria agrees] Since you mentioned LLMs – I see now that this new thing LMM Ops is kind of trendy. Many students from our course are asking, “Hey, when do you plan to do an LLM Ops course?” Once I got a message, “How can I enroll into your LLM Ops course?” Which we don't have. [chuckles] Apparently there is kind of a demand for it. I'm just wondering, what do you think about that? Is it even a thing? Should people really worry about that or maybe it's just one of the hype things that come and go?**

46:41

Maria

I believe that it's largely a hype thing. Having a customer-facing LLM that is not going to cost you a fortune is a real problem, especially if you're talking about another language besides English. In our use case – I mentioned we have brands in Czech Republic, Serbia, Romania, and Greece – so in English LLMs may work well, but what about those countries? I'm not so sure. Also the use cases for retailers – what are the use cases? I guess maybe search will change in the future. It will be more of a LLM-powered search – that would be very cool. Another one is maybe recipes, for the food retailers. That's something that is very popular.

There was a scandal with this New Zealand grocery retailer that suggests some recipes that are not very appealing. You shouldn't input some crazy stuff into it, obviously – but still, people may. So I don't know. It's just very little value, I think. It's very hard to measure the value from these LLMs but the costs to get it running are quite high. You can't even ensure enough GPUs, because if you're in Europe (if you're on Azure) you have a waiting list – even for large customers like we are – you wait for four months before you get a machine that you need to train your thing on. It's an actual problem, you cannot secure clusters with enough GPUs for yourself because of the high demand.

48:31

Alexey

**Of course, for English, probably many of these LLMs work. I mostly use GPT 4/GPT 3.5 from Open AI. But even when it comes to other big languages like Russian, for example (there are quite many people who use Russian in this world) sometimes the grammar is kind of funny, it uses words that I wouldn't typically use – it's kind of weird. I imagine if we're talking about the Czech language, where there are fewer people who use this language, where the model probably saw less data for this language, [Maria agrees] it can lead to even stranger results. Right?**

49:20

Maria

Yeah. Or Greek or Romanian. Yeah.

49:22

Alexey

**Yeah, exactly.**

49:25

Maria

It's because the brands in those countries are relatively small compared to all other brands together. It doesn't even make any sense to invest into it because the value is small.

# What Ahold Delhaize does

49:42

Alexey

**So you mostly work with retail?**

49:46

Maria

Ahold Delhaize is one of the largest retail companies in the world. It's not a very known name for some reason. [chuckles]

49:55

Alexey

**Maybe I should Google it right now. [chuckles]**

49:58

Maria

Yeah. [chuckles] We are quite large. It's food retail, mostly. We also have Bol.com, which is like the Dutch and Belgian Amazon.

50:12

Alexey

**Ah, I know that one.**

50:14

Maria

Yeah, it's also by Ahold Delhaize. It's not only a food retailer, but also other retail.

50:21

Alexey

**Retail is like groceries – this kind of retail, right? Or is it like online retail?**

50:27

Maria

Bol.com is fully online. I guess that's an exception – but mostly it's grocery stores or drugstores.

50:39

Alexey

**Okay. Now I know what kind of company you work for. I was thinking, “How do I ask that?” when we're almost finished with the interview. [laughs] “What does the company actually do?” I saw the numbers now online – it's pretty large. It's a huge corporation, right?**

51:03

Maria

Yeah, indeed.

51:05

Alexey

**I see.**

51:07

Maria

And all the brands do the same. Because it's all food retailers in different countries, but overall, the type of problems that everyone has are just the same everywhere. Everyone wants surge demand, forecast demand – forecast is a big thing. Personalization, loyalty programs, etc.

51:24

Alexey

**Does each of these brands have a separate team – and separate a bunch of teams – for data science and they do data science separately from the rest of the organization?**

51:44

Maria

Well, some do have data scientists, analytics teams – some even have machine learning engineers. Some don't have almost anyone doing anything like that. In our technical team, we have machine learning engineers, which is more like MLOps. We also have some data scientists that work on creating these standardized solutions for a number of brands.

52:14

Alexey

**So if a brand does not have their own data science team or department, they come to you? Or maybe they have one, but the team is small, so they come to you and say, “Hey, we have these use cases. Can you help us?” Right? [Maria agrees] I imagine, for example, Boll already has a huge team – they probably already have their own MLOps processes, right? They probably don't need your help.**

52:41

Maria

And the tool stack is also different. The tool stack is different, which makes it harder for us to work together on certain things. But we try to cooperate more with them. They have a lot of knowledge, so it's always interesting.

52:57

Alexey

**I know about them from conferences. In Berlin, there is a conference that I really called Berlin Buzzwords. I try to go there every year. It's quite common that somebody from Boll attends and presents something.**

53:16

Maria

Maybe I should, too. I didn't know about this conference. But I love Berlin. Berlin is so cool.

53:24

Alexey

**For you, maybe the more relevant one would be PyData PyCon, which happens in April. Berlin Buzzwords – I think they also cover ML engineering, but it's more like... They talk about search, they talk about data engineering, but recently, they also talked about MLOps, ML engineering – this kind of stuff. It's on topic, of course. You definitely should submit something. This year, sadly, I did not attend. But I plan to do it next year. We should be wrapping up.**

# Resource recommendations to learn more about MLOps

54:05

Alexey

**There is one question, “What is the course that you take to become an MLOps engineer?”**

54:12

Maria

MLOps Zoomcamp. Right? [chuckles]

54:15

Alexey

**[chuckles] But I guess this would be your suggestion. Right? But in your case, because the ground was econometrics, economics – how did you become somebody who does what you do?**

54:27

Maria

By doing. I just talked a lot to software engineers. Together, we kind of figured out how to deploy things. No, but we're actually going to create our own course. I can't say when exactly it's going to come out. [chuckles] But we aim for March next year. This will be specifically for data scientists aiming to be machine learning engineers. So stay tuned, I would say. Subscribe to all our media or you will miss it. [chuckles]

55:00

Alexey

**Please send us the links and we'll include them in the description. This person is saying that they have a software engineering background, and then I think you already did the plug so I don't need to promote our own course. Because this course is actually for somebody who has the same background as you. But from what you said – you learn by doing, and I think this is probably the best way to learn things.**

55:31

Maria

Yeah, definitely. But for software engineers, I would recommend the same thing – team up with data scientists. They definitely need some perspective from software engineers on how to write better code. There are some nice courses. I used to do a lot of Udacity nanodegrees before. I also used to be a mentor at Udacity at some point of time. They have some machine learning engineering courses. Maybe it's interesting to check that out. There are, of course, a lot of data scientist courses in general so it's always good to check those out.

56:08

Alexey

**In your opinion, how much machine learning should machine learning engineers know?**

56:17

Maria

I think there is a Google paper that I believe says only 5% of the whole ML system is machine learning. [chuckles]

56:26

Alexey

**It's this famous figure – this famous diagram – where ML is a very tiny part**

56:29

Maria

With the blocks, yeah. [chuckles] Exactly. I think it's about right. I think machine learning engineers definitely need to know something about machine learning. I would suggest doing data science courses and understanding what tools data scientists work and what type of things they produce and how they think about that. I think it's very important because it really influences the deployment. But because they're coming from a software engineering background, they probably know a lot about the other 95%. I guess this is the advantage.

# The importance of data engineering knowledge for ML engineers

57:14

Alexey

**Do you think ML engineers need to know some data engineering?**

57:19

Maria

Yeah, actually. I'm also now interested in data engineering. There is a course from Zack Wilson, who created a bootcamp on data engineering. That's actually a very nice one. I'm following it at the moment. I'm a bit slower than others, I guess. But I am following it. It's really nice. Because a large part of machine learning pipelines is actually data engineering. It's always the first step and the most challenging one. [chuckles] Quite often.

57:50

Alexey

**If you don't have data, you just have science, right?**

57:55

Maria

Yeah. Yeah, definitely. But also, you need to construct your data engineering pipelines, which are part of your machine learning pipelines, in a smart way. Often, those pipelines tend to run for a long time, so you really need to spend time on optimizing those.

58:16

Alexey

**Okay. Actually, the question I was going to ask at the end is, “Any resource recommendation?” But I think we covered that. I'm looking at the time and I think we should be wrapping up so I want to thank you, Maria, for joining us today. It was very nice talking to you. Maybe we should repeat this, because you said you will let us know about something – I don't remember what.**

58:39

Maria

About the course. It will be coming.

58:42

Alexey

**Yeah, but there was some other thing too.**

58:45

Maria

LLMs that we are busy with.

58:46

Alexey

**Yeah, right. Right. So there are multiple things we will need to talk about. So yeah, thanks for joining us today. And thanks, everyone, for joining us today too. Have a great rest of your week!**