1:35

Alexey

**This week we'll talk about machine learning engineering in finances. And we have a special guest today, Nemanja. Nemanja was born and raised in Belgrade, Serbia, but since 2014, he's been living in Leuven, Belgium. He's an electrical engineer turned data scientist and then ML Ops engineer. This is what we'll talk about today, machine learning engineering and ML Ops. I met with Nemanja at an amazing conference in Porto. When was it?**

2:11

Nemanja

October 2023.

2:12

Alexey

**Yeah, it took a while to schedule this meeting. [chuckles] At the conference, he gave a talk about ML Ops and I also had the chance to interview him, among other people, on stage. It was the first time that I interviewed people live on stage. That was a super interesting experience. Today, we will finally get back to this discussion and talk about machine learning engineering, ML Ops, maybe Brazilian jiu jitsu and the purple belt. What else? So yeah, welcome!**

2:50

Nemanja

Thank you for having me.

# Nemanja’s background

2:52

Alexey

**Before we go into our main topic of machine learning engineering in finances, let's start with your background. Can you tell us about your career journey so far?**

3:03

Nemanja

Okay. It was a pretty, I would say, nonlinear career journey. I first obtained my formal education in the domain of electrical engineering, as you already mentioned, in Belgrade, Serbia, where I was born and raised. My first job was actually as a salesperson. I was like a traveling salesman for some automation equipment for industry – sensors and controllers and so forth. After that, yeah, I realized I didn't want to stay in that career. I wanted to do something more technical again.

That's when I came to Belgium. I actually came to do a PhD in the domain of bioengineering – switching from electrical engineering to sales, now to bioengineering. There, I stayed for about a year and a half and I realized that, again, academia is not really the place for me, because especially after I worked a bit in the industry – we didn't click, let's say, together. So that's why I wrote that I'm a PhD dropout. I stopped with the PhD and I moved to Deloitte Consulting, here in Belgium. I stayed there for, I think, almost three years, then switched to a smaller consultancy, called Dataroots – back then it was much smaller, now they grew a lot. [chuckles] Yeah, that was also a very nice time.

From then on, I went briefly on to ING. Sorry, I forgot to mention I work for BNP Paribas – it was my first experience in the financial industry. Then I worked for ING for a year and now, hopefully, I settled [chuckles] in Europe. here in Brussels. This is a company which maintains the infrastructure for trading securities. Basically, I think the easiest way to explain to people what we do is – people usually know SWIFT, the company which you use to send money internationally that maintains this network. Basically your money hops between different banks and lands in the right place in the end. We are like Swift, but for stocks and bonds. Big companies buy and sell stocks and bonds through us. We also keep national bonds for many European countries like Ireland. I don't know if it is in Belgium and France and so forth, I will make a mistake here – I'm just the ML Ops guy [chuckles] the business guy.

But yeah, this is where I am currently. I'm in some kind of ML Ops lead function, I would call it like that, here in Europe, and I'm helping to increase the overall ML Ops maturity of my current company. This is [what I’ve been doing] for the last five years. My title was usually a data scientist, but in practice, I was an ML engineer. Now this is formalized with the title. We didn't call it ML Ops five years ago, but it was just called “end-to-end machine learning/engineering”. That's it shortly – my short trajectory.

6:05

Alexey

**Back then, like, I don't think that even the role of ML engineer actually existed.**

6:13

Nemanja

It did. I think…

6:14

Alexey

**It was more like “data scientist”.**

6:15

Nemanja

The first time my job was called “ML engineer” was in 2019. And I know it was the companies that wanted to differentiate, because “data scientist” was and still is, I would say, a very vague term – it doesn't mean a lot. People use it to describe many different things. But when you say you're a machine learning engineer, it’s pretty specific. You're not doing stats – you're doing machine learning. You're not doing dashboarding, because people hire data scientists and then get them to do the dashboards. So I like this term, “machine learning engineer,” but still, I would consider it a different role compared to a data scientist.

6:55

Alexey

**Yeah, definitely. I want to add a few notes – I see questions. For some people, it might be confusing, because here, today, in our community… by the way, this for editors, please cut this later. Today, we actually have a course launch. We are launching our data engineering course and I see people who joined the stream today thinking that this is related to the course – it is not related. The course launch will happen at 5pm tonight. This is just an interview. Sorry to disappoint you.**

7:30

Nemanja

It's not a waste of your time, you can stay. [chuckles]

7:31

Alexey

**Yeah, this is a very good conversation – you will definitely learn a lot – but it's not about data engineering, it's not about the course. This is a podcast interview. Now I hope the current of people who joined will not drop too much. [chuckles] Another thing I forgot to mention because it has been a month since the last interview – I forgot to give a shout out to Johanna Bayer, who helped prepare all these amazing questions that we will talk about today. So thanks, Johanna, for your help. For editors, please move it before the first question. [chuckles] Okay, now we continue.**

# When Nemanja first work as a data person

8:18

Alexey

**By the way, your first job as a data person was Deloitte, right?**

8:24

Nemanja

I would already count the PhD because it’s a job of a research and teaching assistant, there was a lot of data science there already – systems modeling, control. That's when I started using Python intensively. I was already doing that before but this was when I was getting paid to do Python.

8:45

Alexey

**Okay. And Deloitte is a consulting company, right? You probably needed to speak French. [Nemanja disagrees] No?**

8:56

Nemanja

No, actually not. In Belgium, not really. [chuckles] It's always advised and I think it will certainly open up many more doors if you speak the local language (in Belgium, it's Dutch and French, and you should speak English) but for the more technical positions like technology consulting, it's okay if you speak only English. But for certain clients – for example, if you're working with the government – you need to be able to speak fluently.

9:25

Alexey

**For NATO, right. I remember this is where their headquarters are, right?**

9:30

Nemanja

Yeah, in Brussels is the NATO headquarters. Yeah.

9:32

Alexey

**Yeah, I remember going from the city center to the IBM office and the NATO building. It’s huge.**

9:40

Nemanja

Yeah, they have a new fancy building – an even fancier one.

9:45

Alexey

**Anyways. So you have worked at a variety of different companies, Deloitte, ING, Dataroots, BNP [Nemanja agrees] Now it's Euroclear, right? [Nemanja agrees] Most of them, if not all, involve finances and banking, right?**

10:03

Nemanja

In the last four-five years, yes. So not all of them. But also with Deloitte, you're a consultant, you switch fields relatively frequently. I was also in the biomedical field– for the vaccines, GSK (GlaxoSmithKline) vaccines, we also did projects with Amazon with Abebe, which is the industrial manufacturer. So there was a lot of variety there, which takes time to mention. But yeah [chuckles]. Let's say I settled myself in the last four or five years in the financial industry. That's important.

# Typical problems that ML Ops folks solve in the financial sector

10:35

Alexey

**So what are the typical problems that data scientists, machine learning engineers, or ML Ops folks usually solve in the financial sector?**

10:46

Nemanja

You mean use cases. From the business perspective, I would say there was a clear common line between all the companies where I was in the financial industry. On one side, all of these companies have very strong regulations and they have very strong compliance requirements. These fraud and money laundering cases are always there – it's simply imposed, you have to do it. On the other side, I think, where most of the business value comes from… For this compliance, fraud, AML – you have to do it. If you don't do it, you have to pay billions in fines. But it's an overhead that does not give some concrete added value – it just saves you from a certain very big penalty that you have to pay. On the other side, things that really give value in the financial industry are so-called “smart automation projects”.

In essence, it's usually processing semi-structured and unstructured data, like documents, emails, and so forth – whether it is information extraction: extract an account number, extract the signature, route the email or a case to the right department. It's also very important. All these companies have a lot, a lot of people who are manually processing all these things, still. There is still a lack of, I would say, proper forms for data information so things are arriving in some kind of semi-structured (poorly structured formats). Then you have a lot of parsing, a lot of interpretation, and so forth. So those are the main things, I would say. Definitely, I think it's 80-90% of the cases that we work on.

12:32

Alexey

**So first is regulation compliance. If you don't do this, you get huge penalties, so all the companies have to do that. [Nemanja agrees] They don't want to lose money. Then the second category is more internal, right? I imagine… [cross-talk]**

12:48

Nemanja

Yeah, it’s [things] like process efficiency. But in the end, it impacts your customer as well, because if you are doing this, slowly, manually, your whole case processing takes a lot of time. So the faster you can do it… I also didn't mention RPA. It's not something I do, but robotic process automation is also very big in the financial industry. The thing that also connects all these companies is that they were, in some way, all pioneers of this digital era. They all had, very early, their mainframes and all the digital frameworks. And this is now a bit of a curse.

Because they have all these old systems – they were the first ones to have all those systems. They have a lot of legacy code, they have a lot of legacy infrastructure, and it's very hard to modify or to improve many of these things. They were not really built with enough forward thinking – in a forward compatible manner. So now there's a lot of trouble that is caused by that fact.

13:53

Alexey

**I imagine that there are also some customer-facing applications. Say I want to make a transfer and I have an app – then I just take a picture of my receipt and then… [cross-talk]**

14:07

Nemanja

Oh yeah. Mobile applications are also a big thing. Indeed. There, you see a lot of variety? The mobile applications I mainly saw as a user, because I'm currently using two banks in Belgium. I could see there's a big difference. Some banks pick this really, I would call it, lean, simple approach, like ING in Belgium and others, like the KBC bank, take a completely different approach and really put everything in the app, literally. I think you can almost make lunch for your kids in the app. [chuckles] You can pay for the parking, you can pay for the train tickets, you can order food. You can do so many things in the KBC app. And in the ING app, they just chose this, “Oh, we just give you the banking thing.” They have their reasons, probably

# What Nemanja currently does as an ML Engineer

14:57

Alexey

**As an ML Ops person (ML engineer), what do you actually do? Because I imagine… We talked about more use cases, but as an ML engineer/ML Ops person, you don't work on the model that recognizes the account details, for example. Right?**

15:17

Nemanja

Yes, correct. That was the second part that I wanted to say. On one side, you have the typical business use cases, and on the other side, you have the typical work of an ML engineer in finance, which is mainly, I would say, modernization. It’s mainly getting things to work. In general, an ML Ops engineer is a person that, in my view, tries to abstract all the non-modeling parts of work from the data scientists. My idea is that I want to be kind of like a service to data scientists to make a kind of a platform or a framework so that they can focus on the difficulties that they have in extracting the right entities and making the right models. And then I basically handle all the rest. I give them a standard project structure for the project, I am the one that makes the…

16:12

Alexey

**By “them” you mean data scientists, or who?**

16:16

Nemanja

Sorry, I didn't understand the question.

16:18

Alexey

**Yeah. When you say, “I give ‘them’,” you’re referring to data scientists, right?**

16:24

Nemanja

Yeah, data scientists – we work together in a team. We usually have two-three data scientists and an ML Ops engineer per project. We work together – it's not like passing things over the fence. It's really working together and reviewing the code together. Everything is very closely coupled. My part is to create a repo, create the project structure, create the CI/CD pipelines, figure out how they need to deploy, what will be the target deployment platform for this – will it be a cluster or maybe we'll look into and select the cloud model.

One more thing to mention here as a side note, in the financial industry, I think still mainly on-premise architecture. There are some movements towards the cloud and, I would say, there’s definitely a persistent direction there. But still, there are a lot of core systems that are on-premise. For me, it's also, “Will we deploy on the cloud, or deploy on the on-premise cluster, or on the OpenShift cluster?” There's also networking, “How do you open up certain firewalls for this application to communicate with this API?” And on and on. Those are the main things.

17:41

Alexey

**So it sounds like it's more or less a typical job of an ML engineer, except that you said that there is a lot of modernization work, which I don't know, depends on the organization… [cross-talk]**

17:55

Nemanja

Yeah, definitely. I think the ML engineering role is like a software engineering role, which is very generic. [Alexey agrees] And the more you go to data science, the more it becomes business-specific. I can really see that I have very little exposure to the actual business side of all the projects, which I miss pretty much. When I was doing data science work, I was always much deeper into the communication with the business. But I can live in my bubble of code, [chuckles] and DevOps platforms. So yeah, it's pretty much similar, but yet, there is this thing of modernization, of change management, of seeing how to not fit ML Ops into the classical DevOps, because these companies have pretty established DevOps practices and governance. Now you need to see how to somehow integrate (how to smuggle in) ML into this whole thing.

# The obstacle of implementing new things in financial sector companies

18:52

Alexey

**And these existing DevOps practices, platforms, existing governance… What is there? I assume that it's not the most modern solutions. It's probably time-proven things that… You mentioned OpenShift. [cross-talk]**

19:12

Nemanja

It's really a mix of things. I think when they buy something, it's the newest one. It's usually so. When they say, “Now we're gonna go with this,” we take the best one. But that will probably not change over the next 10 years. That's the thing. Things are slowly moving, slowly changing. I think just the internal IT landscape is so big, that everything has a big impact and you already have a lot of applications. To move, let's say, from one logging (centralized logging) we have Splunk now. This is a modern solution, I would say – I don't know if there's something especially better. But imagine if suddenly you want to switch to another thing – that will be a big cost. That's, I think, the main issue – the main obstacle there is the slowness and the whole planning goes into big time windows.

20:09

Alexey

**Is it specific to the financial industry? Or is it more about the traditional corporate environment?**

20:17

Nemanja

It's not in all corporate environments. Definitely not. I think it's very specific to the financial industry. I think it’s the case for any overregulated industry – an industry which has very big involvement of regulators, of governments, and international bodies. They are very, very risk averse. It's better not to do something than to introduce some kind of risk. In these companies, you also have all these trainings – every month or so, you have some kind of risk training: how to spot risk, how to handle risk, how to manage… Risk, risk, risk. Risk mitigation everywhere. You learn to always think about that.

20:58

Alexey

**Yeah, I remember I worked at UBS. That's a bank. I think it's “United Bank of Switzerland”. Well, it doesn't matter. It’s a financial Institution – a very conservative institution. I was working there as a Java developer. We would release every month. If a release doesn't go through that month, it means it will go to the next month.**

21:21

Nemanja

Next month. Okay. We’re not that rigid. [Alexey chuckles] For us, it's release cycles. There's a whole department, called “change and release management”. When you want to release something to production, you first need a review from your own team to create this kind of change and to say which exact commit (which exact build) you will release. You need to show that it was first released with a test, and that it worked and that this was released, let's say, at least a week before the release to prod – that it was properly tested.

Then you have some three different teams where one person from that team needs to approve. But if it fails, you can do a rollback. There's a procedure to do a rollback. And you can have an emergency change if you need a bug fix. To do something, you can have an emergency change. Of course, you need to follow the procedure for the emergency change. [chuckles] The important thing is that, in the end, you always know what is in production, who put it there, and when they put it there.

# Going through the hurdles of DevOps

22:25

Alexey

**Okay. So these are the existing DevOps practices and the governance framework that you mentioned, right? [Nemanja agrees] Which sounds like a bit of a hassle, to be honest. But there are reasons for that, right?**

22:41

Nemanja

It slows you down. But I would say, you learn how to do it – you learn how to do it quickly. Every time, it's faster. The first time you do it, people don't know you. It's all about people’s trust. The first couple of times people are really looking at your pull requests in detail. “What is this guy doing? Who is this guy? What does his code look like?” And every next time, when you start deploying frequently, you get all these approvals much faster, because people say, “Okay, this guy never crashed anything. There were no incidents,” and so forth.

You know who to ping – it's the little people connections – you really know who to ask on MS Teams, “Hey man. I have this change. Can you please approve?” Lately, this was not really an obstacle. In these corporations, it's always… You need to help people. After a couple of years, you become really productive when you really make your network. Then things start going very fast.

23:39

Alexey

**Yeah. I imagine that, since there are these processes that were set up ages ago, that they're thought through, and they exist for a reason. And you, as an ML engineer/ML Ops person, need to stick to these processes, right? You work on ML Ops, you work on these machine learning pipelines, CI/CD pipelines, or whatever, and what you do needs to stick to the guidelines from the DevOps people. [Nemanja agrees] How difficult is this? How difficult was it to map the ML Ops processes to this DevOps framework?**

24:22

Nemanja

Well, I would say that we are still not fully not where I want us to be. Definitely. There's still improvement to be made. But it's a journey. As I said just a moment ago, it was hard at the beginning. But luckily… I mean, I did not come into an empty room. There were already people in the company who did a lot of previous deployments, so we “piggybacked” for a while on the team of data engineers who were doing really frequent deployments. They explained and they guided us through the whole process until we became independent enough to do it ourselves. You know what we say, “A living person gets used to everything.” [chuckles] This is also true here.

You stop noticing it at some point. When I talk to somebody from a startup and they hear, “Oh, you have to do all these things.” For us, it's no longer something that is an issue. It's like, “Okay, this is how you do it.” It's more like muscle memory – you click here, you make this, you prepare that, and you go to production. I heard much worse stories from certain companies, where you have to literally have a sheet of explanations for every little change, and then you need to have a JIRA ticket source and ServiceNow TFS tickets – doesn't matter. There is much more bureaucracy in certain other places, so I think we have a decent sweet spot here.

25:55

Alexey

**Yeah. I remember… Back to my UBS experience. To be fair, it was a long time ago. It was 10 years ago or more. There is a comment that says, “Today, in finance, CI/CD is a part of the daily routine,” which is a good thing. It’s everywhere now. But I remember one thing that we were super careful with, which was the open source tools that we used. We had to do these things you mentioned, but instead of every change – say we wanted to introduce a new open source package that wasn't used before (a library) – we had to write an explanation why this library would be applied. Is it a good license? Do you have [something like that]?**

26:44

Nemanja

Yes. There is [something like that] to a certain extent. But currently, the default is – we have Artifactory, which is an internal package registry, JFrog Artifactory – and it mirrors the public pipeline. I think only if it registers some kind of a critical vulnerability, then it is going to blacklist a certain package. Recently, there was a situation where I wanted to ask for another package index – not a package from PyPI, but it was a separate index. There, I had to give an explanation. But it was a short chat with the person that was in charge of this. I could really just fill out a form and then it worked. So it was not too hard.

27:28

Alexey

**I imagine that if you want to use PyArrow instead of plain Pandas, you don't need to… [cross-talk]**

27:36

Nemanja

No, it's okay. We don't have an issue with that. Yeah. But if it's a completely separate thing, which is not on any kind of a public repository, then you have to justify why you need to manually import something. Yeah.

# Working with an on-premises cluster

27:51

Alexey

**How difficult is it to work on-premise? I imagine that there is this OpenShift cluster and then, I guess, there are all these procedures, standards, templates… It should be pretty smooth, right?**

28:06

Nemanja

Yes. Currently, we are working mainly on a classical, I would say, Hadoop cluster. There's the OpenShift cluster, but there is something next to it. We're not currently deploying there. But yeah, there is a certain project structure for deployment there. There is a standard pipeline if you want to go there. So it's pretty much already ironed out by the data engineers before us, and we are just reusing that same approach. Working on premise is… I don't know.

I like it. It's really close to the metal – it's simple, in a way. You have the machine, you have the operating system, and you don't have all these disarrays of services that you have on the cloud. They are also helpful in many, many cases, but… You get to learn a lot about Linux. You get to learn a lot about bash scripting, about networking, SCPing, SSHing, and these things. Yeah, I like it. I got used to it. But I would say that it’s much simpler in some ways, but it requires, I would say, more knowledge of computers and operating systems – Linux and networking. But that's the main difference.

29:24

Alexey

**But do you actually need to order hardware? [Nemanja disagrees] I imagine that you want a new thing… [cross-talk]**

29:30

Nemanja

No. Not my team, my team. There's a whole… The thing is, on-premises requires a team to maintain the infrastructure. [Alexey agrees] That's the main deal with smaller companies, I would say, is that they would have to hire somebody to maintain a data center. For smaller companies, it's a problem. For bigger companies, especially in the financial industry, that's really not a problem. I think that's the main attractiveness of cloud – that you can easily start and have some kind of a data center (a rented one) and then you can expand it and so forth.

But recently, there was a certain number of machines being added. Or you say, “Oh, I don't have the capacity,” and there's a whole team that manages that. So there's a platform engineering team. You say, “Guys, this is what we need. Here’s the money. [chuckles] Please do the work.” And they just tell you, “Okay, now you have all these machines added. Include them in the pipeline.” That's it.

30:25

Alexey

**For your needs, you can just assume that it kind of works like cloud – when you need resources, you will have them – unless it's super gigantic that you need to ask in advance.**

30:37

Nemanja

Yeah. In a way, it's like cloud, but you don't click and make things – you ask people. You have to make a request to people and they do these things for you. Usually, that's also one of the things, to find the right request to make, to know who is in charge of what and where – that's a bit of a journey to learn and understand.

# “ML Ops on a Shoestring” (You don’t need fancy stuff to start w/ ML Ops)

31:02

Alexey

**So we met in Porto at a conference, Data Makers Fest, and you gave a talk. [Nemanja agrees] It was a very nice talk. I attended that talk. I think I was even a moderator. I don’t remember. [chuckles] [Nemanja agrees] Yeah. In that talk, you showed that you don't need a lot of fancy stuff to start with ML Ops, right? [Nemanja agrees] Can you maybe give us the main ideas from that talk? It was also interesting for me personally. How did you arrive at this idea and why? What caused you to come up with that?**

31:39

Nemanja

Yeah. It connects with my experience in the financial industry. The title of the talk was “ML Ops on a Shoestring”. I think now, since two days ago, it's available on YouTube – if you just Google “Data Makers Fest Nemanja”…

31:55

Alexey

**We'll put it in the description.**

31:57

Nemanja

Yeah. It’s there. Basically, the idea was, as I say there, we all operate on a certain budget. This can be a time budget, it can be a money budget, or a people budget and so forth. If you're in a larger organization and you want to implement ML Ops, you need some kind of a prioritization scheme, or some kind of prioritization exercise. That talk was about that, “How do you start? What would be the minimal set of ML Ops features – of environments, of components – that you need to implement in order to say, ‘Okay, now we're doing some kind of ML Ops.’” If I remember correctly – I don't want to repeat the whole talk here. [chuckles]

In general, we said, “Okay. First thing, you need to have a development environment and a production environment.” That's the basic thing. “Ideally, you should have a test environment, so that you can independently develop, independently productionize, and independently test.” In the middle, as the central control tower, you do have a DevOps platform, where you integrate all your code, where you launch, where you have the whole audit trail of all the changes of what anybody did. I remember, I called it then, the “cover-your-ass Ops,” basically, in order to play the Git blame game, ultimately. Next to that, you need to have, I would say, as the very bare minimum, some kind of a monitoring solution to know if your application is alive. [chuckles] If it just dies, it will not tell you that. You need to have a model registry and you need to have some kind of a version data registry. I think that was one of the things I stressed two times.

This is also something where it's often an afterthought – we don't often realize that data produces code and that not versioning data means not versioning your code in ML. So that was one of the things if you want to have, what I then called, “the crown jewel,” which is having reproducible ML pipelines. Why are reproducible ML pipelines necessary? Not because you will actually have to reproduce a model. I never had to reproduce a model a year later or anything – that was never a requirement. They always say, “Yeah, we should be ready to reproduce a model,” but actually, if you can reproduce a model, that means you have control over your ML process. That's the main thing. That just proves that you know what you're doing. And if you cannot reproduce anything, then who knows what's in production? That was the main thing. I did not say, “Oh, you should do it like this and you should stop here.” [chuckles] But if you had to choose the minimum set of ML Ops components, that was my proposal.

34:40

Alexey

**But not only that. Now, if somebody has not seen the talk, and listens to us talking about that, they imagine this complex thing with all these tools that do all the things that we mentioned. But what you showed us in the talk was that these components, although they sound complex, they're actually not. You can start super simple.**

35:05

Nemanja

Yeah. For example, the model registry can be just an S3 bucket. And that's okay for beginnings. I like this. This is more like what they call “a consultancy talk”. I learned recently (well, not recently) a term they call a “tactical solution”. It's basically an ugly solution which works, until you reach your “strategic solution”. That's why it's called “tactical”. What you really want is MLflow or Databricks, but in the meantime, you have just some kind of a mess of Excel files or something. When you call it a tactical solution, it immediately sounds like a solution and like something “tactical,” something clever and thought out. You know? So I would say this S3 bucket is a good tactical solution for a model registry, and also for versioning data and whatever.

# Tactical solutions

35:57

Alexey

**Yeah, I love how it sounds, “tactical solution”. [chuckles] You would call this a “temporary solution” but then, five years later, it's still there. [chuckles] Because… [cross-talk]**

36:06

Nemanja

There's a famous quote. They say, “There's nothing more permanent than temporary solutions.”

36:12

Alexey

**Yeah. So, “There is nothing more strategic than a tactical solution.” [Nemanja chuckles] That's nice. So what led you to this idea? What is your experience [that led] to give this talk? What did you see in your experience that caused you [to think of this]?**

36:30

Nemanja

As I said, I work in the financial industry, where you're moving very slowly. Imagine that you have a bunch of bad guys and every bullet costs you 10,000 euros. You need to think very carefully how you're gonna use your ammo. That was the thing, “What is the bare minimum so that you just tick the boxes and can say, ‘Okay, I’m safe. My process is controlled. I have reproducible training. And then I will move forward to user experience and have a nice dashboard.’” For example, if you need to implement monitoring – your boss says, “Hey, do we have monitoring of our ML models?” At the very minimum, you need to have some kind of log of your models and of their predictions, even if it's in a text form. You can spin up a Jupyter notebook and analyze that. But if you don't have that, then nothing – no Power BI fancy dashboard – will help you. So start from the very bare minimum to cover your butt, and then move forward to the fancy things and making your life easier.

37:41

Alexey

**So the idea here, if I try to summarize it, is to kind of close the loop on the process as far as possible, so you have a process there… [cross-talk]**

37:51

Nemanja

Yeah, you need to have a complete framework in a simple manner. In Agile, that whole thing… I don't really Agile… [chuckles] I don’t call it Agile – we call it “the prototyping approach”, which means you make a first end-to-end prototype, and then you iterate and improve bits and pieces. As a whole, it needs to be… you cannot have a Porsche without one tire. You have the fancy doors, you have the fancy motor, but without one tire, it's not gonna go. So first make it operational and then pimp it. [chuckles]

38:26

Alexey

**Yeah, I like… Agile – I mean, it's good to be able to start. Let's say, there is a new team, and you want to make sure the processes are there, so that what people are doing is not chaotic, right? So you establish the framework, you start using it and then you make it complex as you grow – as you become more mature.**

38:48

Nemanja

Yeah, but I would say that the main issue I have with Agile is that it forces you, in a way, to try to make something… I would call it “demo-driven development”. You need to immediately have something to show. I think it forces you, in a way, to create some kind of technical debt because you want to make something quick and dirty and you're thinking from demo to demo if your sprint is like two weeks. I think the beginning of a project should start in some kind of… not “waterfall mode”, but something like, “Okay, let's set it up. Let's set up the groundwork.” If you're building a bridge, you cannot first throw a log over the river and then build a bridge on top of that log. You need to do a lot of groundwork and things – you cannot do it in sprints. Many things… You also have the standard – you know the diagram that they show in Agile training? You first have a tricycle, then you have a bicycle, then you have a car, and you have an airplane. What I always ask is, “Okay, have you ever seen an airplane that started as a tricycle?” It doesn't exist, you know? At one point you had to break down the whole tricycle and make an airplane. So I think that's a bit of…

40:03

Alexey

**I see what you mean. We can just say that everyone can deploy whatever they’re working on to production. It's kind of a process, right? But does it bring us anywhere? So you need to… [cross-talk]**

40:14

Nemanja

What can you really know about a complex problem in two weeks? [Alexey agrees] I mean, there are so many things to explore. I think you can use it to set up some kind of… If it's like the fifth project you're doing – if you're doing some kind of mobile app or something, where you know that 50-60% of it, you can already start and do it, and then later, you will tweak the GUI or something. Fine. But in machine learning, so many times, you start – it takes you a month to get the data, to understand the business, to do this and that. On my ML Ops part – okay, I can immediately make the project, I can make CI/CD pipelines, I can make the API function, and then have a “Hello World” model – say hello from the API. But then it's going to wait for a month or two for a concrete model to sit in there. So certain parts, yes, but the exploratory research and development things… I don't think they fit very well into the whole agile philosophy.

41:14

Alexey

**So that’s why you have two-three data scientists per one ML engineer?**

41:16

Nemanja

Usually, yes.

41:18

Alexey

**Because doing the data science stuff takes more time.**

41:24

Nemanja

Yeah, indeed. Indeed. Because there is… My work is usually more reusable. They all develop similar models, but they all have to go and talk to other people in the business and understand their needs – back and forth, back and forth. They have many more meetings with the business people than I do. I don't have meetings with the business people. I just talk to the data scientists. Yeah, there's the data, there's the cleaning, the understanding – this iterative process is much more. For me, it's pretty exact – my API works or it doesn't work. It doesn't deploy or doesn't start. I don't need another person's opinion to know if my API works – or authentication or whatever. That's my fortune, I would say. [chuckles]

42:14

Alexey

**So if I overly simplify it, then – let's say you have three data scientists. After a month of work, they come in with three models, but your role here is to make sure that all these three models can be deployed in a similar fashion. These three models are different – they solve very different business problems – but your role is to make sure that there is one platform (one piece of infrastructure, whatever) where all these three models can easily fit. For you, it doesn't matter what this model is doing.**

42:45

Nemanja

Indeed. I mean, we agree from the beginning. We say, “Okay, let's see. You want to use spaCy? Do you want to use Scikit Learn, PyTorch?” and whatnot. Then we immediately start to accommodate their approach, their ML frameworks, in this overall framework. Also, a lot of my work is really code review – ML code review – because you don't want just any kind of code. For data pipelines, I try to make sure that they are always modular, that they are testable, that they're not just one giant script. So there's really a lot of interaction. And it's not really a one way process. Because also when I have my own pull requests, I also submit them to review for them, because they also use Python. They also helped me sometimes spot mistakes in my code. So yeah, it's really teamwork in that sense.

# Platform work and code work

43:39

Alexey

**Do I understand correctly that you have two categories of work? You have the platform work and then you also have the code standardization, code review work, [Nemanja agrees] where you help data scientists with their projects. Then in addition to that, you maintain the platform where data scientists can deploy.**

44:00

Nemanja

Well, I don't really maintain the platform. My main work – what I actually create in this company – the approach and… I maintain the applications in production. I’m the one who deploys, and I'm the one who pays attention if there's alerts (if something doesn't work), I go there, and I need to fix something. That's what I do. It's not their problem. But the platform is, in terms of hardware, it's not part of my work. But I also something… [cross-talk]

44:25

Alexey

**More like an approach.**

44:26

Nemanja

Sorry?

44:25

Alexey

**It’s more like not physical… It’s more like an approach.**

44:32

Nemanja

Indeed. Yes. There's also a library, which I maintain with my fellow ML Ops colleague, which… We created this library, which is like a framework on top of FastAPI, which then also allows for the creation of new projects. We saw, “Okay, every other project, we're doing this.” So we put it all in one library so that tomorrow, data scientists could maybe even independently create a whole API with their model. That'll be some kind of ultimate goal.

# Programming and soft skills needed to be an ML Engineer

45:04

Alexey

**I noticed that we have quite a few questions from the audience. Question number one and two – I'll combine them. “What kind of programming skills and soft skills do you need to have as an ML engineer?”**

45:19

Nemanja

I think ML engineering is, first and foremost, more on the hard skill side. I think it's very close. It's very distant from data science, I would say. I think it's useful to know data science because you need to talk to data scientists – they are your clients, let's call it like that. [chuckles] But I would say python programming. If you're in the cloud, then knowing the cloud services of that specific cloud you're working on. If you're on-premises, then it's Linux commands, bash scripting, networking (a bit of networking – you don't have to be an expert, but you have to be able to survive in that environment). So that's why I'm just working on the projects.

But I think soft skills are also important for this “change work”, I would call it – if you want to bring something new to your company, and you want to raise the maturity of ML Ops in your company, you need to have this kind of “evangelical” [chuckles] trait to go around… “Missionary” or what do you call it – you need to go around and talk to people and bother people and ask and pull their sleeve, and you need to be persistent. Then, ultimately, [you need to be] a nice person – not get into conflicts with people because you're annoyed [chuckles] that something doesn't exist. [cross-talk]

46:40

Alexey

**But just in case… You have Brazilian jiu jitsu.**

46:44

Nemanja

[laughs] That will be the end of the story. [both laugh] I will not get anything – I will not get a simple Python pipeline in the end. [chuckles] But yeah, I think these skills are important to create change in any kind of organization – to be able to build networks, and present things. If you want to get a certain kind of… Let's say you want to get the fancy vendor-based project, sorry… All registry – You need to convince your boss and your boss speaks business language.

So that’s something I want to make a talk out of – how to change organizations, and how to really sell internally certain technology solutions, and how to fight for those things. One of the main things, I think, is to align your goals with the goals of your superiors and to speak their language. As I said, in the financial industry, it's about risk. So if you can show how your solution reduces risk, for example, for the company, that's what they like to hear. That's what they can sell upwards. So you need to help them sell that upwards. I think those are the main skills.

48:02

Alexey

**That sounds like your skills (your experience) working as a salesperson helps, right?**

48:09

Nemanja

Yeah, a bit. A bit. That helps. Consulting also helps, because in consulting, there's a lot of sales also. In consulting, you have a day job of implementing the things you need to do and then you have a lot of work in selling new proposals to new clients and explaining how you bring value, how you do this and that. There's always this translation work there. If you just talk about technology, nobody cares. I mean, nobody cares –it's always hard to follow. But if… [cross-talk]

48:37

Alexey

**Data scientists maybe care, but if you talk about top management, they care about… [cross-talk]**

48:41

Nemanja

The people that open up the purse to pay for your solution, they want to know, “How will you make me sleep better at night?” I think that's [chuckles] the main thing.

# The challenges of transitioning from and electrical engineering and sales to ML Ops

48:55

Alexey

**Okay. There is an interesting question from Debora, which… We talked about sales a little bit. “What was the most challenging aspect of transitioning from doing sales and doing electrical engineering, to a machine learning role, especially in terms of technical skills?”**

49:12

Nemanja

Well, I think for me, the sales part was more of a deviation, in a way. My formal education in electrical engineering was… I already had machine learning there. I already had a bit of python. Electrical engineer, in the domain of signals and systems – so it's systems modeling, which is something I already did there. “Machine learning” was still an emerging term, but then later, when I started doing machine learning, I was like, “Oh wait, I know these things.” [chuckles] We have the mathematics, the signal modeling, single processing, control – on top of that, we did control, “How do you go from modeling the system to controlling the system?” Which is, I would say, an extra next step.

I can’t say that I had some kind of a big struggle and battle to get into it. Maybe the hardest thing for me was understanding this probabilistic way of thinking. Because most of my engineering was pretty exact – you calculate a number, and you calculate a solution to this. Then to understand distributions, and how, for example, you add two probabilities – I still remember that I just could not get my head around it. When you have one distribution and another distribution, if you add it together, is it one of the top of the other? [chuckles] What does it look like? I remember, I was actually arguing with my professor and I was wrong. [laughs] I was convinced he's wrong. [chuckles] But he found a gentle way to tell me, “No.” [chuckles]

50:47

Alexey

**That's why that person is a professor. Right?**

50:50

Nemanja

Yeah. It could have come to my mind, maybe. But no, I was so convinced that he was wrong. [chuckles]

50:57

Alexey

**But sometimes it also helps not to… How do you say it…? “Put anyone on a pedestal.” [Nemanja agrees] Just because of their formal role.**

51:04

Nemanja

Yeah, indeed. But this guy doesn't… [cross-talk]

51:07

Alexey

**It doesn’t mean that they are right, right?**

51:09

Nemanja

Yeah. Yeah.

51:10

Alexey

**I guess for you, this sales role – this deviation that you mentioned – it was more difficult than the transition.**

51:19

Nemanja

Yeah. It's like a first job. I was like, “Okay, let's do something, whatever.” It's close to sales of the technical equipment. What I should have programmed, I was just telling that. And I was good at that because I was not a typical sales guy – because I was like a “nerdy” guy that actually knew the equipment. I actually knew what they needed to do with that. Then the technical guys I was selling it to were like, “Oh, you really know something. Oh, you have a degree!” [chuckles] [cross-talk] …configure, and I even helped them solve technical problems with that equipment and that's something they appreciate. I would say, if we're talking about business-to-business sales, like this was, it's very important to work on long-term relationships, and not just push, push, push to sell. This means even to decline a sale if you really think it's not solving somebody’s problem.

52:15

Alexey

**But I imagine that the work of electrical engineers is not actually soldering components, but more like… a computer, right? [chuckles][Nemanja agrees]**

52:24

Nemanja

That's an electrician. That's an electrician. Electrical engineers – I mean, you have the power line, you have the high voltage electricity, you have microelectronics, you have physical electronics… Biomedical engineering was also there, so we learned about medical imaging, for example. That's also a very interesting domain. Not soldering.

# The ML Ops tech stack for beginners

52:51

Alexey

**[chuckles] Okay. Another question, “What is the general tech stack for machine learning for a novice/beginner?”**

52:59

Nemanja

What's the question? Can you repeat?

53:02

Alexey

**Yeah. The question is, “What is the tech stack for machine learning for a novice/beginner?” I'm just wondering. It's quite a broad question. I guess the question is asking, “What kind of technology [do you need to know] if you're a beginner in machine learning?” Tech stack. [cross-talk]**

53:23

Nemanja

For a beginner, I would still say “Python, above everything”. Python is definitely the glue language, which connects everything. It's even getting injected into browsers and into Excel recently. [chuckles] So I think you cannot make a mistake if you invest your time to learn Python properly. With libraries, it changes all the time… It changes all the time. So your basic skill is to be Googling [chuckles] and being ready to change. SQL is definitely something that stays. Pandas has a long tradition, but now you see people using more and more Polars, and this and that.

But still, you see that Pandas has such a strong legacy that some other projects have accepted the Pandas API – the Pandas way of doing things, although it's not pandas. On PySpark, I think you also now have some kind of a Pandas API – you can also use Modin on Ray as a Pandas (it looks like Pandas, but it's not Pandas). In that sense. I think it's also just fine to start with that. I think it's good… I would say open up the job postings and look at what people are looking for. Definitely do that. That's how I did it every time. You open it up… For example, how I started learning Python – at university we did a lot of MATLAB. I was like, “Okay, I like MATLAB, actually. It's a really cool tool.” But then I was looking for jobs and I was like, “Okay, who’s looking for a MATLAB person? Nobody, apart from some German car manufacturers or parts manufacturers for cars.” Nobody was using MATLAB and I was like, “Let's look for the positions I want. Oh, everybody’s asking for Python. Let's learn Python.” I would say, “Don't obsess.” No one tool across the stack that will be good. They are all pretty similar. [Alexey agrees] In the end, it's easy to transition from one to the other.

I would say learn Pandas, Python – those are the basics – and they're still being used everywhere, and all the rest is just derived from there. It's good to know how to do something in the cloud. I had a lot of pet projects – I made websites – I think it's good to know how to make a simple website. Just basic HTML, CSS, JavaScript. It can look as ugly as hell, but if you know just a bit of that – how to compose everything together – I think it's valuable.

It depends on how much time you have. If you have unlimited time and you're young, I would just say to go wild and learn everything. [chuckles] Focus on projects. Focused on making something end-to-end, and you will see what's required there. You will be forced to [find out].

# Working on projects to determine which skills you need

56:19

Alexey

**Yeah, I think that's the best [advice]. Instead of thinking, “What kind of tech do I need to learn?” Think, “What kind of projects do I want to make?” [Nemanja agrees] “What kind of problems do I want to solve?” And then, “Okay, this is the problem I want to solve. I want to detect when the cat’s drinking water is gone. I want to detect that moment.” Then you have an idea of what you can implement. And then you think, “Okay, what do I need to implement that?”**

56:54

Nemanja

Yeah. But the thing is, I think it's mainly a question of not what the best tool for the job is, because you have many rich domains, and you have many tools that are good for the job. But then the question is, “What will get me hired?” [chuckles] [Alexey agrees] When I think about what to learn, I think, “What's gonna increase my chances of getting hired?” And then you need to just explore. Be a data scientist about the tech stack. What's the best way to be a data scientist about the tech stack? It’s to go and look for job descriptions, and see what people are asking for.

Always know that every job description has three times the required [things] than what is actually needed in the stack. If they find somebody that knows a third of what they’re asking for, they will hire him immediately. [chuckles] They always just write a nice wish list for Santa, but nobody knows all these things. In the end, Googling and, I would say, thinking on your feet – this is usually what gets you.

57:57

Alexey

**I remember my first job. Actually, I was already, so for my second job – I was a Java developer. I opened like 20–30 different job descriptions and then noted what is common among them. Every single one of them had a tech (piece of technology) that no one else had. And then like… [cross-talk]**

58:21

Nemanja

There’s a data science project! Make a web scraping application that scrapes LinkedIn jobs, or Monster.de or whatever you're using, and do some kind of NLP and analyze [the requirements]. And make a website out of it and you can maybe earn money out of it. People can then follow the daily trends in the tech stack per domain. You say, “Okay, for a machine learning engineer, these are the most common keywords (excluding the stop words, of course [chuckles]).” And yeah, there it is. We just gave you an idea. Perfect. [chuckles]

58:56

Alexey

**That’s cool.**

58:57

Nemanja

Yes, when you earn money, give us a cut. [chuckles] We have the recording.

59:04

Alexey

**Okay. I think that's all we have time for today. We should be wrapping up. Thanks. It's always a pleasure talking to you.**

59:13

Nemanja

Likewise.

59:15

Alexey

**Thanks for joining us today and sharing your experience with us, giving all that advice. So thanks a lot. We're looking forward to connecting with you again, either in person, in Porto, or some other conference or elsewhere. Yeah, thanks. And thanks, everyone for joining too. Bye.**

59:39

Nemanja

Ciao!