Blue – Interviewer1, Purple – Interviewer2, Green - Interviewee

Let me mention again that you agreed that you're going to be recorded and that your interview will be transcribed and shared.

Yes, okay.

Ok, so let's start with general questions. So we would like to ask some background information. Could you tell us what is your current position at your current job?

Okay, now I'm working in Digital Innovation Lab as a Data Scientist and Machine Learning Specialist.

Okay, where do you work?

In the [removed for anonymity].

How much of overall work experience do you have?

In Data Science it is like 3 months, but in general it's about 2.5 years.

Okay. So, in Data Science it is 3 months as you said.

Yes.

So, can you tell us what type of Deep Learning networks have you been working on? Like is it supervised/unsupervised/reinforcement learning?

I have been using supervised learning in my current project. We are trying to, it is like a probability of default, the project’s name is the “Probability of Default”. It is like to predict how the customers will treat their loans. It means will they return their loans on time or

will there be any delinquencies.

So, sorry are you trying to decide whether you should give loan to people? Or are

you just guessing some probabilities on….

So, we have some probability and we have some threshold and if it is above threshold it means that it's not advised to give this customer some loans. If it is below this threshold then it is ok.

If the probability of them not returning the loan is above that threshold, right?

Yes, we have some probability and we have some threshold and we just compare this probability with our threshold.

Okay. And which programming languages or frameworks are you using for that? Python, Java, Keras, Pytorch?

We use Python in general.

And which framework do you use? Tensorflow, Keras?

We use some libraries like pandas, numpy, pykit and so on. It is not any framework in Python, just general libraries.

Okay. So what we would like is to ask you just one general question, which is what kind of bugs and problems have you faced so far while doing this project. So it would be great if you just, you know, come up with the problems that you had and just tell us about them.

Okay, the big problem was gathering the data because in our bank data was not saved in a good manner. I mean we have a lot of tables, a lot of DWH, I mean data warehouses where you should connect and gather data, and a lot of important features were empty. And that was a big problem and we tried to manage and handle this problem, I don't know predicting something, finding the mean of the data and filling it, or removing some outliers. So it was the biggest problem in this project.

Okay, so your data was distributed in different ways and you had to put it together.

Yes, and it was not enough for a good prediction. I mean the important features were empty in general. And the biggest problem also it was that the salary in our country, the legal salary, which was stored in the database was in a little range. I mean it's around 200-300 and when someone with 1000 salary will apply for some loans our model predicts like someone who has 300 salary. I mean there is no big difference for us like 300 or 1000. I mean it is a little range for some features.

So because your training data consisted of people who get 200 or 300, and when someone with 1000 salary applies, what does your module do in that case?

It's like treating like someone who has a salary of 300. I mean we have so few cases like this that it is like an outlier and we clear data like this. Our general range of salaries is this small.

So, did you do anything to remove this? Like to make it to predict better for 1000 or did you leave the model as it is?

Actually, we could not do anything for fixing it all, because it is the real data and it was like loans for two years in our bank and we cannot do anything with this. It is the data we have.

Okay. So after you have trained your model and your system is making predictions do you add this new people who have applied, like this person with 1000, do you add it to your training data and retrain your model?

Currently, no. We just predict, give some probability and that is all.

So do you know at your workplace, if they're planning to do that?

We can do it only after when we give them loans and we'll see their attitude. I mean it's like someone has applied for a loan. Ok, we say we can give you the loan. But we do not know how they will behave in the future. And we can add this data only when the loan ends and we see whether they had any delinquencies or not. Just applying for the loans does not mean that we can add this data to our training data.

Is it in a way will be the evaluation of your model? Like if you have predicted that they will return and then they do not it is gonna kind of either confirm or deny your predictions, right?

Sorry, I did not get you.

Let’s say you have given a loan to a person because of the prediction of your model.

Ok.

The way to test you model is to see whether these people will return the loan in the end or not.

Yes. There is a restriction I should have noted. We only consider little amounts of loans, not big amounts which has more risks.

So the model decides loans for a limited amount of money.

Like, less than 10.000.

The ones that are more than 10.000 are decided by your experts in your bank?

Yes. It is like a category for loans, risky, not risky, it depends on amount. So we first started with the small amounts.

So, you told us that you pre-processed data before using it for training. Can you tell us about the pre-processing actions that really helped you to improve the accuracy of the model?

Yes, we used the oversampling method.

Ok.

It is like reproduce our data. I mean, we can improve it like two times by oversampling some data. And we used some Python libraries for it. Yeah, oversampling helped a lot.

So, oversampling was useful for you and it helped to improve accuracy twice you said?

Not the accuracy, the amount of data. The accuracy improved like 3-4%.

Ok. Could you tell us about the problems related to training data that you faced? You said that you did not have enough data and used oversampling. Did you have any data that you found out that it’s content is somehow wrong and you had to correct it? Or no?

Okay. Well, it was not like that. First we created our model based on our features. I mean, we select some features which are things that have a contribution for our model. We run it and we had a good accuracy, about 80-85 percent, but then we discussed our features with other departments in our bank and they said that it is not a good way. I mean, they said it is a big accuracy, it is not the real number and that in real time it doesn't work in that way. And they say it has to do with the features we have selected. And they added some other features like branch codes, I don't know product category, loan's category. And for the second some we tried to manage our model like under their requirements. And now we have this probability, as they say that it is a more real probability.

Okay, could you tell which kind of people you talked to? Like loan experts?

Yes, we speak to loan experts and to a lot of other departments which were related to this project.

Okay. Yes. I have a question. Do you remember any like bugs that were associated with training data, when a program crashed, so you're getting some messages.

Mmm, let me remember. We had a lot of bugs regarding the type of storing our data, like persons who store it. I mean someone applies for loans and some bank expert saves this data and this data is not stored in a logical way. I mean, it was not, how to explain, like we have some column in our database and we have to split it and catch some data in the whole general picture. I mean they store data like one line and we have to split and catch it and a lot of and we have a little bugs regarding this because they don't save this data according to some rules. They just wrote some comma or dot or some slash and you have to predict when they store in which way and a lot of problems were regarding this problem, this manner of having data not in the right way.

Don't you have like a standard forms to save this data? Why?

No. Actually, we have, but it did not work back then.

Okay, so, okay.

And we have a big problem regarding delinquencies.

What?

Delinquencies, like when someone doesn't pay at time.

Okay. Okay.

So the problem was that they have a big table with all loans till 2016. And you can select only one day. I mean, we have some transaction date and we generate report for all loans in our bank for this day. I mean for now some of them have some delinquencies and someone has some debts and so on. Did you get what I mean?

I mean, you said that you generate transaction on some specific day.

Yeah, like we have some day and we create report for this day, current date or some particular day. And this table shows that how much someone has debts for this particular day and we have to select all the time. Like, if we want to create a report for one month, we should select for 30/31 times. We have to send a lot of select statements for collecting a reasonable data.

So, I understand correctly that you cannot get data for a period of time. But that is only for a specific day.

Yes. it was regarding this problem.

This is a problem for you. Why?

In our current model, for example, when we give a prediction, you consider how many debts they have in the last one month or if we have to collect a lot of data not such for one day and we consider behavior for some period and we have to give a lot of select statements for that.

Do I understand correctly that because of this structure in the system it is hard for you to collect training data, it's rather expensive?

Yes. It's expensive and time-consuming.

Okay, anything else you would like to tell us about the training data?

No, okay, actually, no. I said all of our problems I think.

Can you tell us a bit more about how you were testing your model? So you have some amount of training data…

Yeah, we just split our data for training and test.

Okay.

First it was 80 to 30, like 80% was for training…

80 and 20?

Sorry, yeah, okay, 20. But when we first started just testing our data, we actually wanted to improve our accuracy and we just gave a lot of data for training, not testing. Finally, it was like 85 and 15.

So when you use this ratio, you get the highest accuracy?

Yes, we also have some confusion table. We just like when we predicted [inaudible], you know, maybe about testing on real data like we didn't predict anything from other sources and something like that.

Okay, so when you were using just 80% of your data for the training they accuracy was lower it when you use more it increases?

Not so much, but it depends.

So, how much did it improve?

Up to 1 - 1.5%, not so much.

Okay. Okay. So maybe we can talk about the structure of the modules that you have been using like the layers and etc. Did you come up with the model structure yourself? Did you use something already used?

Actually we tested several models. Such as, SVM, MLP, XGBoost and Random Forest and we just compared their accuracy and selected the best one. In our case XGBoost was the leader. Random Forest and XGBoost were almost the same.

Can you tell it letter by letter, the XG something.

XG, B, double O, ST.

Okay.

It's like a random forest and XGBoost were like on the same range, but XGBoost won for our model. I don't know, it's hard to explain why. Actually, the worst and the best models didn't have a lot of differences. But we compared and selected the best one.

Do you have a number? How different were their accuracies?

Now not, but I have some presentation slide where we just show it and if you wish I can find it.

Do you remember any bugs that you had while using some particular models? Like wrong input or whatever?

Actually we are a team and each of the team members just worked on some data, I mean on some model and my case was SVM. No, actually it worked nice and easy.

Okay when you were using your final module so you took its implementation from somewhere. Right? Did you change it at least like something in that implementation or it just remained exactly the same?

I only changed some model parameters, like they have some model parameters like gamma function, alpha, it is just technical stuff.

Technical stuff is what we want to hear about. So if you remember how you changed that parameters and why that would be very useful.

Actually. We have a grid view searcher. It's like a bit of code where way you just type some parameters and they choose the best for you. There are a lot of, ahh ok, how to explain. So it's like a grid view search, maybe you know.

Is it an algorithm?

Yeah, it's like algorithm which where you just in input you give some parameters for gamma function. For example, like 0.1, 0.01.

Do you give any parameters?

It's from 0 to 10, for example, okay, it's intuitively you should understand how it works. And they run your model on each of these parameters and give you the best for you, the best model parameter.

Okay, it picks the parameters that are best for you. So did you do anything about the number or types of the layers in the system?

Layers? What kind of layers?

I'm not sure maybe the model you have been using doesn't have this. Yeah. Okay, so, did you do any hyperparameter tuning, like batch size, activation functions, things like that?

We just added some features like I mean if you have some income and we know the duration of the loan, we can divide these two features and add some new number to our training set. We tested to improve our accuracy like this, adding some new features, but in our case it did not help a lot, so in the end we deleted it. We also used square, log, power of some parameters, but it did not work in our case.

Okay. Can you tell us like from how many features you started and how many you ended up using in practice?

Let me count.

In our case, it was gender, age, marital status, employment type (employed/not employed), salary, credit amount, delinquencies in the last 50 days. Ah.. Sorry. Ok, did they have any loans before? Did they have any delinquencies for these loans. Actually, these are our features that were selected for our model.

Okay, you remember any that you have removed and why?

Yes, we had some DTI feature, it is like the ratio of your income to the credit loan. Yeah, this feature, actually it was important in our first model, but then our colleagues adviced that it's not working, just forget about it and we deleted this feature. Also we have some branch code feature.

But you said that it was important in the first model…

The problem was because it is the same as the salary, the range of this DTI feature was very small. It's like 3.8-3.9. Okay, we never got 3.5 or 4, for example. And it's important in real time. But in training, it doesn't help. Oh, it's not training our data.

Because it is in a way proportional to the salary, right?

Yes. And actually this amount comes from [organisation: removed for anonymity]. It's not like some ratio, maybe it is some other feature which I don't know. We just request this feature and we get it from [organisation: removed for anonymity]. And because of the small range and it doesn't help, it doesn't have training power for our data.

It's not exactly clear. So the ratio of the salary to the amount of loan is something…

It is our assumption. Maybe it is not in that way. That's why I say that we do not calculate this feature. We just request it and get it from [organisation: removed for anonymity].

So you tell this organisation that we have this customer who has this salary and wants this loan?

No, we do not say they have some salary this time. We just say we have such a customer and based his personal data, I don't know how, they just give this DTI feature to us.

Okay.

Yeah, I guess it is some ratio to income.

Sorry, go ahead.

Okay. I mean it's my assumption that they take into consideration their salary, the loan, blah blah blah, but I don't know the exact formula, how it is calculated.

Okay. So [organisation: ] calculates it per person for individual person.

Yes, it is individual for current date for individual person. It can change in the future. I don't know how they manage it.

Ok, you realized that it is not useful for your training later.

Yeah. Also, we have a feature like branch codes.

Branch codes of the bank?

Yes.

How is that one useful for anything?

It's a little bit strange, because maybe, for example, in some area people tend to return loans on time. And for this branch code we have a positive assumption, it's like wow. But for some areas in Azerbaijan, they are not tending to return loans on time. But we just decided like okay, if we take into consideration our branch code and if I am a bad customer, I can get to some area where branch code is good. So, probably they will give me this long and it's not good.

So this raises the question, do you do some kind of fairness testing because it gives an implication that if you're from a poor area, you're more expected to be refused a loan. So do you do any kind of this testing where you check by gender by either…

Yes, of course. We did it in all stages of our model.

Did you improve your model somehow based on this kind of testing?

Testing was more related to feature selection and regarding this results we just decided to take this feature in our model or not, and we were always in touch with the other departments. And the biggest problem I should also note, that data in our bank for customers is saved in the more or less good manner from the beginning of 2018. So we only have one year and a half in our training set here. So it goes all over the problem, yet because our data was not so powerful, we always were in touch with the other departments to share experience. I mean, we just ask them how do you think real choice is made.

You said that you used this kind of testing for feature selection, meaning that if let's say changing one parameter as gender makes your system to give a different result you use that feature or that you do not use that feature?

It depends on how they improve our model, how much and again..

So the aim of this testing was not about ensuring that people are not discriminated against but just to improve the model, do I get it correctly?

Yeah, of course, we don't discriminate against anyone.

Okay. Where do you train your models? Do you have some specific machines to do that or now?

We just do on our computers the training.

So do you get in any trouble when training the model. Like does it take long? Do you get errors?

Yeah, sometimes it takes a long, like we run our program when we leave our work space. We come at the morning next day and it is still running. Actually. This problem was also in the graphical part. When we were pre-processing our data, we start to draw some graphics to understand how it managed the ratio of our data, how it worked to predict prediction horizon, to predict our threshold. And I mean we used some plotting blah blah blah blah in Python and it took time, a lot of time to draw such graphs and bar charts and something else.

But it is doable on your standard work machine.

Yes.

Do you use Windows machines?

Yes.

Okay, so you never encountered any problems related to hardware.

If time consuming is problem, then yes we faced it.

So, how long does it take approximately to train your model?

You mean the whole project or just training part.

Just training.

Because before training we have a lot of stages like, processing, graphical blah blah blah and finally, we start to imply some models in our data and it took like in general with testing and so on like one month or a little bit more.

No, this is how long it took you to develop the model.

Yes. What do you mean?

I meant that if you leave it running on your machine how long…

Ahh, okay, hard to say.

Okay, but is it hours or days?

It's hours. Like 12 maybe.

12 hours. Okay.

Approximately.

Any other problems maybe? Like maybe you gave your model too much data. Did you face something like you could reduce the data size, but if you can get the results and do it fast.

You know, in our case we just want to grow our data, not reduce because I say again we have we don't have a lot of data.

What is the approximate size of the training data, like for how many people do you have or how many long history do you have information there?

Like after all pre-processing blah blah removing outliers we have some big table, dataframe in Panda in Python. It's like 30.000 rows in our table now.

And usually in such applications in this field, what is the normal size of the training set?

There is no normal size, you have what you have. You have some data, you have some loans, you have some tables. It does not depend on us how they much data we will have.

No, no, we get that. We were asking like in other Banks, like in other places, which use similar algorithms.

It's hard to say because I don't know actually.

Okay. So now, anything else you remember about problems. Technical bugs, crushes you face during development?

That case scenario when everything is so bad, haha.

Like anything you can remember, even the small things.

Actually, the biggest problem was at the beginning when we tried to explain to people what we are trying to do and why we're doing this. And they don't rely on us, like okay, of course, you will, sure. Ahh…

Did you have a process where your model, like you have a real human who says whether you should give a loan to this person or not. And then you have the prediction of your model. Did you do this kind of testing to see if human decisions and your model predictions are in line?

Okay after ending, we are just ending this project now and we are creating some API for it, like we have some web page where someone can. It's only for local use, I mean it's not published yet and we are just testing on our cases. For example, I just enter my data and some other cases and see if it will give me a loan. Then I change and see how it changed depending on what I put.

The prediction of the model is changing, right?

If I change some parameters, of course, it is changing.

No, no. But what I was asking if you have checked whether the predictions of the model and the prediction of a loan expert, for example, are these…

I mean we are testing in real data. I mean, how can I get someone who will apply for it?

No, no, I was just guessing that your model is on production now.

No, it's not on production now.

Okay.

In two weeks, for example, I can give you more information.

Okay. I see. Is there like the most interesting bug or the one that took you a lot of time to fix? Do you remember like one specific bug which was very painful for you?

Haha, let me remember a painful bug. All bugs are painful.

Tell us about some, then.

So, give me time, please. [Pause]. I had some bugs, of course, but I fix it like in 2-3 hours. It wasn't some global bugs, environment stuff, language stuff, version problem, blah blah blah blah, but general bugs in programming.

Which problem, sorry?

I think I said that its general problems in programming. Not related to our data.

Yes, tell us about it.

Yes, tell us about this general ones.

Version problems, I don't know setup environment.

Okay, tell us more about what kind of version problem did you have?

I had a big problem to connect with our database through Python. I don't know why.

Which database management system are using? If it's not secret information, of course.

Oracle SQL.

Yeah, and you had a problem connecting it?

I had to add some environment variables in my Windows, like adding some environment variables and I downloaded some Oracle client on my machine. Extra software which I should download and when I understood it, I just downloaded it and it solved my problem.

Okay, then environment setup, you said. Was it about these variables or was it something different?

Sorry, what?

You said that you had problems with versions and then setuping your environment. Is setting an environment part was also about this Oracle SQL connecting or something else?

No, I just missed some Oracle client. It's like you should download it and run it on your machine. And then you will be able to connect your Oracle SQL database from Python. In my case it wasn't that way. So, just it was my first experience in Python and I just tried to download some Jupiter notebook. For example, I'm working on Jupyter Notebook and I should take some virtual environment in my local machine. It took a lot of time for me, maybe for experienced ones it's not so strong or not so time-costing, but for me it was time-costing. I mean, it's a new field and everything you do is like a new thing and you have to make a big research on it. I mean programming, looking for some video tutorials and so on, and then you are able to something valuable. Ok, but some big problems…

Not essentially big, maybe even the small problems but related to the model and machine learning part of your application. General, small ones, anything.

The biggest problems for the first time was that every time when I run my model, I have not the same prediction and it was confusing. I didn't change anything, the data is like the same data. The model is the same. It was regarding about some random state in the model. And you should have some run random state variable in your model and it should be fixed number, no matter which number, you should just declare it in some number. And in that case, when you re-run your model, you will have the same prediction.

How different were the predictions when you did not have a fixed random state?

Actually, a lot. Sometimes I had a difference like 10%. It's so much, actually. So you should add some random state to your model, otherwise, it does not work. I mean it isn't informative when you run and have some other prediction. That's all maybe.

I have one question about the input to your model. So you told us that you have these features that you have selected. How are they passed to your model?

What do you mean past?

Do you give just vectors of input to model?

Ahh, okay. We store our data in dataframe in Pandas. It's like Python library and we just extract this data from the dataframe. It's like it's like table column in our way.

Okay, um, output of your system you said is a probability. So it's a number between 0 and 1 right?

0 and 100.

And 100. When you're training, I guess when you're training the person has either returned the loan or has not. So it's either 0 or 100. How did you decide on the value of that label when training?

Okay, we have some probability, like 40% and we have a threshold.

No, no, I get that your models returns this thing and you have a threshold. But when training your model you keep all the personal information of this person and then give information on whether this person has returned the loan or not?

No, we input delinquencies. How many days this someone does not pay this loan on time. Like the day number.

So when training your label for the person is the number of delinquencies?

Yes.

And based on that the model gives a probability?

Also we have some threshold on those days. I mean, for example 20 days is for us, but 21 it means it is a bad customer and we just labelize, like if these days are less than some threshold, like 1, if greater then 0. We labelize some features.

So this is the decision-making that you're guiding your model towards right? So how did you decide on this threshold? Did you try with different ones? And did you see they will…

We used the prediction horizon, when we had to decide. It means that we just draw some dataplot regarding on data and this plot shows when the delinquency started. I mean you are a customer, you apply for a loan. You have some loan , first three months you pay in a good manner. It's okay, you pay on time, but after for example 3 months you start to have some delinquencies. And plotting our data in some graphs, we will find the best case. I mean from which days customers start to have some of the delinquency and we select these days [not very clear] like our prediction horizont. I mean our models give some prediction for three months. I mean if you're applying for a loan now and I gives the probabilities, this probability is not for tomorrow. It's like for your behavior in few months. Do you get what I mean? I give a prediction for the future, like three months later, for it depends on our model. It was a prediction horizon. And your question how select the threshold for good or bad, mostly just find the ratio for all customers and making our model more... I mean we try to have a balanced state, I mean we select such days when the number of good customers and best customers are equal.

Okay. So, you had an unbalanced data.

Our data now is more balanced than unbalanced.

Okay, but it was unbalanced. Did you see any problems with that?

Yes, of course, we tried with unbalanced and our prediction was low. And we decided....

So the data was unbalanced, the accuracy was low.

Yes, yes.

You balanced your data so that you can improve the accuracy.

We have selected all our thresholds in that so that our data is in balance. I mean good customers, bad customers, approximately like if you have 52% of good customers, then 40 something bad customers.

Do you try to also balance in terms of age balance, gender, salary range or not?

No. No, we didn't have any thresholds for these features. Age, salary does not depend on us. We have thresholds only in prediction horizon, in the delinquencies days and for probability of default. Three features when we have thresholds.

Ok, good. Did you remember any other bugs that you have faced?

Oh my God. Okay, do I have five minutes for checking this kernel and I will remember maybe something.

Yes, of course. You can check, anything that will make you remember about a bug is good for us.

Okay. I actually try to forget bad things.

[Pause]

At the end I don't have anything else, sorry.

Okay, so thank you, thanks a lot.

Thank you.