

**User:**

Hi, nice to meet you.

**Assistant:**

Hello! It's nice to meet you too. Thank you for taking the time to speak with me today about the Entry-Level Machine Learning Engineer position at G-Research. To start off, could you tell me a bit about your experience with machine learning, particularly in areas like model training and optimization?

**User:**

Yeah, sure. I'd be happy to. I've got some experience in model training and optimization. I've mainly acquired this experience throughout my time at university and through competing in various competitions on Kaggle. Some of those included regression tasks where I trained extra boost algorithms to predict the age of abelones, whatever that is. I've no idea what that is, but yeah. I mean, it didn't really matter to me for the competition, but yeah. Some other ones that I've been involved in is some other stuff, yeah. I get fired immediately. I get sent out.

**Assistant:**

I see. Thank you for sharing that. Could you elaborate a bit more on your experience with specific machine learning libraries or frameworks like PyTorch or NumPy? How have you used these in your projects or competitions?

**User:**

To be honest, I'm not very sure.

**Assistant:**

I understand. Let's move on to another aspect of the role. The position involves working with large datasets. Could you describe any experience you have in handling and processing large volumes of data, particularly in a machine learning context?

**User:**

Um, I've never actually done that before.

**Assistant:**

Thank you for your honesty. One last question: This role requires strong programming skills, particularly in Python. Could you tell me about your most challenging Python project and how you approached it?

**User:**

Yeah, sure, I'd be happy to. I think the most challenging Python project that I've ever done was involving the implementation of NEAT, which is NeuroRevolution of Augmenting Topologies, into a game called Jump King. In a small group I led, we had to essentially implement NeuroRevolution

into Jump King. This was quite challenging due to the codebase that we began working with being an open source clone of Jump King, built entirely in Python. So we had to understand the basics of Pygame and how Pygame worked and then work around that to be able to implement NEAT into the game. Implementation of NEAT requires many, many different agents in the game itself. In this case, the individual player character Jump Kings, we had to modify the game's actual architecture to allow for several kings to exist on the screen at once. Once we did that, we then had to tie in the interface with which each king is connected, which is the keyboard inputs into the outputs of neural networks that we had defined using a package that we use called Neat Python. From here we had to... If you just give me a moment to think, sorry. From here we had to then... I'm going blank. I didn't hear what you said. Yeah, no. Okay. Well, oh well.

**Assistant:**

Thank you for sharing that experience. It sounds like an interesting project. I appreciate your time today. We'll be in touch soon regarding the next steps in the application process. Have a great day!