1. In recommendation problems the real-world data varies overtime, the larger number of subscribers get the larger the spectrum might be for a given recommendation (assuming it's a item-based recommendation) also in the user-based, users don't always have the same interests over long periods of time, also there might be seasonality/trends in the data that changes over time.
   To check for this they might try to take a new sample from the live data and "refresh" the model training and testing data and check if the accuracy is any higher; If so, then I think the best solution for such problem is to have some kind of a pipeline that runs to feed the model frequently with new data and deploy the new model weights. This should be enough as long as the architecture doesn't underfit the data.

2. Probably what happened is a very popular problem with GANs, "Mode Collapse" it's when the generator produces small range of outputs to "fool" the discriminator so that the generator loss is decreasing and the discriminator loss increasing over time because the same trend occurs the discriminator error is getting higher so the generator updates itself in the same direction at that point it has already lost the ability to generate different outputs. So the training needs to be restarted, and they should be monitoring the llosses there has to be a 'right' balance between the losses of the two where we would want **both** losses to be kind of stable in relation to each other also maybe attempting to experiment with the data dimension would help a lot in such situations.