**Case 6**

In this case one option for monitoring if the model still predicts okay, would be to have a null model as comparison. It could be something like randomly predicting hobbies and comparing the results with the model suggestions: if the model suggestions are as good as the random suggestions, maybe it is time to retrain. This requires tracking and logging the ground truth, whether the user added that hobby to their hobbies for example. In general, this requires a sort of shadow deployment, where a candidate model is trained in parallel to the existing model.

As mentioned, useful thing is experiment tracking, especially tracking and logging some model performance metrics, especially the ground truth. Granted, the variability is high in the ground truth acquisitions times, but it would still be worth tracking, because this is the data that you can check the error rate with.

It would also be possible to use a version of slice-based evaluation in the background, training the model with more recent data, which you have gotten the ground truth, versus the older data, and compare the accuracy-rated metrics. If possible, it would also help if the user feedback could be collected. It could be rating the suggestions with a scale or just upvote/downvote. Then the pipeline could include mechanics that transform the acquired new metrics into logs and retraining parts of the pipeline. The pipeline could include alerts for the engineers or automated tracking systems that alert when a new hidden model performs better than current deployed model.

When deciding when to re-train the model, the system needs again good monitoring and tracking of the predictions and means to evaluate the distribution shifts and the pipeline mechanics to alert or automatically retrain the model with more recent data. The pipeline could include two-sample tests and statistical methods to test whether the distribution has shifted. Standard drift detection methods include DDM, which monitors the error rate of the predictive model over time, and ADWIN, which monitors the response distribution within a dynamic sized window.

Usually, when dealing with this kind of applications, you get new data all the time. It is benecifial then to have assessment methods in the pipeline, when to retrain with the newer data. One metric to use is a learning curve, where you draw a plot of the model’s performance over time, for example the training loss or accuracy, against the number of training samples it uses. This would help us determine whether we would gain any performance gain from more data. It could be done also with trying to determine if we need to use only more recent data, or does it help to include all the historical data, resulting in more training data. The pipeline could then transform the plots to a monitoring dashboard for example.

The ground truth is determined to be whether the user updates the recommended hobby to their profile or not. Alternatively the ground truth could be determined to be the user feedback of the suggestions, such as a upvote/downvote feedback, or even rating the suggestions from 1 to 5 starts for example.