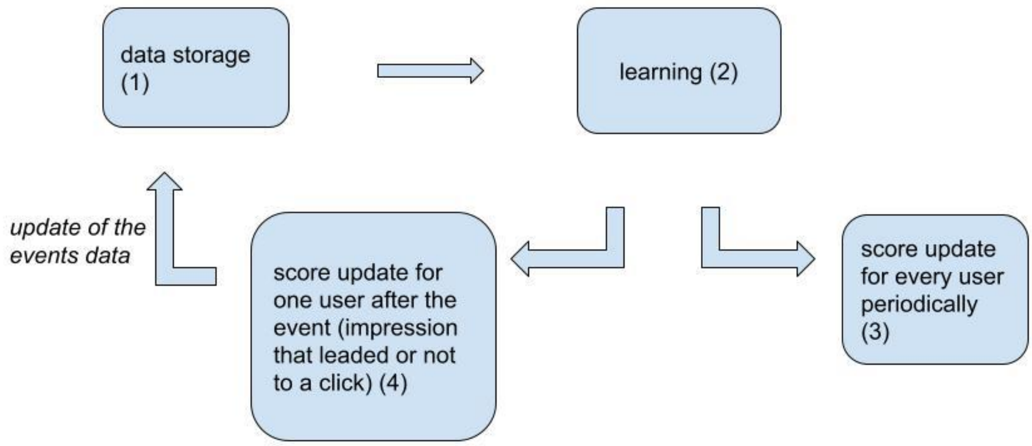
2 - Maintenance considerations (e.g. automatic retraining)

Depending on the business use case, approaches for retraining a model include:

Periodic retraining: In this approach, the model is retrained at a time interval we specify. Periodic retraining is useful when underlying data changes within measurable time intervals. However, frequent retraining can be computationally costly so determining the correct time interval is important.

Trigger-based retraining: This method involves determining performance thresholds. Models can be retrained automatically when the model's performance drops below this threshold. This is why the choice of metrics are very important.

3. Design deployment in production



In (1), we can use to store the data:

- a structured DB (if needed, for querying it)
- Column oriented file (parquet) to partition and select only the dates corresponding to the duration on which we want to learn

- Possibly a row-oriented file (csv), on the condition that they are separated by date to avoid having to load the whole file

The files of the last 2 points can be stored in a cloud storage solution (s3 with AWS for example).

In (2)

This is a server where the learning task is performed.

In (3)

After storing the model (in a pickle format for exemple) we use it to update all the scores periodically.

In (4)

The idea is to be able to run the model (predict only) for a single user and not in a batch like the step (3).
The sine qua non condition for not deteriorating the customer experience would be that the program is sufficiently optimized to run quickly.
We can use a serverless service that acts as an api, which takes a user id as input and returns the result of the recommendation as output.