Machine Learning has been added to Tiki23 as a built-in (but optional) feature, using the Rubix ML library.

How can this be used? It opens up so many possibilities!

- To get an idea of the diversity of the use cases, look at the “Tutorials and Example Projects” section here: [https://rubixml.com/](https://rubixml.com/) (Yes, we will now be able to do this in Tiki!)
- Machine Learning will be deployed all over Tiki to leverage both system data (ex. logs of user activity) and content managed by Tiki users, like Trackers and Webmail.

**Video**

**Code**

- Tiki23 will be released in June2021. But you can use and test it today via a Daily Build. After sufficient testing, the machine learning code could be backported to Tiki22 (ex: in 22.3 or 22.4) given that it's pretty self-contained (if you don't activate the feature, nothing changes)
- Here is some of the code:

**How to configure**

(better documentation is being worked on)

*Victor* wrote:

**Quick summary of how it works:**

1. Enable ML feature in admin features.
2. Give permissions to admin ML models to relevant users and use the models by (other) users.
3. Go to ML models (new menu entry when feature is enabled and you have permission to see it).
4. Start with a new model. You can use MLT template for bootstrap.
5. Select relevant tracker and field. Depending on the model, you can select more than one field but MLT works best with one field at the moment.
6. Check model creation process - you can choose from basically all Transformers from rubix base and extras packages, then specify a learner (regressor, classifier, etc.) then tweak the parameters and save.
7. You can test models at every time against the real data.
8. All exceptions/errors are shown during params tweaking and testing/training.
9. If data source is not big, you can train via web interface.
10. Otherwise, it is better to use the new console command (ml:train) and do this with a scheduler - e.g. train every night.
11. After model is trained you can use it - with MLT that consists of entering query content and seeing top 20 (by default) relevant matches. All parameters are tweak-able including this 20 more like these entries.

The MLT template uses the best approach we came up with Andrew up to now. If we figure out something better, I can modify the default template. The plan is to add such templates for each use-case we handle but also give people the opportunity to experiment with their own models.

We need help pages on doc site - Machine+Learning and Machine+Learning+Models. It is probably best to describe implementation details, have suitable links to rubixml docs and such things. Something important I see here is that trained models are stored as cached content on disk. Clearing Tiki cache will require to re-train the model.

That's in short. I suppose you will have a lot of feedback, so curious to hear how to continue/improve here.

One way to experiment is different source options - more than trackers, e.g. wiki pages, action log items, calendar items, files, etc. Files should probably give us the ability to upload a csv of a sample data to train against.