Twitter Thread by haltakov.eth ■ ■■

haltakov.eth ■ ■■
@haltakov

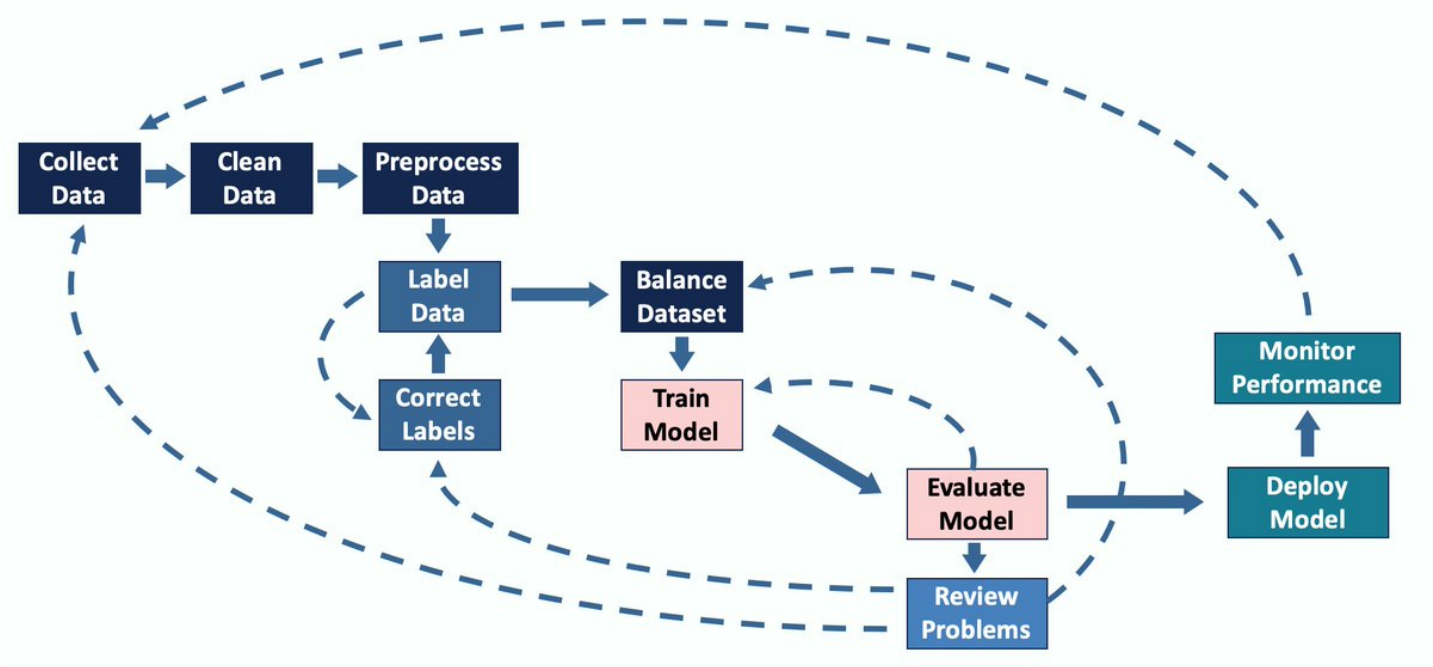
Machine Learning in the Real World ■ ■

ML for real-world applications is much more than designing fancy networks and fine-tuning parameters.

In fact, you will spend most of your time curating a good dataset.

Let's go through the process together ■

#RepostFriday



Collect Data ■

We need to represent the real world as accurately as possible. If some situations are underrepresented we are introducing Sampling Bias.

Sampling Bias is nasty because we'll have high test accuracy, but our model will perform badly when deployed.

■

Traffic Lights ■

Let's build a model to recognize traffic lights for a self-driving car. We need to collect data for different:

■■ Lighting conditions
■■ Weather conditions
■■ Distances and viewpoints
■■ Strange variants

And if we sample only ■ we won't detect ■ ■■■■

■



Data Cleaning ■

Now we need to clean all corrupted and irrelevant samples. We need to remove:

■■ Overexposed or underexposed images
■■ Images in irrelevant situations
■■ Faulty images

Leaving them in the dataset will hurt our model's performance!

■







Preprocess Data ■■

Most ML models like their data nicely normalized and properly scaled. Bad normalization can also lead to worse performance (I have a nice story for another time...)

■■ Crop and resize all images
■■ Normalize all values (usually 0 mean and 1 std. dev.)

■

Label Data ■■

Manual labeling is expensive. Try to be clever and automate as much as possible:

■■ Generate labels from the input data
■■ Use slow, but accurate algorithms offline
■■ Pre-label data during collection
■■ Develop good labeling tools
■■ Use synthetic data?

■

Label Correction ■

You will always have errors in the labels - humans make mistakes. Review and iterate!

■■ Spot checks to find systematic problems
■■ Improve labeling guidelines and tools
■■ Review test results and fix labels
■■ Label samples multiple times

■

The danger of label errors ■■■

A recent study by MIT found that 10 of the most popular public datasets had 3.4% label errors on average (ImageNet had 5.8%).

This even lead authors to choose the wrong (and more complex) model as their best one!

https://t.co/dfZPz6xnU0

■

Balance Dataset ■■

Dealing with imbalanced data can be tricky...

Let's classify the color of the ■ - we can get 97% just by learning to recognize ■ and ■, just because ■ is severely underrepresented.

I have a separate thread on this topic:
https://t.co/R8z3AeDD2b

■

> Dealing with imbalanced datasets \U0001f401 \u2696\ufe0f \U0001f418
>
> Real world datasets are often imbalanced - some of the classes appear much more often in your data than others.

The problem? You ML model will likely learn to only predict the dominant classes.

What can you do about it? \U0001f914

Thread \U0001f447

— haltakov.eth \U0001f30d \U0001f1fa\U0001f1e6 (@haltakov) February 10, 2021

Train and Evaluate Model ■■

This is the part that is usually covered by ML courses. Now is the time to try out different features, network architectures, fine-tune parameters etc.

But we are not done yet... ■

Iterative Process ■

In most real-world applications the bottleneck is not the model itself, but the data. After having a first model, we need to review where it has problems and go back to:

■■ Collecting and labeling more data
■■ Correcting labels
■■ Balancing the data

■

Deploy Model ■

Deploying the model in production poses some additional constraints:

■■ Speed
■■ Cost
■■ Stability
■■ Privacy
■■ Hardware availability and integration

We have to find a good trade-off between these factors and accuracy.

Now we are done, right? No...■

Monitoring ■■

The performance of the model will start degrading over time because the world keeps changing:

■■ Concept drift - the real-world distribution changes
■■ Data drift - the properties of the data change

We need to detect this, retrain, and deploy again.

Example ■

Drift ➡■

We now have a trained model to recognize ■, but people keep inventing new variants - see what some creative people in Munich came up with ■

We need a way to detect that we have a problem, collect data, label, and retrain our model.

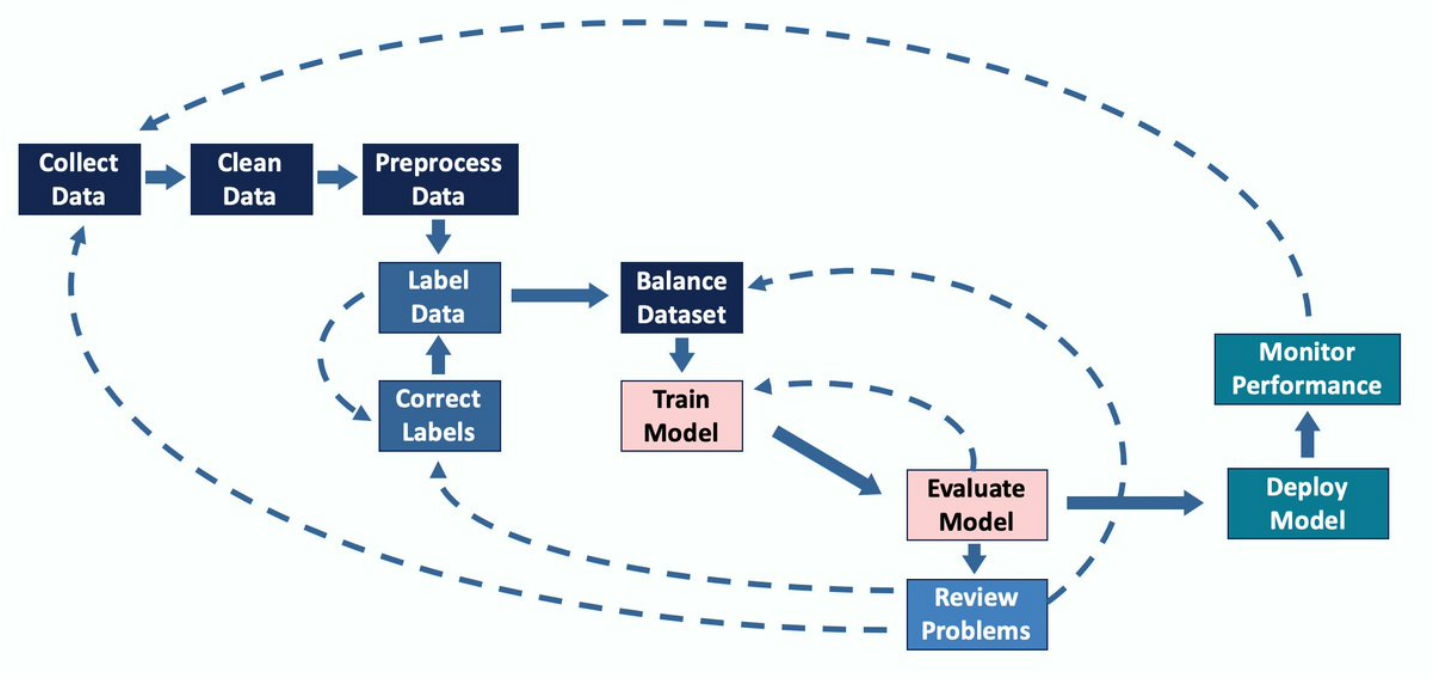
■



Summary ■

This is how a typical ML pipeline for real-world applications looks like. Please remember this:

■■ Curating a good dataset is the most important thing
■■ Dataset curation is an iterative process
■■ Monitoring is critical to ensure good performance over time



Every Friday I repost one of my old threads so more people get the chance to see them. During the rest of the week, I post new content on machine learning and web3.

If you are interested in seeing more, follow me @haltakov
.