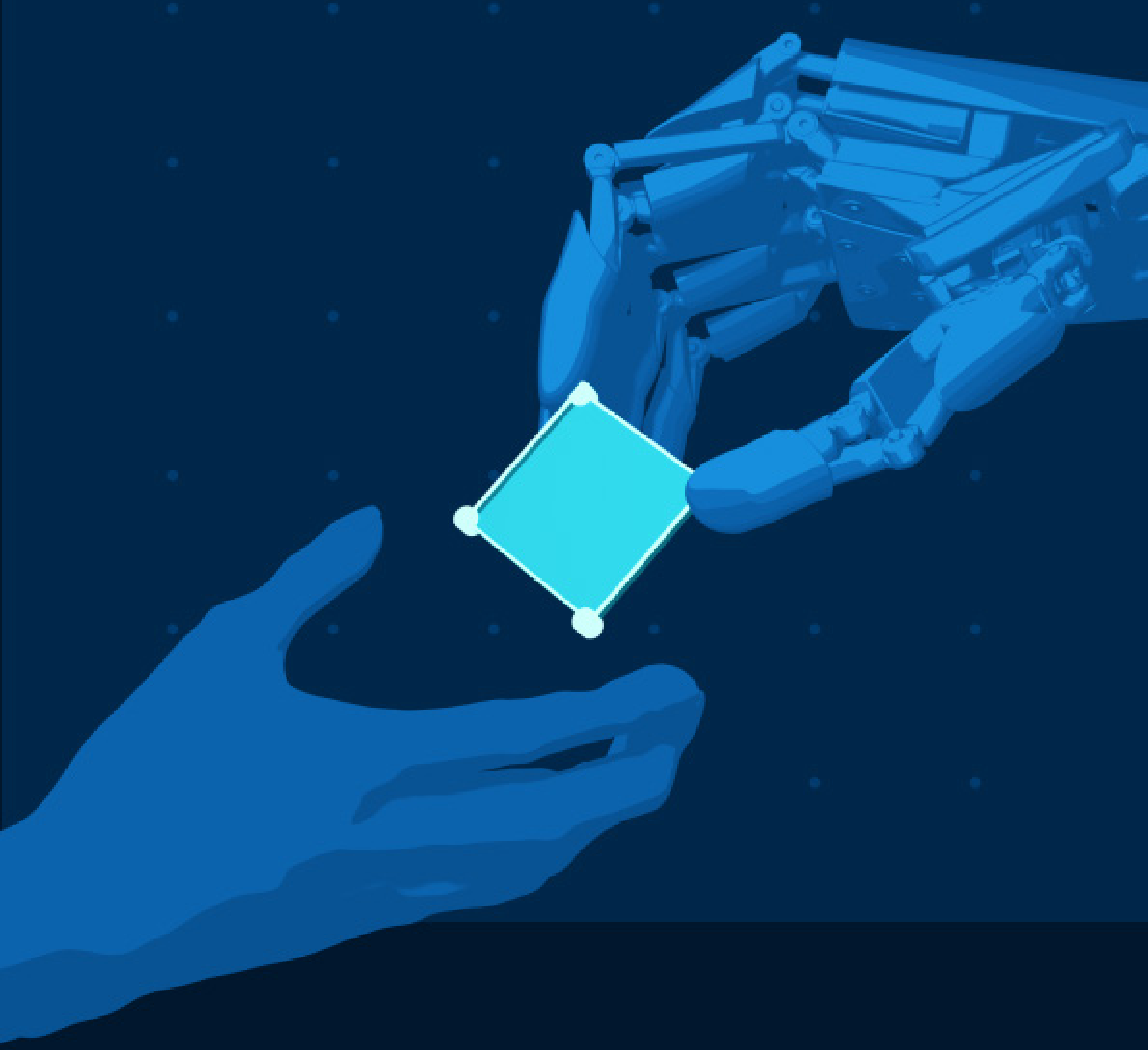


Labelbox

The Labelbox Guide to Data Labeling Automation

00 Content

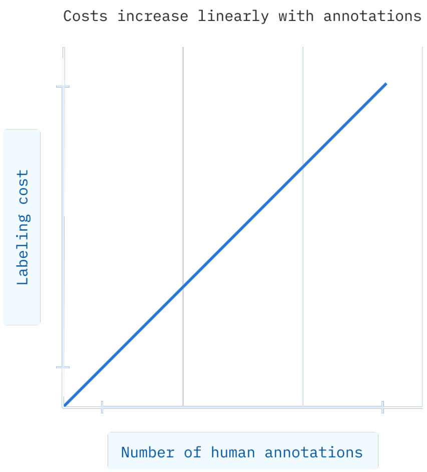
01 The case for data labeling automation

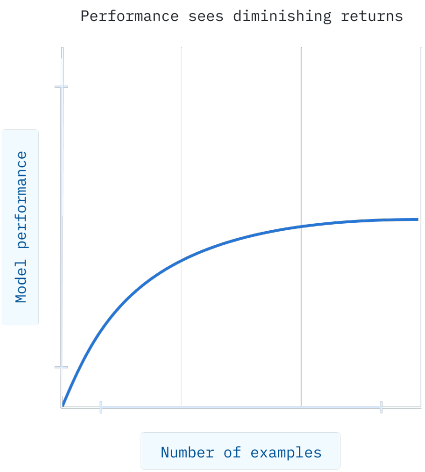
Developing and maintaining a performant service using supervised machine learning requires a huge amount of data. Training large models on large datasets creates several challenges, including:

1. Working with distributed graphics processing units (GPUs) and specialized hardware like tensor processing units (TPUs)
2. Figuring out how to run potentially time- and cost-intensive experiments to validate that your changes do improve model performance
3. Labeling the vast amount of data necessary to train the model — often the most time-consuming task in the machine learning process

This last challenge is especially difficult to address, as research has shown that models perform better when they are given exponentially more training data on a day-by-day, iteration-by-iteration basis. Training a model can take a long time, which means that iterations can then take weeks to complete, just because of the time it takes to prepare and label all the training data. A model's performance must increase continuously as it learns new information, so faster iterations are key to a highly performant model.

Machine learning projects are more likely to succeed when they iterate quickly as this allows teams to better identify and correct for any biases in datasets and, add new datasets as use cases expand, since changes in the real-world may affect previous distributions.





Labeling vast quantities of data quickly, efficiently, and accurately is an immense challenge, but advanced ML teams have found a way to cut both labeling time and costs with an innovative solution: labeling automation.

Labeling training data also requires a lot of human effort and expertise, so this part of the process is typically also the most expensive. As the chart below illustrates, costs increase linearly with the number of annotations, but model performance sees diminishing returns as the number of labels increases. For of example, if you can get your model to 80% accuracy with five thousand labeled images, you might need an additional ten thousand images to get the model to 90% accuracy.

Model training methods

Traditional supervision	Subject matter experts hand-label training data. This method is expensive and time consuming — but usually results in higher quality training data and a more performant model.
Active learning	Train the model only on the most informative data — usually, the examples that the model is having the most difficulty with.
Transfer learning	A model developed for a certain task is reused as the starting point for a model on a second related task or domain.
Weak supervision	This is a way to use lower quality labels. ML teams could: ■ Have non-experts create labels, which are then checked over by experts ■ Use heuristics to speed up the initial labeling ■ Use the work-in-progress model to pre-label data, which may then be reviewed by labelers/experts. This is model -assisted labeling, the most straightforward labeling automation method.

This guide will cover three commercially proven categories of automation and how each of them can help — and sometimes hinder — the labeling process. We'll also cover model-assisted labeling, a promising automation strategy that's already benefiting advanced machine learning teams across industries.

02 The three labeling automation methods

There are three general categories of labeling automation based on how much the method knows about the data you need to label. These three categories represent the most promising commercially proven methods to automate data labeling.

1. **Data unaware** methods have no model of reality informing its understanding of the target domain; they're usually just tools that help create segments
2. **Data semi-aware** methods have a basic understanding of the task at hand through generic, public datasets

3. **Data fully aware** methods have an application-specific model of reality informed by ground truths — that is, it knows exactly what your model needs to be trained on

There are myriad underlying algorithms that can be used for each of these three methods. In many cases, multiple algorithms are used together to increase the effectiveness of the method.

When we state that the data is “aware,” it refers to how much the underlying algorithm(s) “know” about the data and patterns of interest that the human labeler perceives and annotates on the data. In other words, the extent to which the method understands the end state of the labeled data, i.e. the ground truth label.

Data unaware

Data unaware methods typically automate the process of drawing segmentation sub-sections between contrasted color boundaries. The algorithms used for this method, such as graph cut, are unaware about what the data actually is, or what the labeler wants to label. It is entirely concerned with the numeric values of contiguous pixels. This type of tool can make the process go a little faster for your labeling team, but it might not be enough in cases of complex segmentation, where correcting automated segmentation can require more work.

Often, data unaware methods don’t actually speed up the labeling process in the long run because they don’t account for several important factors, including the skill of the labeler, the optimal segment type, and how well the labels translate to production data.

Here are some common implementations of the data unaware method:

- Superpixel
- Extreme clicking
- Watershed

The performance of the data unaware method varies depending on the nature of the data and how closely the labels align to contrasted regions. For a given data set, the performance will also vary on each data point — and this variance is completely independent of the sequence that the data is labeled. So the average performance of the data unaware method is static on a given dataset, which means that as more data is labeled, the performance of this method also remains static.

Data semi-aware

Data semi-aware algorithms are loosely structured for your use case. These tools typically use a generic or public set of labeled data (like COCO) similar to the data required to train your model to automate labeling for a new dataset. The labeling team can then correct or adjust the generic labels instead of doing all the labeling on their own — which saves them hours of time.

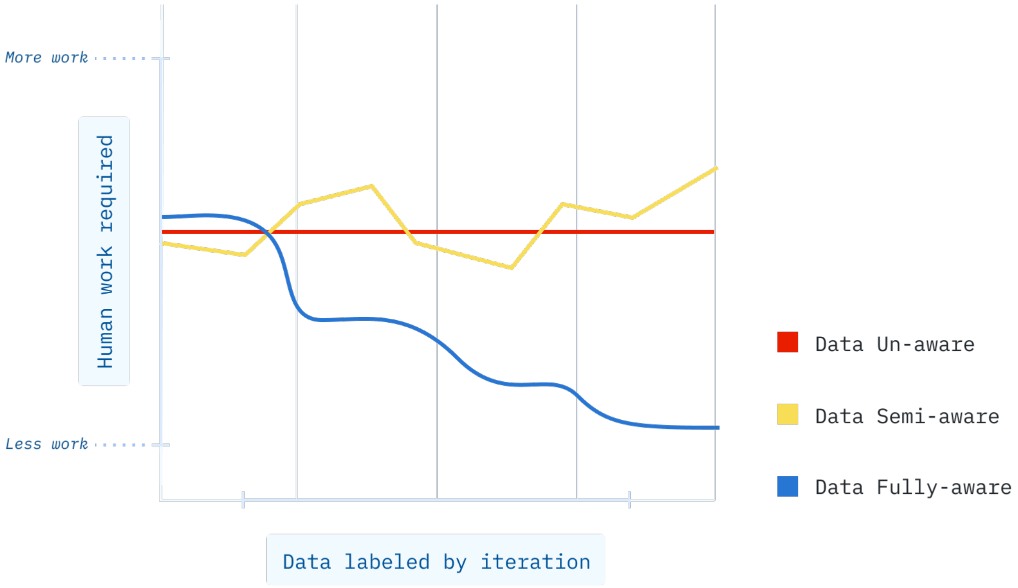
Model performance is usually better when machine learning teams use a data semi-aware algorithm as opposed to a data unaware one, but each image in the training dataset still needs to be adjusted for your model's exact needs, and this process will still take more labeling time as the model iterates, since it will require exponentially more training data with every iteration.

In practice, the data semi-aware method can provide immediate performance for some data. With this approach, the automation performance is usually static over the data or varies (sometimes unfavorably) after a few iterations and the labeling task moves away from common object labeling and towards the machine learning system's evolving needs. The method can be useful to get your model to a passable level of performance — say, to 80% accuracy — but improving it beyond that might require a more data-aware automation strategy.

Data fully aware

We've established that the more aware of your data your labeling automation method is, the more successful it will be — reducing time by hundreds of hours, and cutting significant costs. The data fully aware method uses algorithms that have been **trained on your dataset**.

Typically, an ML model is continuously retrained on the labeled data with each iteration, so that automation performance goes up over time. With data unaware and data semi-aware methods, the human time required to label data stays the same iteration after iteration; with a data fully aware method, human effort reduces as the number of iterations increases.



As the number of iterations increase, the data fully-aware method is the only automation method that consistently reduces human work.

This is particularly useful when labeling requires expensive expertise. For example, to train a model to detect cancer cells, a pathologist might mark a contiguous group of cancer cells on an image as a first step, and then a less-skilled group of workers might outline the malignant cells.

In this case, a predictive model trained on the pathologist-labeled data can learn to pre-label groups of cells as cancerous, speeding up the work of the pathologist. A second predictive model, trained on the data in which groups of cells have been outlined, can pre-draw segmentation boundaries around those cells, speeding up the second step as well.

Here's another example of the data fully aware method: a computer vision model built to diagnose lung cancer might need to be trained on lung scans labeled by experienced radiologists. The labeling task is expensive and time consuming for the first dataset, but over time, the model itself can be used to pre-label the scans. Once the pre-labeling is accurate enough, the task can be turned over to less skilled – and less expensive – workers to verify the automated labels, greatly reducing time and expenses. This is what we call **model-assisted labeling**: where models are used to train themselves.

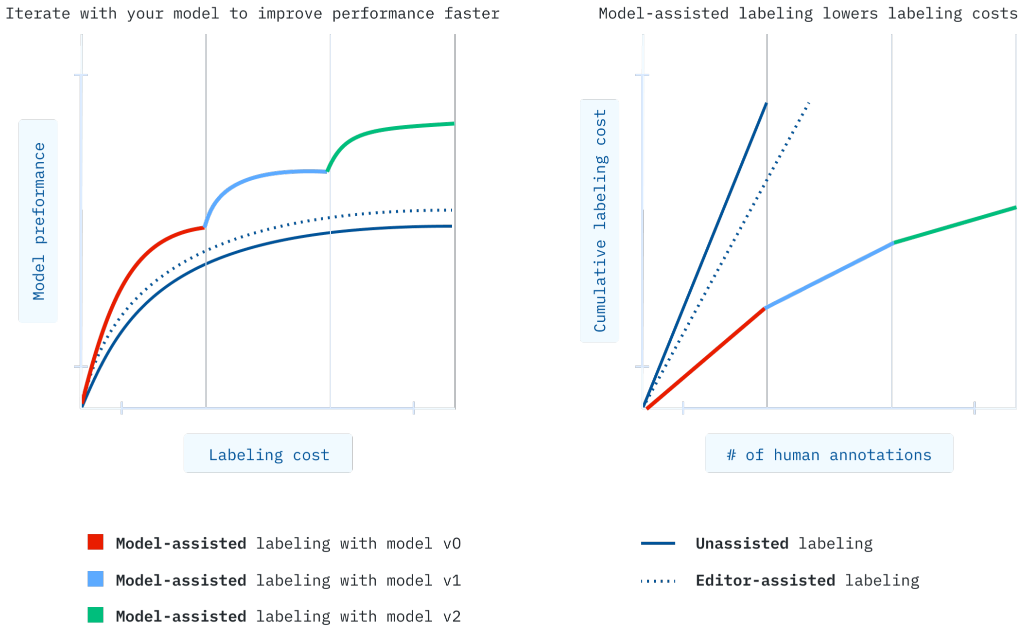
03 Model-assisted labeling: a labeling automation strategy proven to reduce time and effort

The benefits of model-assisted labeling are clear and significant.

1. **It's faster.** With each iteration, the human work required for labeling decreases even as the amount of training data required increases
2. **It's less expensive.** In use cases that require specialized experts to label data, teams can cut costs by 50 - 70%
3. **It will deliver better model performance.** Model-assisted labeling is the only automation method that speeds up the labeling process over time, enabling faster iterations and leading to a more accurate, performant model

Model assisted-labeling is the most effective labeling automation method. We know that labeling effort scales linearly with the number of annotations, and that the amount of labeled data required increases exponentially with each iteration of the model — presenting a clear incentive for ML teams to increase labeling efficiency. But no automation method is going to know your specific use case better than the model you're building.

Your model, trained on your data will train faster and become much more performant than any off-the-shelf algorithm trained on a generic dataset. Let’s explore a few examples of model-assisted learning in action.


Iterate with your model to improve performance faster

Model-assisted labeling lowers labeling costs

Model-assisted labeling increases model performance, reduces the number of human annotations required, and reduces costs with each iteration, unlike less aware automation methods.

Real-world examples

Model-assisted labeling with heuristics: crops vs. weeds

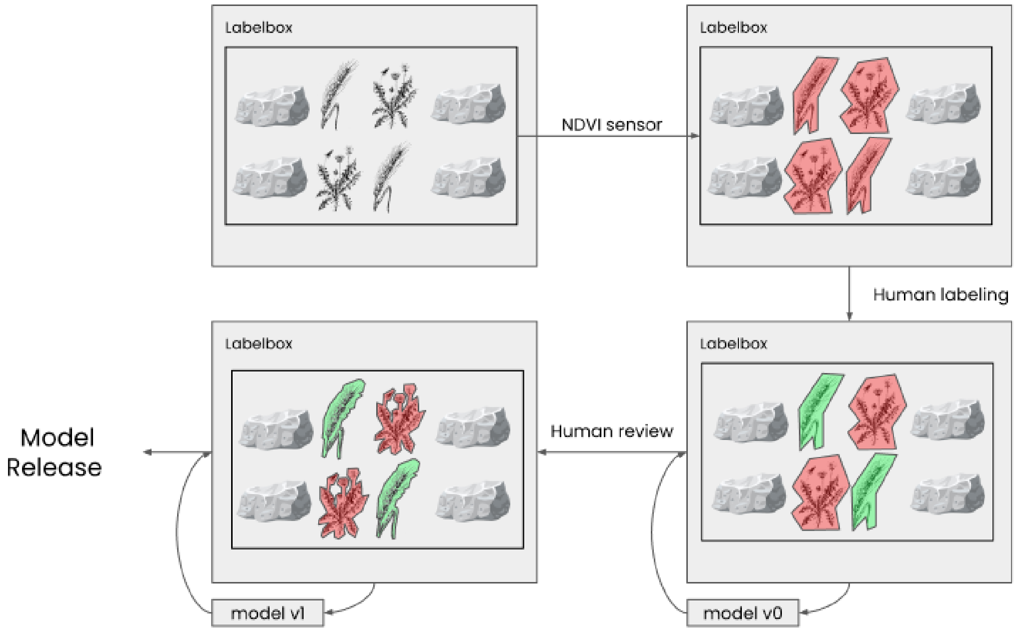
A large agtech enterprise is training a machine learning model to differentiate between types of crops and weeds based on images of fields. These images usually also show dirt and rocks. At first, to produce segmentation masks, they had a labeling workforce do a first pass to label the crops and weeds in the images. This was followed by a second pass by a team of botanists, who reviewed and corrected the labels.

This soon became an extremely time consuming process as the model required more and more labeled data, so they adopted two model-assisted labeling solutions to speed up the process.

First, they imported a mask generated by their NDVI sensor, which picks up organic plant matter indiscriminately. Then they had their workforce correct and further differentiate between weeds and crops, as they now knew where to focus their efforts.

Second, they trained a model using this labeled data and imported these model predictions into Labelbox, a training data platform. Their labelers and botanists were then able to complete final reviews and fine edits, allowing their model to improve even more.

Today, they continue to import these model predictions back into Labelbox, so that the labeling process goes faster with every iteration as their model becomes more accurate. By using model-assisted labeling, this agtech company was able to **cut their labeling costs in half**.



This agtech company used model-assisted labeling to speed up the labeling process and cut their costs in half.

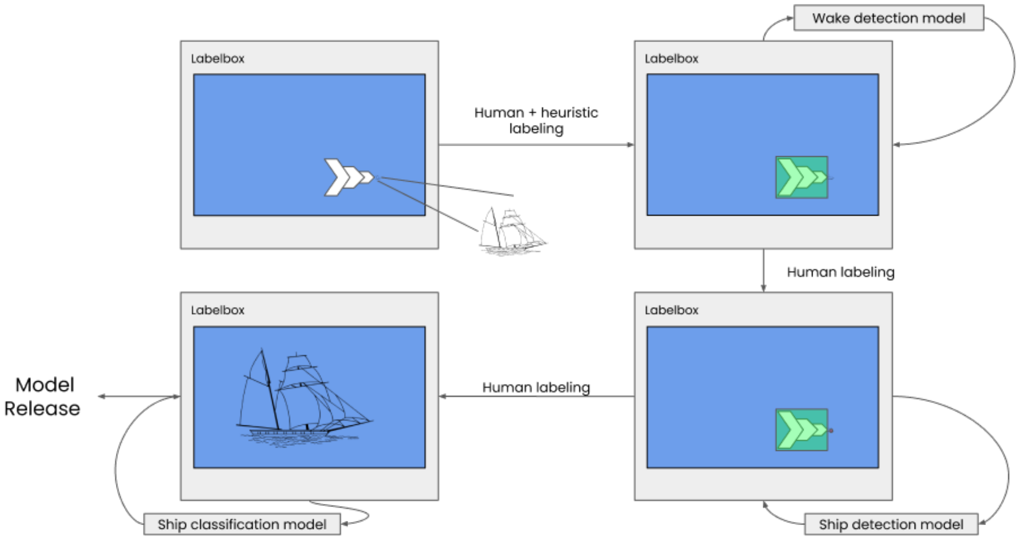
Model-assisted labeling with related models: ships in the sea

An ML team was training a model to detect moving ships in satellite images of large swaths of the ocean. At the beginning, their labeling team was presented with huge images, where ships appeared as tiny, hard-to-spot dots. The task of labeling them was tedious and difficult.

They soon realized, however, that moving ships leave wakes, which form large white patches on a sea of blue, making them easier to spot than the ships themselves. Using a combination of heuristics (like finding a white patch on blue) and human labeling, the ML team was able to produce weak labels to find wakes on the images, which they used to train a wake detection model.

With this model labeling wakes on the satellite images, the labelers only had to confirm that there was a ship at the end of the wake. This strategy helped the team drastically cut down labeling time and improve accuracy.

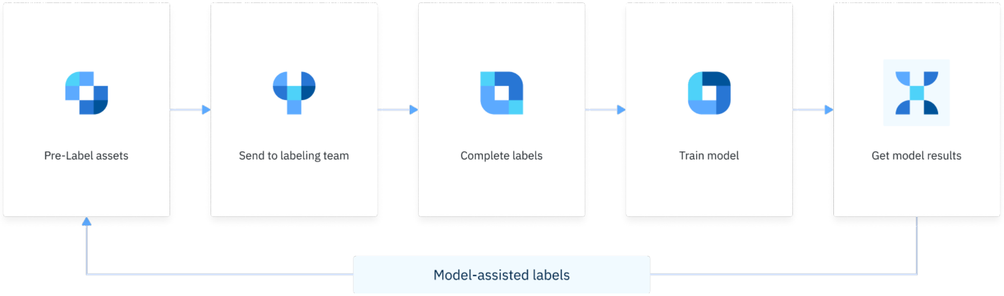
The team then used this data to train their ship detection model, which they used in turn to train data for the next iterations, speeding up the labeling process even more.



This ML team first trained their ship-finding model with another, similar model — and then used their own model to generate pre-labeled data for further iterations.

Training Data Platform

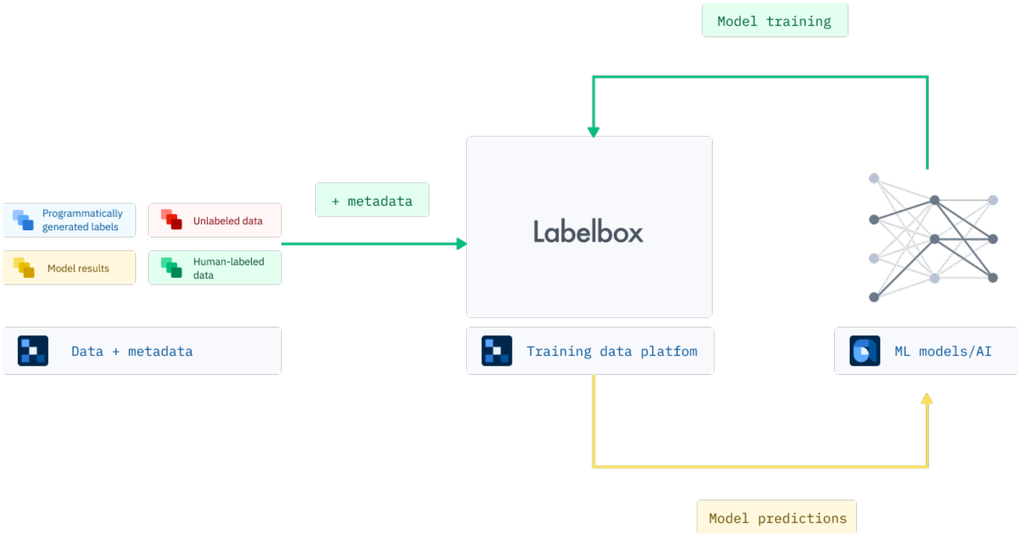
The model-assisted labeling method presents its own challenges, however. A machine learning team that chooses this method must find a way to connect their ML systems securely with the labeling tools and workflows, so that labeled data can be easily fed to the model and the model's output data can be brought back to the labelers.



Model-assisted labeling requires a loop of data between the machine learning system and the labeling process.

This is why model-assisted labeling is best achieved through the use of a **Training Data Platform (TDP)**, a tool that effectively creates a loop between the labeling process and machine learning systems. This loop creates a fast and efficient iteration cycle — a key component to improving model performance and gaining a competitive advantage.

A TDP helps machine learning teams manage their training data workflows so that data science teams can work securely and efficiently with internal and/or external annotation teams. For the purpose of model-assisted labeling, a TDP can be an invaluable tool that closes the gap between the ML model and the entire labeling process, including tools, datasets, and the labelers themselves. ML teams looking to implement a model-assisted labeling system will require a TDP that seamlessly loops their partially trained model's results back through the labeling process.



A TDP connects the machine learning system with every part of the labeling process.

Automated labeling services: risk factors to consider

It is possible to do model-assisted labeling without using a TDP. Some vendors now offer automated labeling services built around data semi-aware automation methods that quickly become fully aware as their model learns more about your labeling needs. While these services often seem like efficient solutions, they present risks that ML teams should be vigilant about.

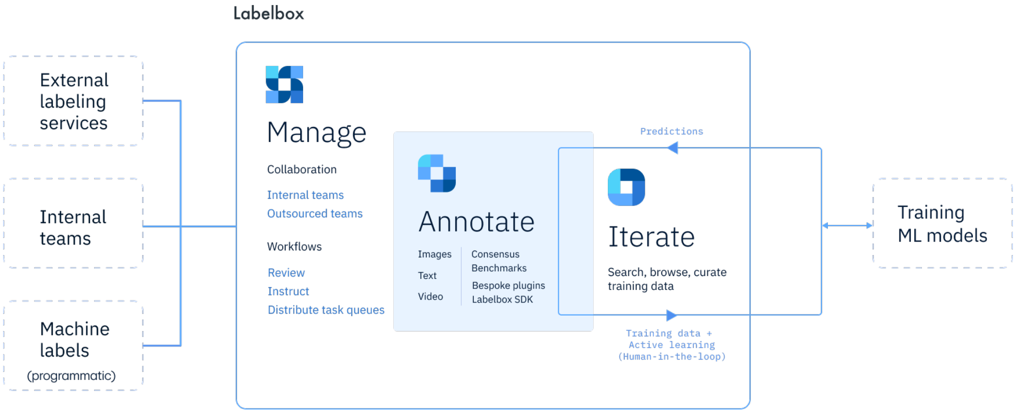
When a service automates labeling for your model, your model will only ever become as performant as theirs — meaning that it's limited by the level of service they provide. By using disparate tools throughout the training and iteration process, your team might also be unintentionally making it more difficult than necessary to place quality control measures such as benchmarking and reviewing. They may need to manually deliver the labeled data from the automated service to the reviewers, or to a separate labeling team for benchmarking, and then again transfer it to the model itself.

Your data and model are important pieces of IP. An automated labeling service will have access to this data, and if the vendor is training their model on your dataset, they will effectively be creating another instance of your model that you have no control or visibility over. While speeding up the training process and reducing costs are important factors to consider for ML teams, it's certainly more important to secure your work against potential competitors.

Using a TDP to implement model-assisted automation will ensure that your model's performance will never depend on the tools or services you use. It will also give your team full visibility and control over every aspect of the training process, from the dataset to the labeling team to the new pre-labeled data generated by your model. Neither the TDP nor the labeling team will ever get a copy of your model.

04 Model-assisted labeling with Labelbox

It's easy to implement model-assisted labeling with the Labelbox Training Data Platform.



The Labelbox TDP delivers all the capabilities you need to manage your collaborators and workflows, annotate your datasets, and iterate on your model.

1. Upload your datasets and add the labeling team, who can use our tools (or their own) to annotate data
2. To loop your model-generated data back through the labeling process, just upload the data into Labelbox using our Python SDK
3. Your labeling team will then have access to the new dataset and can adjust or correct the annotations

You can find more detailed instructions on the process in the Labelbox documentation. Alongside enabling model-assisted labeling, Labelbox also provides a suite of tools and services for all your training data needs.

Labelbox

Learn more about our offerings, sign up for a demo, or start using our free version today at www.labelbox.com.