The essential guide to AI training data

TELUS® International

Training data is a resource used by engineers to develop machine learning models. It's used to train algorithms by providing them with comprehensive, consistent information about a specific task. Training data is usually composed of a large number of data points, each formatted with labels and other metadata. How you build, format and annotate your training dataset has a direct impact on the model you create. In fact, poorly processed data is one of the most common reasons that machine learning projects fail.

However, if you haven't worked with training data before, it can be difficult to know where to start. After all, data can be surprisingly complex. It's hard to figure out what a dataset should look like and how to improve it.

TELUS International provides professional data services to some of the world's largest companies, and our collection capabilities at scale remain a source of competitive advantage for disruptive brands and leaders in artificial intelligence (AI). We've put together this helpful guide to address some of the fundamental considerations when embarking on an AI data project.

Contents

What is AI training data?

Training data is a collection of labeled information that's used to build a machine learning model. It usually consists of annotated text, images, video or audio. Through training data, an AI model learns to perform its tasks at a high level of accuracy.

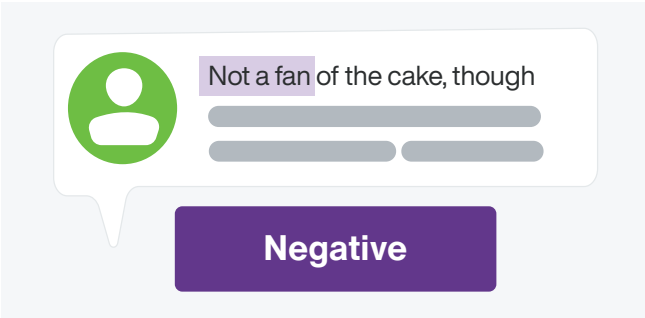
In other words, training data is the textbook that will teach your model to do its assigned task. Think of the algorithm as a student and the training dataset as a textbook filled with example problems. The more the algorithm "studies", the better it does in the final test, which is real world application.

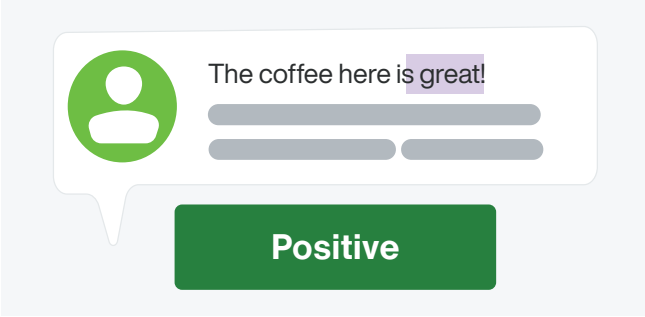
Training data plays several different roles in the development of your model. In fact, most datasets are used multiple times throughout the training process, as this helps to refine the model's predictions and improve its success rate. This is possible because of the variables contained in the data. Through identifying and evaluating variables, it's possible to assess their impact on the model and adjust them in ways that will strengthen the final product. The best data is usually extremely rich in detail and capable of improving your model over many training cycles.

Most training data contains pairs of input information and corresponding annotations, which are often called the target. These annotations, also called tags or labels, contain relevant metadata that helps your model to make more accurate predictions. Since these labels are so important to the training process, the makeup of each individual dataset can vary drastically.

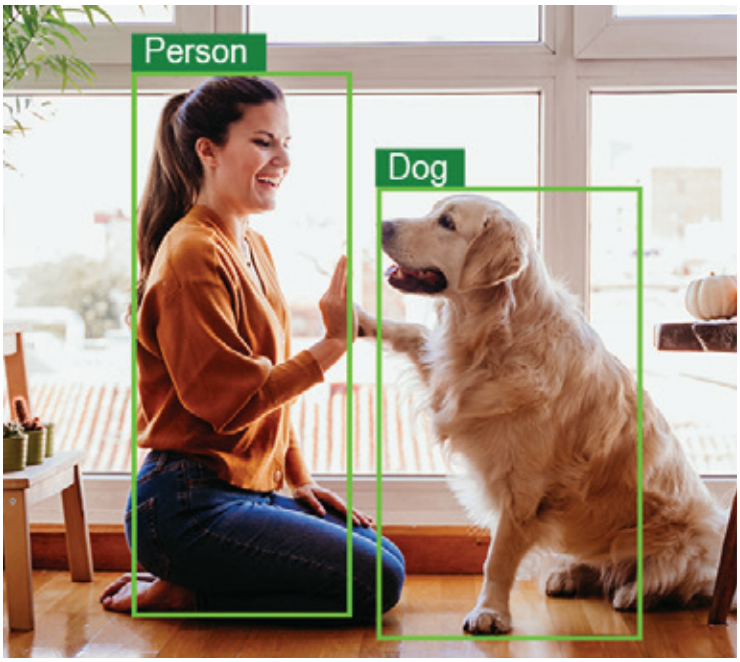
For example:

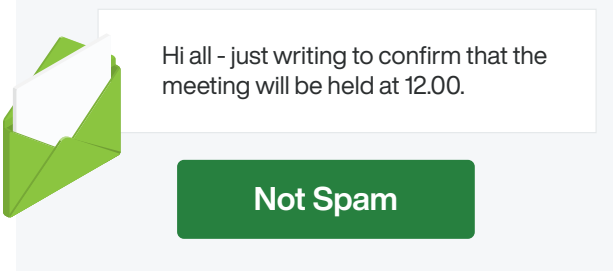
In **sentiment analysis,** training data is usually composed of sentences, reviews or tweets as the input, with the label indicating whether that piece of text is positive or negative.



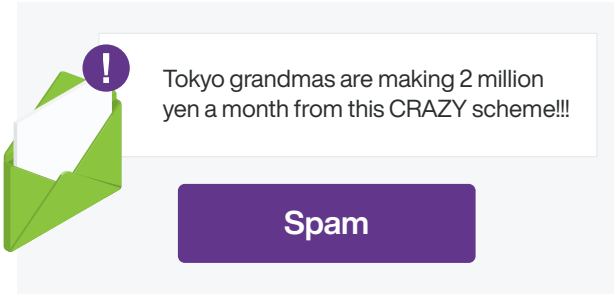


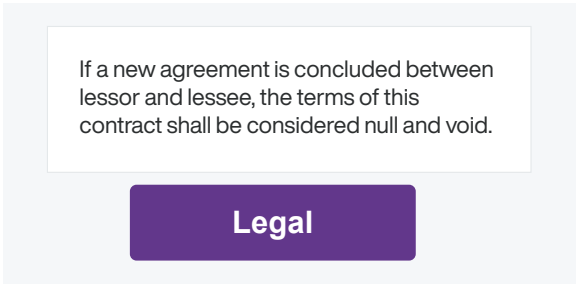
In **image recognition**, the input would be the image, while the label suggests what is contained within the image.



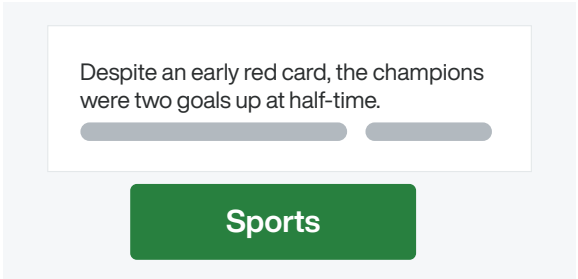


In **spam detection**, the input is an email or text message, while the label would provide information about whether the message is spam or not.

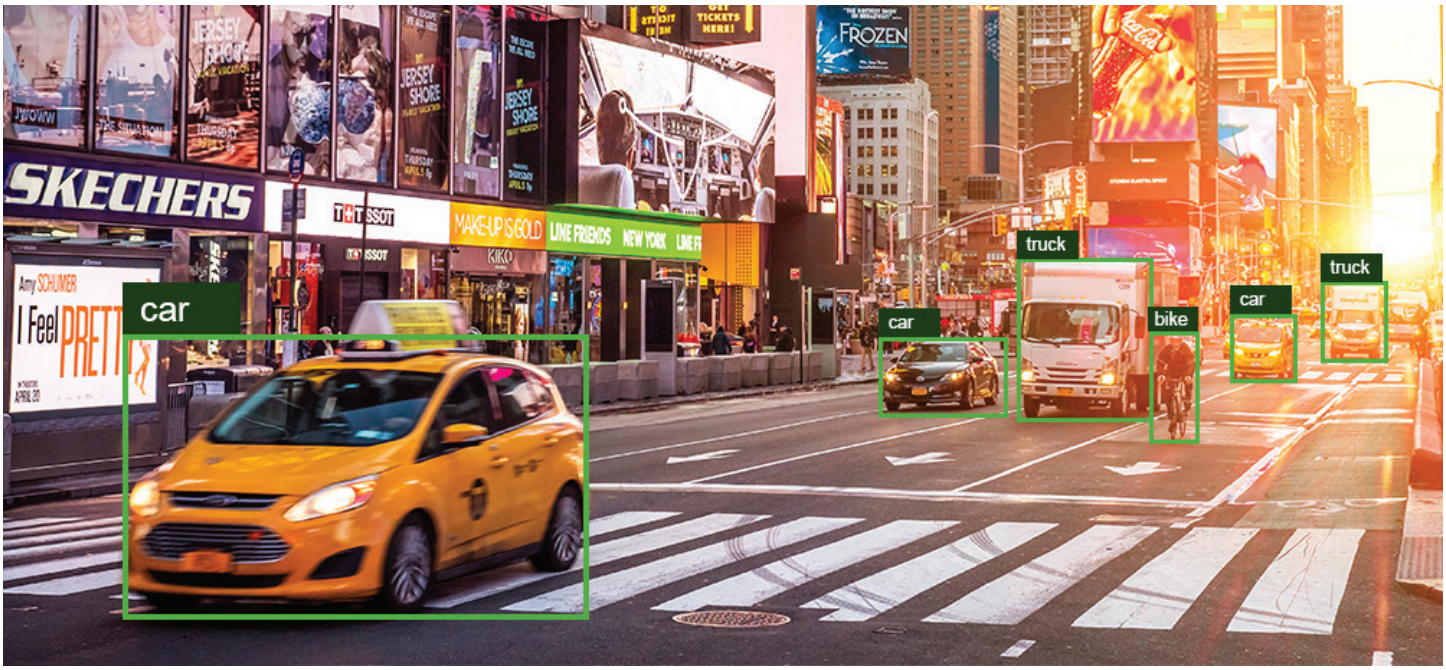




In **text categorization**, sentences provide the input while the target would suggest the topic of the sentence, such as finance or law.



It's also possible to have multiple labels for a single piece of raw input data. For example, in bounding box annotation, various types of vehicles can be ordered into different classes:



It's also common to have multiple labels for a single data point in text-based tasks. For text classification, words can be ordered into a variety of categories based on meaning:

World War II was a global war that involved most of the world's countries, including the Allied and Axis powers. With 40 to 50 million deaths recorded, it is still regarded as the largest and deadliest war in human history.

World War II

It's easy to see from these examples that datasets are often highly specialized. If two different AI programs use the exact same training data, it can result in at least one suboptimal model. This is true even if both programs deal with the same broad category of input information, such as sentences.

Training, testing and validation data

Datasets can be used in multiple ways to serve a range of different training processes. In fact, to build a machine learning model you'll need three types of training data, each of which performs a different role.

Before going any further, it's worth noting that the term 'training data' has two separate meanings. Training data is used both as an umbrella term for the total data needed for your project, and to refer to one of the specific subsets of data below. While this might be confusing initially, there are important differences between these three types of data:

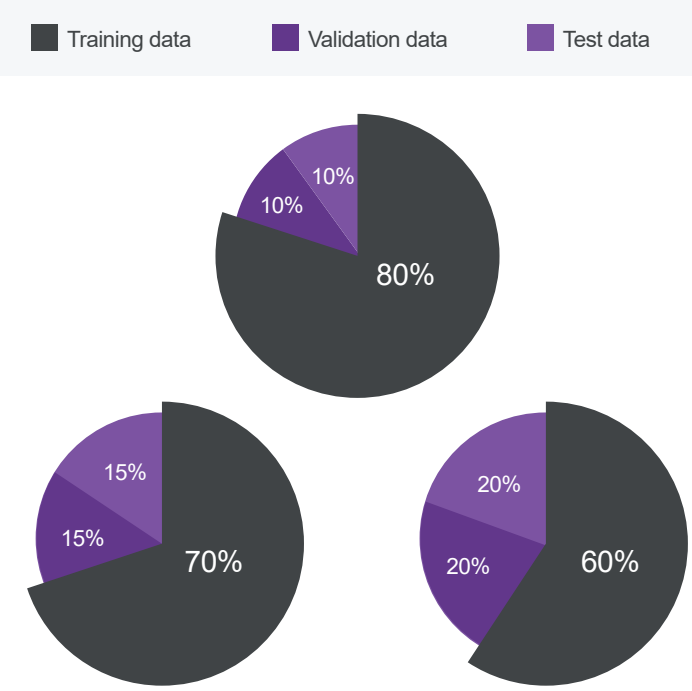
Training data is used to help your machine learning model make predictions. It's the largest part of your dataset, forming at least 70 to 80% of the total data you'll use to build your model. This data is used exhaustively across multiple training cycles to improve the accuracy of your algorithm. Training data is different from validation and testing data in that its classes are often evenly distributed. Depending on your task, this might mean that the data doesn't accurately reflect its real-world use case.

Validation data is primarily used to determine whether your model can correctly identify new data or if it's overfitting to your original dataset.

Validation enables your data scientists to adjust hyperparameters and improve your model's accuracy.

Testing data is used after both training and validation. It aims to test the accuracy of your final model against your targets. It also provides further confirmation that your model isn't overfitting to your training and validation data.

These three types of data usually work best if they're smaller parts of one overarching dataset. This helps to ensure consistency and keeps the data relevant to the goals of your project. To avoid selection bias, it's best to shuffle your data into these three categories randomly. Common ratios for these splits are as follows:



Why is training data important?

Without training data, machine learning would not exist. In fact, the success or failure of your model will be defined by the cleanliness, relevance, and quality of your data. Just as a student with an outdated textbook is unlikely to achieve top marks, your model will only excel at its task if your data accurately reflects the real-world scenarios your model is built for. As a result, the world's best models are built on comprehensive datasets, complete with a range of detailed labels.

Think of data as your not-so-secret weapon. The better your data is, the better your model will be.

How much training data do I need?

When starting out with a new machine learning project, it's common to ask how much training data you'll need to ensure high performance from your algorithm. The question itself is simple. The answer – not so much. The blunt truth is that there's no magic number of data points that will turn your model from good to great.

The reason for this is that the number of data points you'll need for your project is affected by a wide range of factors, all of which can influence the eventual size of your dataset to a greater or lesser degree. Due to the nature of machine learning, it's unlikely that you'll ever know all of these factors. You'd probably need another machine learning model just to figure them all out and calculate their weights. Instead, it's better to approach data quantity as an iterative process, adding more data as and when your team decides it's necessary.

So, the short answer to the question is usually to start with as much data as you can reasonably get and let the model's needs take it from there. However, if you really want a ballpark figure for those hundreds, thousands, or millions of data points, you'll need to do a little bit of research. Below, we'll discuss some of the more common issues to watch out for when it comes to dataset size. After that, we'll discuss how you can develop a rough idea of how big your dataset should be and provide some publicly available figures from real machine learning projects.

Why is it difficult to estimate dataset size?

It's difficult to figure out the exact size you'll need for your dataset due to the nature of the training process. Training aims to build a model that understands the patterns and relationships within the dataset as a whole, rather than just each single data point. This is why it's important to gather enough data to give your algorithm an accurate understanding of the complex network of meaning behind and between your data points.

The parameters that contribute to this network of meaning are different for every dataset. Usually, they're so numerous that it's impossible to work them all out ahead of time. However, there are some factors that often have a high degree of influence on the size of your dataset. They are as follows:

Complexity of model

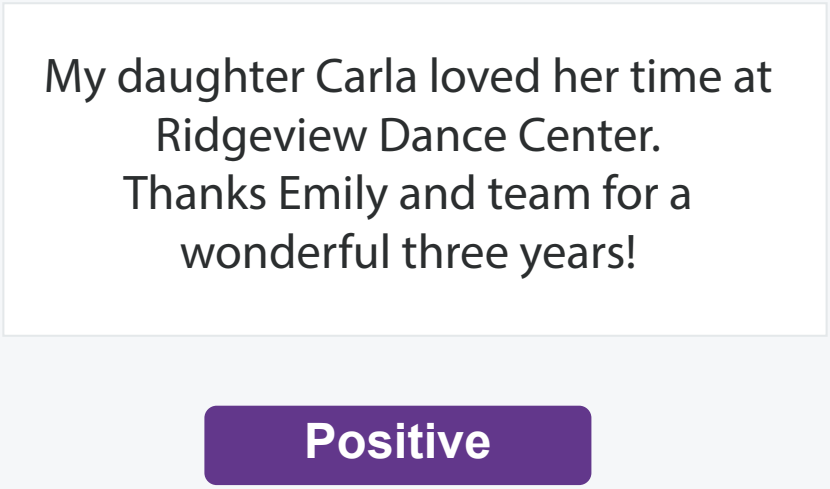
For every parameter that your model needs to account for, it will need more training data. For example, a model that identifies the make of a car has a fairly small, set number of parameters that mostly relate to the vehicle's shape. However, a model that has to determine the cost of that car has to understand a far bigger picture, including the car's age, condition and any other economic or social factors that might impact the price. The higher number of parameters involved here mean that the second model requires a larger dataset.

Training method

Different levels of complexity also require different training methods. Many traditional machine learning algorithms use structured learning, which has a fairly low ceiling for additional data. With this method, you'll quickly find that more data has very little impact. In contrast, models that use unsupervised learning can improve without structure and figure out their own parameters. This method requires a lot more data and also extends the learning curve where further data can have a positive impact.

Labeling needs

You can annotate a data point in a variety of ways. As a result, how you label a data point can create significant variation in the number of data points you need. Let's imagine that we have 1,000 sentences of input data. For sentiment analysis, you might only label each sentence once, as positive, negative or neutral. However, for entity extraction, you might label five words in each sentence.



> Despite having the same raw data, one task yields five times more labels than the other. If one single data point can contain a large number of labels, then you might be OK with a smaller overall dataset.

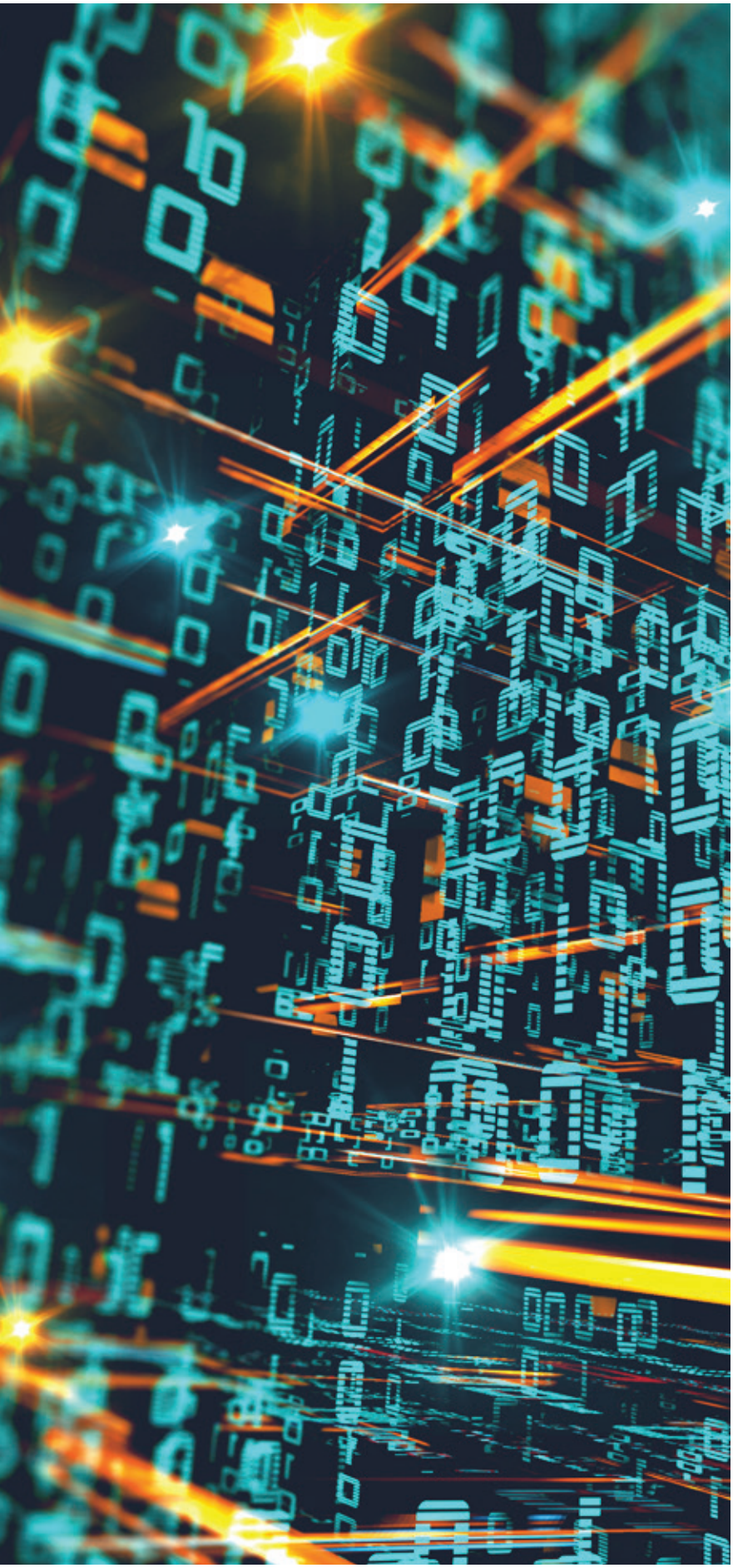
Tolerance for errors

Of course, the task you want your model to complete has a big impact on the amount of data you need to collect. This is because some models need to have a higher level of performance on edge cases and a lower rate of error. Think of the difference between a model that predicts the weather and one that identifies patients who are at imminent risk of heart attacks. One of these has a much lower threshold for error than the other. The lower your acceptable level of risk, the more data you'll need to ensure that risk is mitigated.

Diversity of input

In the real world, a model can encounter a wide variety of input data. A chatbot, for example, has to be able to understand a variety of languages written in formal, informal or even grammatically incorrect styles. It has to be able to understand all of them in order to provide a high level of customer service. In cases where your model's input won't be highly controlled, you'll need more data to help the model function in that unpredictable environment.

The deciding factor for how much data you'll need is your project's unique requirements and goals. Each project requires a unique balance of all of these influencing factors, which you'll have to figure out for yourself when coming up with that target dataset size. Keeping this in mind, let's now dive into some of the ways that you can begin to figure out your data needs.



How can I calculate my data needs?

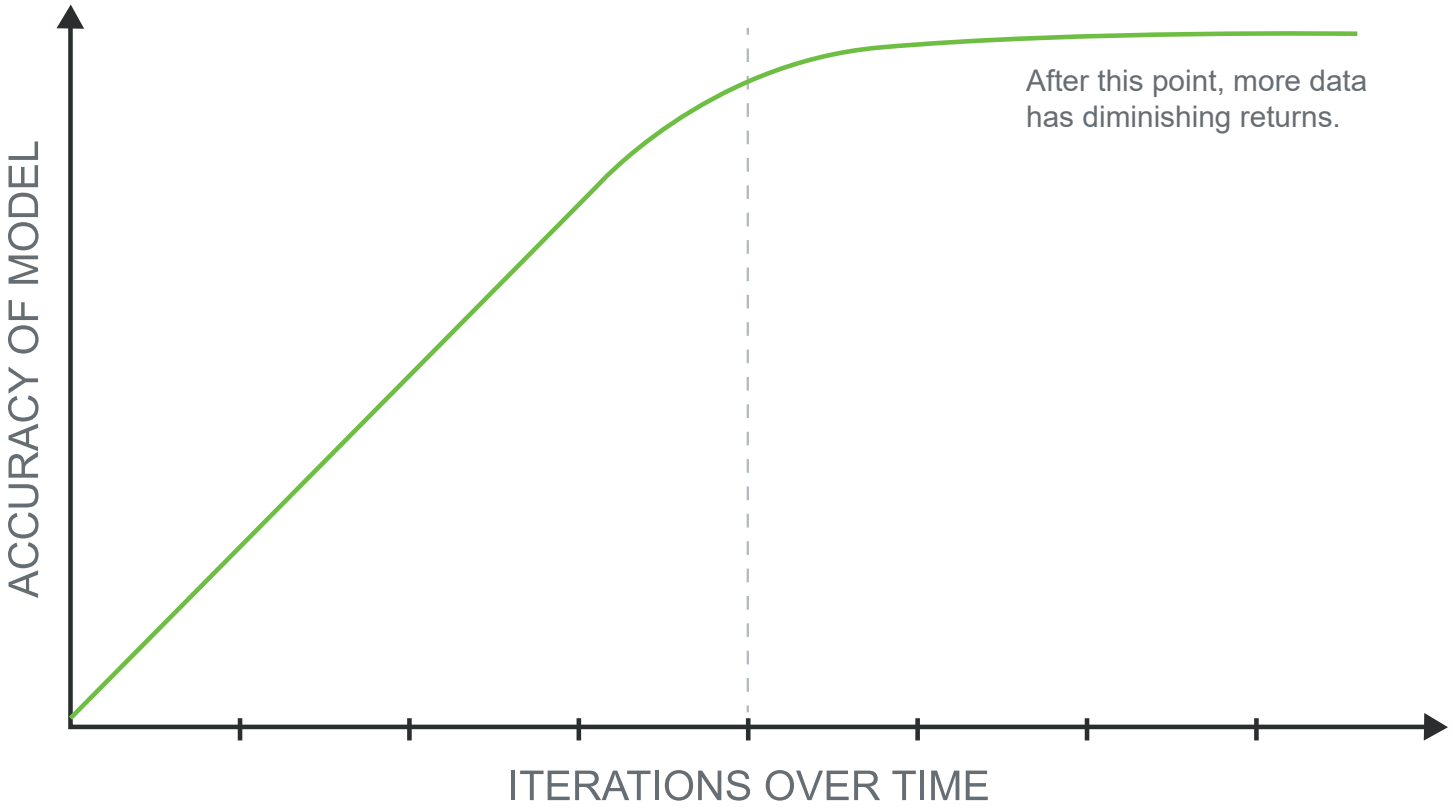
A general estimate of the amount of training data you'll need should be more than enough to get you started. Here are a couple of common methods for reaching a ballpark figure:

1. Rule of 10

This is a common, if controversial, rule of thumb which states that a model requires ten times more data than it has degrees of freedom. A degree of freedom could be a parameter that affects your model's output, an attribute of one of your data points, or even just a column in your dataset. The aim of this rule is to compensate for some of the variability that all of your parameters bring to the model's input. Although some people think this just reframes the debate around another question that's impossible to answer, it does provide a quick estimate that will help you to kick off your project.

2. Learning curves

If you want to make a more evidence-based decision and you already have access to some data, you could create a study to evaluate your model's ability based on the size of your dataset. By plotting results on a graph after each iteration, you can start to see a relationship between dataset size and your model's ability. This will help you to identify the point where more data starts to provide diminishing returns. This approach is slightly more technical, involving the creation of a few logistic regression problems, but it should give you a more reliable result.



In the end, an iterative approach to building a dataset is probably the best. Just start working on your model with the data you have, and add more as it becomes necessary. Once you start seeing results, your needs will clarify themselves.

Even if you manage to source the perfect number of data points for your training program, it's highly unlikely that your dataset will be ready to train your model. In fact, data requires a lot of processing before it can have an impact. We'll explore exactly what that looks like in the next section.

How can I improve the quality of my data?

Read any article about training data and you'll quickly come across the well-worn phrase "garbage in, garbage out (GIGO)". While it's true that your model will only be as good as the data you feed it, what exactly constitutes 'garbage' is often glossed over. As a result, many people who are working with machine learning for the first time find it difficult to approach quality assurance for their training data.

In short, a high-quality dataset is one that has undergone a rigorous cleaning process and that contains all of the necessary parameters for your model to learn to do its task. These parameters will be consistent across the entire dataset, with no overlap between them. The goal of the quality assessment process is to develop a model that performs at a high level of accuracy and precision against real-world data. What constitutes high accuracy is determined in the early stages of the project, along with any additional metrics that need to be used to measure quality.

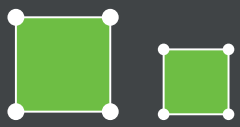
During this process, it's important that your dataset accurately reflects the real-world environment in which the model will be used, from the characteristics of the data to its distribution across classes.

As a result, quality assessment for training data focuses around processes that develop clean, annotated and distributed datasets – a mirror image of the data your model will encounter after it's implemented.

Cleaning and assessing your data in this way isn't always the most exciting part of machine learning, but it's absolutely crucial if you want to build a useful model. Let's unpack quality for machine learning training data and explore what that looks like in your dataset.

What is quality?

As we explained above, quality training data has a wide range of consistent, distinct parameters that provide your model with all the information it needs to perform its task. But let's dive a little deeper. A high-quality dataset usually has all of the following characteristics:

CHARACTERISTIC	DEFINITION	ACTION ITEMS
 Uniformity	All data points attribute values equally and come from comparable sources	Check for irregularities when pulling data from multiple internal or external sources
 Consistency	All data points are the same	Ensure that classes are distributed
 Comprehensiveness	Dataset has enough parameters to cover all of the model's use cases, including edge cases	Check that you have enough data; include examples of edge cases in an appropriate volume
 Relevancy	Dataset contains only parameters which are useful to your model	Identify important parameters; consider asking a domain expert to perform analysis
 Diversity	Dataset accurately reflects the model's user base	Perform user analysis to uncover hidden biases; consider pulling data from both internal and external sources; consider employing an expert for a third-party perspective

While these general themes are present in all great datasets, it's important to remember that every project is different – and that quality means something slightly different for each project as a result. Always make sure that you have a thorough understanding of your project specifications, as this will help you to build a customized QA process that will improve your dataset.

Training data quality best practices

There are a whole range of processes that you could undertake to improve the quality of your model – so many, in fact, that it can be difficult to know where to begin. Before wading in, it's important to first clarify what your model's target accuracy is. Knowing the acceptable rate of error can help you to understand how thorough your QA process needs to be, which can change how you approach certain tasks.

It's also important to remember that quality in machine learning is inherently iterative. As you train your model, you'll probably find patterns in your inaccuracies. For example, you might find that your voice recognition system struggles to recognize children's voices. When this happens, you'll need to collect, clean and annotate further data which takes this into account.

Keeping that in mind, this section outlines some of the biggest problems that you'll need to address at each stage of the dataset creation process. It will also give you some basic advice on measuring quality and point you towards some general processes that you can implement as you finetune your dataset.

Data collection

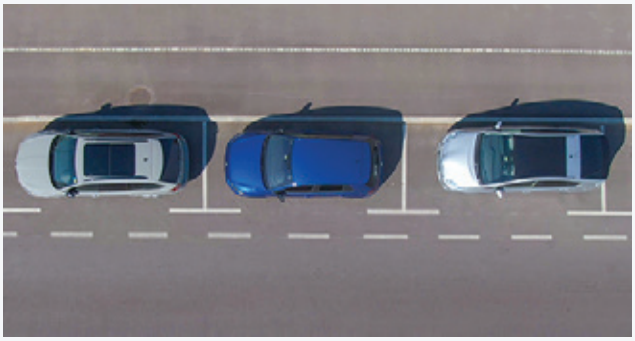
Ensuring high-quality data collection is no simple task. However, there's plenty you can do to make sure the data you gather is suitable for the task at hand. In particular, there are a few best practices that you'll want to focus on. These are:

Coverage planning

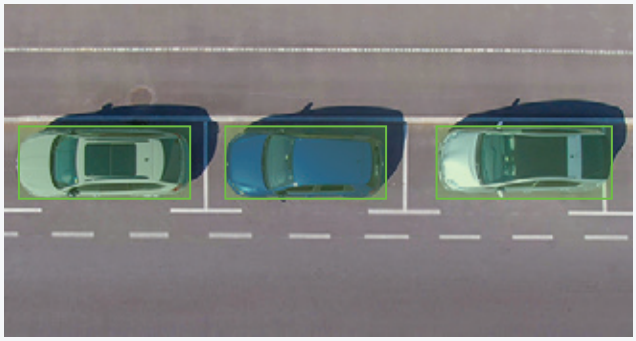
Before you start collecting data, you need to know what your target classes are, how you plan to distribute your data across those classes and your metrics for measuring data quality. If you don't have an answer to these questions, it will be difficult to do anything meaningful with your raw data without the risk of losing information.

Check for structured and unstructured data

The data you're about to collect will fall into one of two categories. Structured data is machine-readable, annotated and has metadata that the model can learn from. Unstructured data includes no annotations or metadata, including only the original images, text or video.



Unstructured



Structured

However, it's usually more comprehensive than structured data. The vast majority of data is unstructured at the point of collection, but it's worth checking whether you're primarily collecting structured data or mixing the two.

Feature extraction

Raw data usually contains a lot of noise, so it's important to separate the features you want from those which are irrelevant. The process of feature extraction is useful when you want to remove redundant data, as well as shrinking the amount of data you need to process without losing valuable information.

Mitigate bias

Bias occurs when the data you have doesn't accurately reflect the conditions your model will operate in. It comes in many forms, from gender or racial bias to observer and selection bias. It's extremely important that you avoid all forms of bias for your model to work correctly.

The most successful AI projects treat data collection as an iterative process. Throughout the life-cycle of your model, collect data from any user engagement to improve your model. After it has been refined, you'll be able to use this new data to improve accuracy and performance.

Data cleansing

There's a reason that 80% of a data scientist's job is preprocessing. How well you clean your data determines its overall quality – and quality data means quality results.

What is data cleansing?

Data cleansing is the process of preparing your data for modeling, particularly through improving the quality of your dataset. A clean dataset will have clear, precise parameters and consistent processes for managing outliers and missing values. In other words, it ensures that your data is accurate, complete and relevant.

This is an essential step in the dataset preparation process, largely because of the nature of raw, unannotated data. It's common for a recently collected dataset to be incomplete, partially incorrect or contain a variety of formats. Training a model on this can cause a variety of errors, which will ultimately result in a much lower overall accuracy. As a result, data cleansing tasks are primarily based around formatting datasets, standardizing them and dealing with missing data.

Common data cleansing problems

Since every model has a different task or specialism, the process of data cleansing will look different from project to project. However, there are certain data cleansing tasks which are required for every dataset. They target specific structural problems that are present in many datasets and are extremely important to ensure your project's success. If you spot any of these problems in your data, it's crucial that you take action to solve them:

Duplication

Unsurprisingly, duplication involves data points or parameters that have somehow made their way into your dataset more than once. This often happens when you pull data from multiple sources or when you're scraping data from a huge corpus, such as Twitter. Duplicated data points don't add any value to your model and may actually lead it to make false conclusions. Make sure you remove any duplicate data points from your dataset before you begin training.

First data

Duplicate data

Outliers

There are occasionally large groups of observations within your dataset that aren't actually related to the problem you're trying to solve. For example, you could find that a dataset of product reviews captures the date of the review. It's possible to remove these outliers and improve your model's performance, but first you have to be absolutely certain that the observation is irrelevant to your model. If you've determined that these outliers are not an important source of information, you can delete them.

Structural errors

In some cases, you may have classes that have been mislabeled within your data, leading to messy class distribution. For example: 'USA', 'usa', and 'the US' could all be separate classes within a dataset. Inconsistent capitalization and similar errors can lead to multiple classes within the dataset that denote the same thing. Fix any typos, capitalization or mislabeled classes in your dataset for a cleaner, more accurate distribution.

us

USA

Missing values

Some of your data may have fields which are missing information. It's important not to ignore these, since most algorithms can't accept missing values. How you handle missing values will depend on the model you're trying to build, the variable which is missing data and how much data is missing. Based on these considerations, there are several possible strategies, including deleting rows with missing data, deleting the variable or replacing missing values. Make sure to do some thorough analysis before making your choice.

Product	Category
Harry Potter and the Chamber of Secrets	Fiction - Young Adult
The Shining	Fiction - Horror
The Odyssey	Empty data
On the Origins of the Species	Non-Fiction - Biology

Data labeling

Of all the steps in the dataset creation process, data labeling is the most likely to need a QA system. From developing guidelines to monitoring annotators, it can be a pretty complex process. In fact, many teams outsource data labeling to specialist providers just to avoid the hassle.

If you need to assess data labeling quality yourself, there's plenty that you can do to make the process smoother. Below, we'll share some tips from our years of experience in data annotation on how to measure quality, identify common labeling errors and develop processes that will keep your data in great shape.

How can I measure data labeling quality?

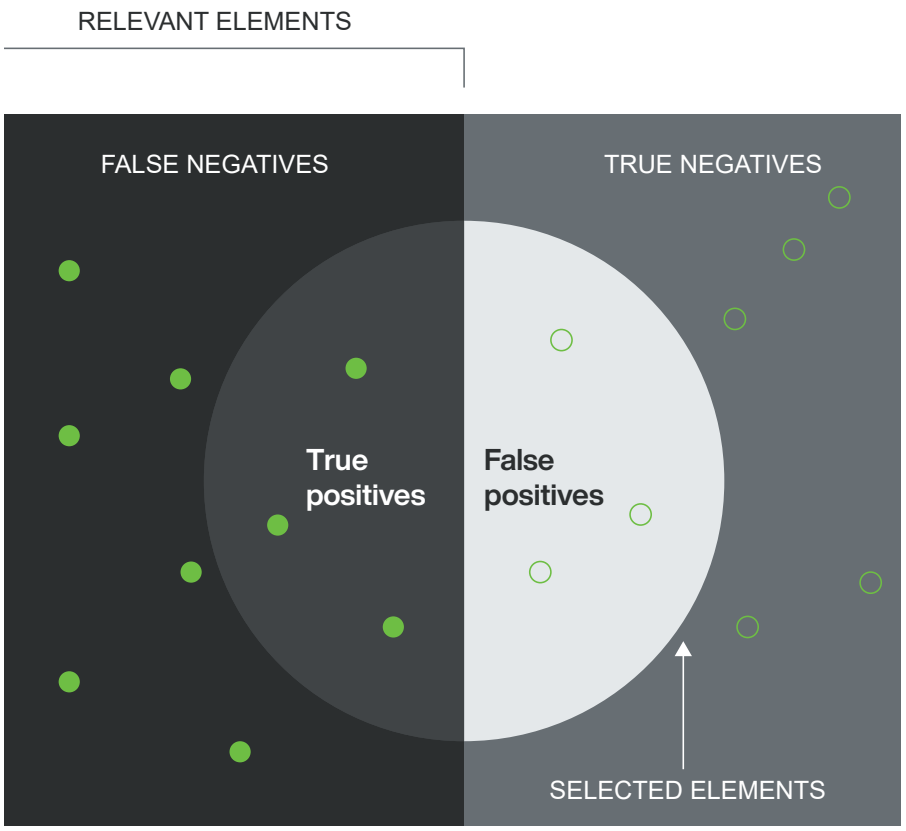
Quality measurements for data annotation revolve around two interconnected ideas. The first is accuracy. This involves measuring the annotations in your dataset against an ideal set of annotations, which in turn should accurately reflect the real-world conditions in which you plan to use your model. If your labels aren't accurate, then your model won't be able to deduce useful rules from the parameters within the data.

However, it's also important to measure the consistency of your annotations. Consistency means that every person in your team of annotators adds labels in the same way. If your labels aren't consistent, there will be a lot of noise in your dataset which your model might use to form judgements.

This is most often a problem in subjective tasks, such as classification, but can occasionally be present in objective tasks too. Most datasets require a combination of both accuracy and consistency measurements to track whether the annotations are improving the overall quality of the data. Here are two common metrics that data scientists use to measure accuracy and consistency:

1. F1 score

This measurement is commonly used in machine learning to monitor how well a model classifies data through its precision and recall scores. Precision checks how many of the model's answers were actually correct. Recall measures how many correct answers the model returned against how many it missed. The F1 score is the harmonic mean of these two calculations, where a score of 1 indicates perfect precision and recall.



How many selected items are relevant?

Precision:

How many relevant items are selected?

Relevant:

2. Inter-annotator agreement

Also called inter-rater reliability, this measures the degree of agreement amongst your team of annotators. This is particularly useful for subjective tasks that could be labeled differently by a range of annotators, such as sentiment analysis. There are several statistical methods for doing this, but one of the more popular amongst data scientists is Krippendorff's alpha.

Best practices for labeling

It's extremely important to monitor the quality of your dataset while you annotate, but that doesn't mean that it has to be extremely difficult. In fact, there are a variety of processes you can put in place to ensure that labeling improves your data. Here are some of the more common ones:

Create a gold standard

A gold standard is a set of data that reflects the ideal labeled data for your task. It's usually put together by one of your data scientists, who understands exactly what the dataset needs to achieve. Gold standards enable you to measure your team's annotations for accuracy, while also providing a useful reference point as you continue to measure the quality of your output during the project. They can also be used as test datasets to screen annotation candidates.

Use a small set of labels

Systems with a wide variety of labels can negatively impact overall annotation quality. This is because having a large number of options can lead to annotators becoming more indecisive or confused at the differences between options. For example, asking your annotators to categorize celebrities into "Actress – TV", "Actress – Movie", and "Actress – Stage" could lead to a high level of disagreement between annotators. Instead, a smaller number of possible labels leads to more reliable results.

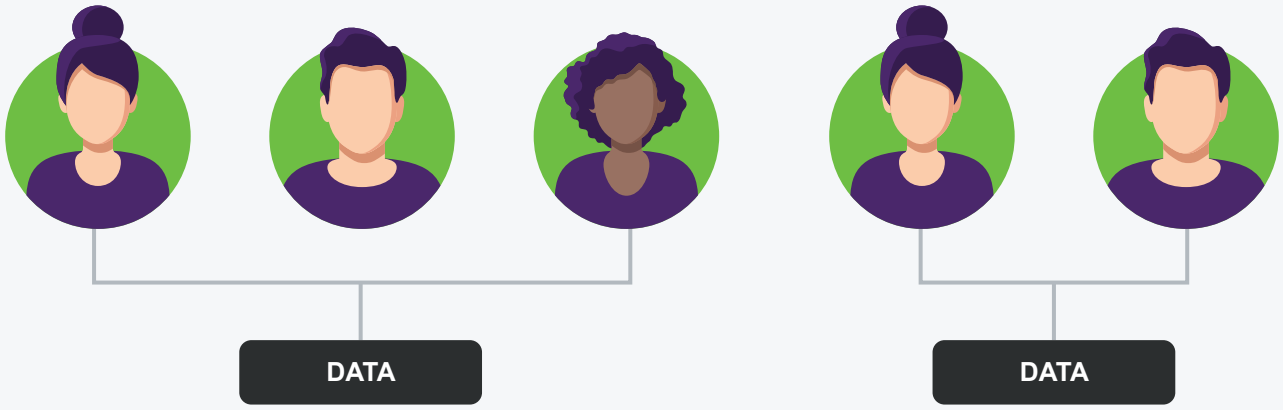
TV actress | Movie actress | Stage actress

Perform ongoing statistical analysis

Statistical representations can enable you to easily pinpoint outliers in your newly-annotated data for further review. While these often indicate human error from annotation, you should confirm this before taking action. After all, genuine outliers can be an important source of information for your model.

Use multipass

This involves asking multiple annotators to label the same data point. Multipass allows you to directly measure consistency within your pool of annotators and is particularly useful for improving quality for subjective tasks. If done to a large enough extent, it can allow you to make assumptions about the overall level of agreement in your dataset.



Review each annotator

Most data labeling projects spend a significant amount of time reviewing the output of each annotator in the team. One way to do this is to implement self-agreement checks, where you give the same annotator the same piece of data twice to see if they label it the same way the second time around. By doing this multiple times with different pieces of data, you can observe annotator consistency, begin to target sections of the dataset which are likely to have errors and give your annotators advice that improves the quality of their work.

Hire a diverse team

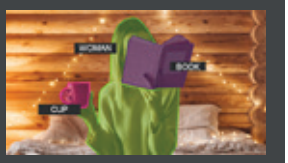
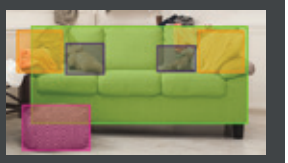
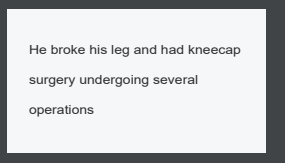
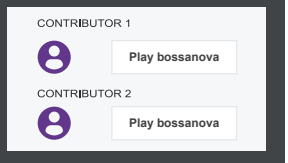
The easiest way to combat annotation bias is to have a diverse team of annotators. By doing so, you can avoid selection bias, where you only hire annotators who fit your expected user base rather than your actual user base. This is particularly important when you're labeling data for a subjective task, such as sentiment analysis.

Iterate continuously

As you annotate, it's inevitable that you'll find new edge cases, a lack of clarity between different classes, or even problems with the quality of your raw data. Resolving these will make your dataset stronger. Make the most of these solutions by disseminating the information throughout your team and updating your gold standard to reflect any changes.

Common annotation errors by task type

Certain tasks have issues that often crop up during annotation. Get a headstart by resolving them straight away.

TASK TYPE	PROBLEM	EXAMPLE
Image annotation	Image quality	
	Occluded / partial subjects	
	Overlapping annotations	
Sentiment analysis	Personal sentiment over text sentiment	
Content moderation	Cultural differences	
Audio classification	Speaker demarcation Background / ambient noise	
Entity annotation	Context	
Machine translation	Terminology	
Training phrases	Duplicates	
Video annotation	Inconsistency	

Maintaining quality

While there are a lot of potential issues at each stage of a machine learning project, there's no big secret to building high-quality datasets. If you can prepare for and resolve most of the issues we've outlined above, you'll be well on your way to developing a competent model. In the end, refining quality is an ongoing process. Treat it as such and build out quality processes for each stage of the life cycle. You might be surprised at how much your results improve.

Where can I get more training data?

There are three main ways to get training data for machine learning projects. The first is to explore free options via open datasets, online machine learning forums and dataset search engines. The second is to evaluate your internal options and see if there is a way to repurpose the data you already have. Finally, the last and often most efficient option is to outsource training data services from a third party.

Free options

Pro	Cons
• Free • Relatively easy to find raw data	• Requires internal work hours to find data • Requires internal work hours to preprocess the data to suit your model • Some open source data can't be used for commercial purposes

Most of the time, open datasets consist of information that is publicly available through government sites or social media. While there are an increasing number of useful open datasets available online, there will be times where free options can't get you the training data you need. Luckily, there are other inexpensive ways to create custom datasets for your specific use cases.

Internal options

Pro	Cons
• No need to consult third-party companies • Minimal costs to find raw data	• Requires internal work hours • Must coordinate with multiple departments • Data augmentation requires highly-trained data scientists to implement properly

Before opting to outsource training data services, you should first check to see what in-house options you have available and if they'll help you to create the datasets that you need. For example, if you're building a chatbot to handle online inquiries, you should get in touch with your customer service department to see if they have stored chat logs or email threads you can use to train your model. Of course, data availability depends highly on the problem you are trying to solve with your machine learning project.

Before you look for datasets elsewhere, you should try to repurpose the data you already have to build a larger dataset. One common way to do this is through data augmentation. For image datasets especially, there are numerous simple ways to increase your training data through simple image rotations, color contrasts and other image manipulations.

Paid options

Pro	Cons
• Quick turnaround times • All project management and QA tasks are handled for you • Data augmentation requires highly-trained data scientists to implement properly	• Outsourcing can be expensive

Sometimes free and internal options aren't able to provide you with machine learning datasets at the scale and quality you require. In these cases, it's often more efficient to simply outsource training data from a data annotation company rather than building a data collection and annotation infrastructure on your own. Luckily, there are a variety of training data outsourcing options available to you.

Outsourcing data collection

One option is to partner with a data collection company. For example, if you are building a voice recognition system and you require voice samples from 200 different people, you could simply hire a company to record the audio files for you.

One of the main advantages of this method is that the data collection company will handle all of the project management tasks for you. From finding and training contributors to reviewing the data for accuracy, your project is completely managed by the training data company. All you need to do is provide specific guidelines.

Outsourcing data annotation

If you have the data, but don't have the tools or workforce to annotate the data internally, you can offload all of your annotation tasks by partnering with a data annotation company. These companies can provide the raw data itself, a platform for labeling the data and a trained workforce to label the data for you. Companies like TELUS international already have platforms built to collect and annotate data, as well as a large, trained workforce that can annotate hundreds of thousands of data points at scale.

Once again, the main advantage of partnering with a data annotation company is that you don't have to deal with building a data annotation infrastructure from scratch. All you have to do is build specific guidelines and QA protocols for the company to follow.

About TELUS International AI Data Solutions

Incorporating AI into your business comes with a lot of boxes to check: gather data, but not too much or too little. Make sure the data is properly formatted, but not biased. Determine how you want to label the data, while simultaneously avoiding errors. The list is long and extensive, which is why an experienced partner in AI data solutions can help guide the way.

TELUS International is a leading global provider of scalable data solutions for text, image, video, audio and geo data, as well as 3D sensor fusion to train computer vision models.

Our proprietary AI training platform, in combination with our AI Community of more than one million professional annotators and linguists, has enhanced AI systems across a range of applications, from advanced smart products, to better search results, to more human-like bot interactions and more.

Connect with us to learn how our **AI Data Solutions** experts can customize the exact project to advance your machine learning needs.

telusinternational.com

TELUS® International