The Essential Guide to Training Data

Improving Your AI Models with High Quality Training Data

An Appen eBook

appen

Table of Contents
The Essential Guide to Training Data

Introduction 3

How Do Machines Understand the World? 4

Sourcing the Right Data 6

Why Data Needs to be Labeled 7

The Unreasonable Effectiveness of Data 9

How to Make the Data You Have the Data You Need 11

What You Should Know About Training Data 14

- How Much Training Data You Need 14
- How to Reduce Bias in Your Training Data 16
- Why You Need Separate Training and Validation Datasets 17
- Protecting Your Data 17
- Evaluating and Maintaining Your Models 18
- How Training Data Can Solve for Overfitting 19

Our Capabilities 20

About Appen 21

Introduction

It's common knowledge that every machine learning solution needs a good algorithm powering it. Plenty of ink is spilled on tech sites about advances in machine learning methods like deep learning and how the newest models are driving business success for everything from personalized shopping to national security. There are plenty of analogies you can use to drive this point home. Data is the oil, the model is the car. Data is the ingredients, the algorithm is the recipe. The point is: neither works without the other.

In this guide, we'll cover everything you need to know about creating and updating the training data necessary to drive and scale successful machine learning programs.

The Importance of Data in AI Innovation

No matter how cutting edge a model happens to be, it's useless without enough high-quality training data. In fact, when you dig into the history of the major breakthroughs in AI, many of them were preceded by voluminous datasets - Human-level spontaneous speech recognition, IBM Deep Blue's chess win against Garry Kasparov, IBM Watson's win at Jeopardy!, Google's object classification, DeepMind's win at Go!, GPT-3 general purpose model, Deepfakes, Quantum protein folding with AI and more.

How Do Machines Understand the World?

The short answer:
from labeled examples.

Fei Fei Li, the woman behind ImageNet, the most famous and widely cited dataset for computer vision projects, once gave a talk where she described teaching her daughter what a dog was.

When a child is born, of course, it has no idea what a dog (or really anything) actually is. When a toddler sees a golden retriever for the first time and her parent says, "that's a dog," she now has a word for that four-legged furry thing that's running around, hoping to get pet. She can observe how the dog moves, how it behaves, all the while knowing: this is a thing that's called a dog.

But say she sees a cat. Well, it's got four legs and it's soft. It wants to be pet. She very well may assume that's a dog. Her parent can teach her differently though, telling her what this new animal is called and pointing out how it looks and behaves differently than the concept of "dog" she's already learned. The cat is smaller, for example. It purrs. It's not going to bring that tennis ball back to you.

Now, this young girl has, for all intents and purposes, new training data. A cat has been "labeled" by her mother and she can observe it, understand what distinguishes it from a dog, and so on.

Broadly speaking, for supervised learning, this is how machines understand things. They learn from labeled examples. ImageNet, the dataset we mentioned above, is a massive library with examples of everyday objects, from chairs to pizzas to, yes, dogs and cats. And it's widely cited as the basis for a vast amount of computer vision projects, both professional and academic.

This concept doesn't just apply to computer vision, though. It holds true across nearly every successful machine learning deployment:

For natural language processing (NLP), you need contextual examples to teach a machine what words mean. After all, the sentences "that burrito was so bad" and "I want a burrito so bad" share a lot of common words but mean vastly different things. Labeling the words or utterances in the sentence or the sentiment of the statement are the examples a machine needs to make sense of similar language.

Labeled examples teach a machine that "CNN" can mean the cable channel in one context and "convolutional neural net" in another. It's really about the context in which you find that abbreviation and the language that surrounds it that clue in human listeners. Machines need to be shown those rules as well.

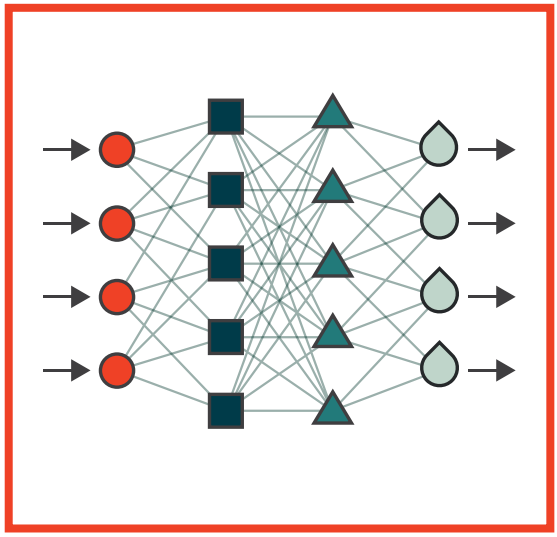
The same goes for audio. We've gotten good at understanding different vernaculars and accents, but that's because we've heard them and learned to decipher the nuances of different speech styles. An audio assistant trained by AI practitioners in California will almost certainly understand a mom in Silicon Valley when she's verbally ordering something. But until it's heard a woman in Alabama order the same, it might have trouble understanding just what she's saying. The same might be true for Australians, people from New Zealand, or British people verbally ordering similar things through a virtual assistant. Global customer bases come with global challenges.

Simply put: identifying anything–a word, an object, a sound, a product, a moving image–requires examples. None of us knew what a dog looked like until we were told and given an example, then internalized, learned, and extrapolated from it.

Machines need the same things. The big thing to remember is: **they need a lot more examples.** And those examples need to be meticulously labeled. And they need to be labeled by people from diverse backgrounds so that the AI we build is inclusive and works for everyone.

CNN – News

VS.



CNN – convolutional neural net

Sourcing the Right Data

Now you understand that machines need a lot of examples to make accurate predictions. The next question is: where do you get those examples? First, identify the data your project requires. Data scientists can help figure this out and usually the project itself will dictate what you need. For example, a model that needs to identify cancerous cells in medical images will need a ton of images containing both cancerous and non-cancerous cells so it can learn to tell the difference.

Once you know what kind of data you'll need, figure out what data you already have (if any). If there are gaps, you'll need to look for off-the-shelf datasets and/or a third-party source for the rest of your data—and ideally a steady source so you can continue to train your model as needed. Ask yourself these questions:

- Can I afford all of the data I need?
- Do I have a consistent source of high-quality data?
- Does the data I have and/or plan to source cover all the use cases of my model?
- Do I understand what's in the data I have?

If you're still short on data, there's one other solution to consider: synthetic data. That's where you (or you working with a third-party) create artificial data to fill in the gaps and make sure all of your use cases are covered. In any case, you'll want to make sure the data you're collecting is high-quality: that means it's not messy, irrelevant, or incomplete. You can set up a process to clean up your data if needed.

Why Data Needs to be Labeled

The short answer:
so that a machine can actually understand it.

In order for a self-driving car to "see," "hear," "understand," "talk," and "think," it needs video, image, audio, text, LIDAR, and other sensor data to be correctly collected, structured, and understood by machine learning models.

Breaking this down to just what a car "sees" requires annotating many images so that a model can learn and understand all the different street signs under all conditions. While speed limit signs may have the same shape, the car must also interpret the number on the sign to drive safely. A car must also be able to "understand" what a person is – including an adult, a kid, and a baby, for example. To do this, pictures of many different people must be shown from all different angles so that it can start to say what is and is not a person. And this becomes complicated fast: the car also needs to differentiate between, say, the reflection of a person in a window or a poster of a person's face vs. an actual pedestrian.

To break it down further, a picture is simply a series of pixels to a machine. Those pixels have values that correspond to colors but those pixels don't have values that represent the object – just a tiny dot on a massive canvas of other pixels. But labeled images show machines that certain collections of pixels are certain objects.

> Our project is still in the pilot phase, and we needed to speed up the cycle to reach production, which requires training data that rapidly meets our algorithm requirements. The annotation tool, including 3D LiDAR, high-quality control features, and workflows, is already built into the Appen platform. This is helping us ensure the process is optimized based on our project requirements, enabling a smooth collaboration between our team and the Appen team. We are looking forward to moving this internal pilot into production.

ECARX

Senior Project Leader
An automotive technology company building an intelligent, connected platform for multiple vehicle models.

Let's go back to ImageNet. Every image in that dataset was labeled by a person. The end result: thousands of examples of different objects. From those labels, machines can make sense of the pixels of which they're made up.

Now, image labeling can be done in many different ways. You can run rudimentary labeling tasks like "is there a dog in this picture," (a form of classification) but it's going to take a ton of images for a machine to start to understand that dataset. It's usually better practice to use one of the following common labeling techniques:

- Object detection: Use bounding boxes, dots, and lines to identify target objects in an image
- Semantic segmentation: Assign each pixel to a category, such as pedestrian or sign

Video annotation can leverage the same labeling methods, with the use of software to help annotators label entire videos quickly rather than frame-by-frame. Audio and text annotation use different labeling techniques, like sentiment annotation (where the labeler decides the attitudes and emotions behind the audio or text). Overall, there are many different annotation methods depending on the type of data you're working with.

Generally speaking, the more examples a machine sees, the better it understands. This usually holds true no matter the use case—images, text, audio, what have you.

The point is that the data you have likely isn't the data you need to create effective machine learning algorithms. It's far more common that the data you have needs to be labeled or annotated in some way, shape, or form so that a machine can actually understand it. And the more labels a piece of data has, the more complicated an ontology it can create.

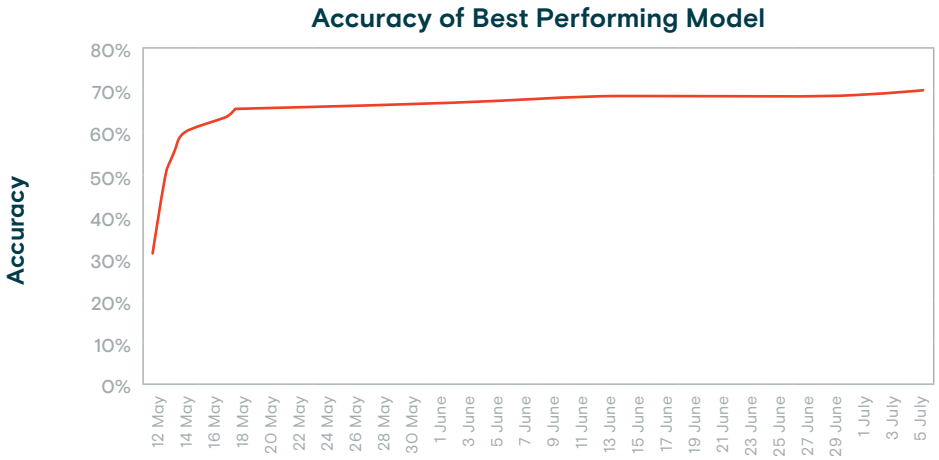
So far, we've covered two important concepts:

- Machines understand the world by learning from examples
- Humans need to label data so a machine understands what the examples mean

The Unreasonable Effectiveness of Data

A while back, we sponsored a competition on Kaggle. Kaggle, in case you aren't familiar, is a site where organizations can run data science competitions and practitioners compete to craft the most effective algorithms. These contests range from fun stuff like "Santa Gift Matching Challenges" to more professional-grade model building like speech recognition on TensorFlow or object detection. Teams constantly update their models, and, with contests running over weeks, some teams will submit hundreds of models just by themselves.

Our competition was about creating a search relevance algorithm from a modestly-sized dataset of semi-random products. But the particulars of that dataset aren't really what's relevant. Instead, look at what happened to the top algorithm accuracy over time:



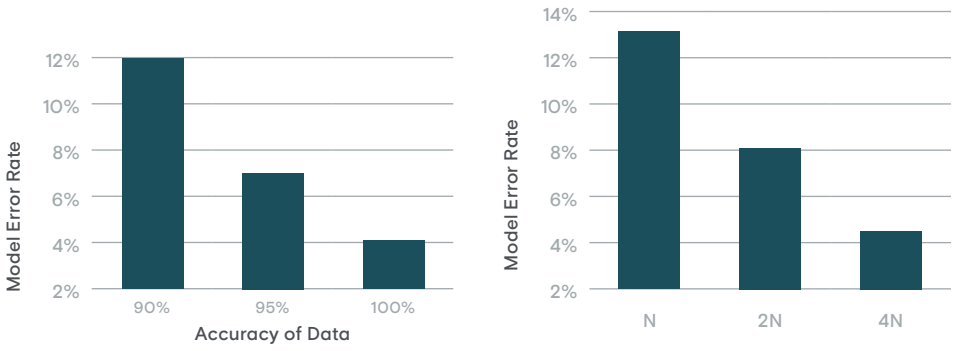
See that? In the first throes of the competition, models improved immensely as smart data scientists found features and iterated on promising avenues. But as time wound on, their models barely got better, improving by fractions of fractions of a percent.

So why exactly did that happen? Simple. Because that's as good as the algorithm could get with the data it had.

At a certain point, without the inclusion of more and better training data, models will simply plateau. Yes, you can find marginal improvements, always, but those improvements are of vanishing importance.

Of course, that doesn't mean your models can't get more accurate and confident. Because even when you've squeezed what you can out of the data you have, marked improvement comes from, you guessed it, more training data.

Now Take a Look at This:



Both graphs show the benefit of data quantity and quality on the same algorithm. A classifier with an error rate of 13% gets nearly twice as accurate with twice as much data. If you give it four times as much data? Its error rate drops to less than 5%. And the same general principle holds for cleaner, more accurate data.

This is exactly what Peter Norvig, Director at Google Research, was referring to when he wrote about the unreasonable effectiveness of data.

The best data scientists working with the same data will arrive at very similar solutions. That's what Kaggle contests–and not just ours, mind you–show time and time again. But more and cleaner data are simply more effective than better algorithms. **Nothing helps machine learning projects more than improving the data they run on.**

But not all data is created equal.

How to Make the Data You Have the Data You Need

The short answer: by labeling it.

The reality is, most data is messy or incomplete. At least as it applies to machine learning, that is.

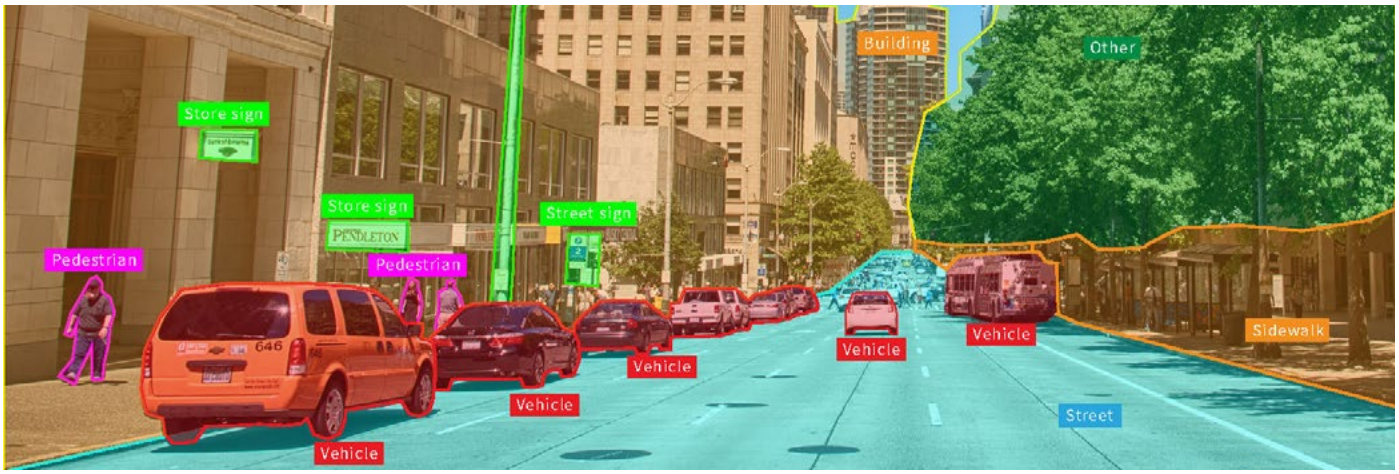
Take a picture for example. To a machine, an image is just a series of pixels. Some might be green, some might be brown, but a machine doesn't know this is a tree until it has a label associated with it that says, in essence, this collection of pixels right here is a tree.

If a machine sees enough labeled images of a tree, it can start to understand that similar groupings of pixels in an unlabeled image also constitute a tree.

Not every piece of data is like this. Take the example of the corpus of grand master chess games. That data is plenary. It's complete. Nothing needs to be added because the data itself is a series of chess moves and a series of chess moves is the entirety of the game.

But that's simply not the kind of data most machine learning projects are built upon. Sentiment algorithms need in-domain labels to understand the vernacular and conventions of a platform. Audio data needs to be translated to words that a machine can more easily understand so that it can make sense of the myriad ways people say the same words and phrases.

So how do you prepare training data so that it has the features and labels your model needs to succeed? **The best way is with a human-in-the-loop. Or, more accurately, humans-in-the-loop**.



Adobe

Take Adobe, for example. One of Adobe's flagship offerings is Adobe Stock, a curated collection of high-quality stock imagery. The library itself is staggeringly large: there are over 200 million assets (including more than 15 million videos, 35 million vectors, 12 million editorial assets, and 140 million photos, illustrations, templates, and 3D assets). Every one of those assets needs to be discoverable.

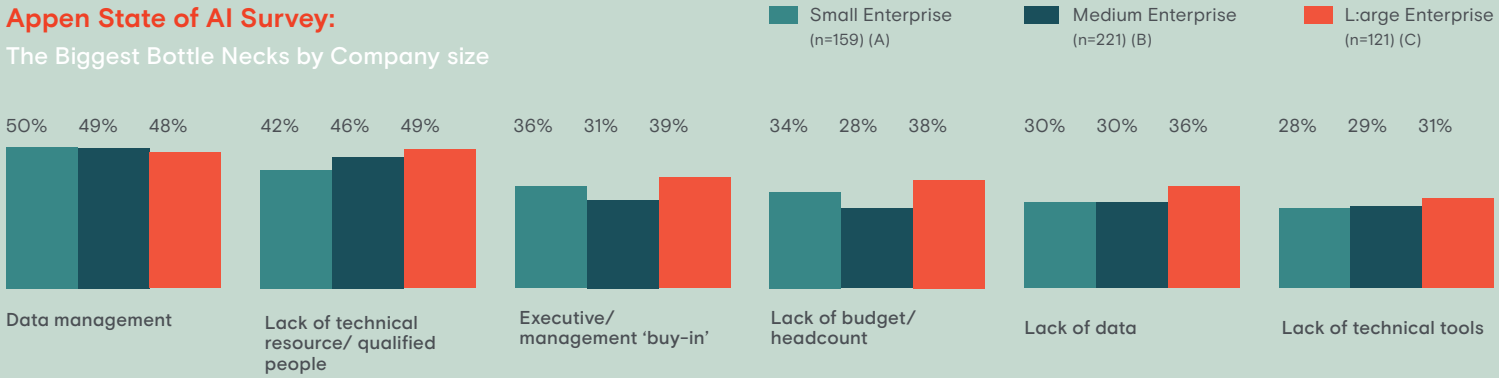
Adobe, of course, has plenty of metadata for their searches that's provided by content uploaders. They provide their own information like the objects in the image, the mood, the aesthetic, and more. But that isn't quite enough. For starters, those user-provided tags can sometimes be over-broad or incorrect. And most importantly, they don't speak to the way end-users actually utilize these images in marketing collateral.

For example, many Adobe Stock customers are looking for images that allow them to place text over an image. That requires a certain type of image, where copy can sit on a clean background free of busy extra objects. These types of images are quite popular for marketers looking to create clean, vibrant collateral.

The issue is that while these particular images tend to be some of the most frequently downloaded assets, that feature isn't something that exists in the metadata Adobe's uploaders provide. To better serve their customers, Adobe needed to create a model that could find key attributes in images like copy space or object isolation.

That's where Appen can help. We've been preparing training data for machine learning for 25 years, and we understand how to leverage human intelligence at scale to get you the training data you need for your AI projects. We've worked with video, images, audio, text, sensor data, and more. Chances are, if you need more and better training data, we can provide it for you with machine-learning powered annotations by our crowd of over 1M contractors all over the world.

Appen State of AI Survey:
The Biggest Bottle Necks by Company size



One of the biggest bottle necks in building AI is getting access to the right kind of data. This is why organizations of all sizes need to be mindful of how they equip their most precious assets – their machine learning and data science teams – with the right data partners. Labeling projects are complex and time-consuming, and your teams didn't start working on AI problems to spend time on repetitive data work. But as we've learned, you need a lot of quality data to make smart models. And even if your model is already performing, more data will make it perform better. After all, data is unreasonably effective.

Who's annotating your data?

If you want your data scientists to spend their time doing the work they actually went to school for, you're going to need additional people to serve as your humans-in-the-loop. What's more, you want these people to be as diverse as possible. Ideally, you want them to have different demographics, geographies, experiences, education, and backgrounds. Why? Because that way, you'll reduce the chance of one-sided perspectives being introduced into your data.

If you have only one person labeling all of your data, for example, those labels are going to reflect that person's bias (and remember, we all have bias). But if you have multiple, very different people labeling your data, suddenly you're getting a range of judgments that lead ultimately to more accurate labels. As much as you can, try to recruit a diverse group of individuals for your human-in-the-loop annotation process.

So far, we've established:

- Machines learn from examples (a.k.a. training data), just like people
- Training data is as important–if not more important–than the algorithm itself
- High quality and high quantities of training data are the surest way to improve models
- Training data needs labeling to be truly useful
- Training data labeled by a diverse group of people is the most accurate way to do this

Let's dig in a little deeper though.

What You Should Know About Training Data

We've been in the training data business long enough to not only hear the most common questions and concerns but to solve them. We'll start with some basics and move into some more technical bits later on.

How Much Training Data You Need

Anyone who's dealt with a lawyer will recognize the classic response here: it depends.

There are a lot of factors in play for deciding how much training data you need. First and foremost is how important accuracy is.

Say you're creating a sentiment analysis algorithm for a social media platform, one that analyzes people's attitudes toward a new Hollywood movie. In other words, your algorithm is evaluating social media posts to see if they're overall positive, negative, or neutral toward the movie. Your problem is complex, yes, but it's not a life or death issue. A sentiment algorithm that gets to 85% or 90% accuracy is more than enough for most people's needs and a false positive or negative here or there isn't going to substantively change much of anything.

Now, a cancer detection model or a self-driving car algorithm? That's a whole different story. A car that's 85% or 90% safe is actually remarkably unsafe and should never see the road. A cancer detection model that could miss important indicators is literally a matter of life or death.

Of course, more complicated use cases generally require more data than less complex ones. A computer vision model that's looking to only identify foods versus one that's trying to identify objects generally will need less training data as a rule of thumb. The more classes you're hoping your model can identify, the more examples it will need.

Which is all to say: the amount of training data you'll need is contingent on the complexity of your ontology and how necessary high levels of accuracy are.

Interestingly, that cancer model we teased above? That needs far less data than an autonomous vehicle model will need. That's because a cancer classifier is looking at cells whereas a self-driving car model will need to identify everything from other cars and pedestrians to street signs and medians, among countless other classes. Though the cancer model's accuracy is incredibly important, it should need far less data to get to an acceptable accuracy threshold than one looking to drive on city streets.

It's worth resurfacing something here: there's really no such thing as too much data. Better training data, and more of it, will improve your models. You need to set the threshold for success, but know that with careful iterations, you can exceed that with more and better data.

CallMiner, the pioneer of the artificial intelligence (AI)-powered speech analytics space, works with audio from customer service calls, which is then processed through a speech recognizer. The CallMiner Research Lab is tasked with training AI models to understand those conversations, including sentiment and emotions, and other relevant insights between organizations and their customers. In one of their largest projects to date, the CallMiner Research Lab worked on accurately identifying and analyzing the sentiment in customer interactions, negative, positive, and neutral.

Before working with Appen, the CallMiner Research Lab could only parse through 3,000 or so data samples in a few months. Now, they've analyzed tens of thousands of data samples. The Appen platform lets them process more calls faster and with more accuracy, enabling them to expand their customer base and explore new types of conversation insights with the extra time saved.

By leveraging global annotators, CallMiner gained a greater diversity of perspectives, helping the research team to expand definitions in new ways to achieve better analysis. The Appen platform also supports CallMiner's efforts in documenting the decision-making process behind its models, an essential step toward contributing to the explainability and overall responsible use of the model.

How to Reduce Bias in Your Training Data

Bias is a real problem in AI. There's a lot of biased data out there and because data is the foundation of algorithms, there's also a lot of biased algorithms. Even the top facial recognition models built by companies with plenty of resources were found to have trouble identifying the full array of skin tones . And there are many other examples of AI gone wrong in the media.

So what can you do? It starts with making sure your data is unbiased.

We mentioned earlier that we all have some level of bias, and that's why it's so important to have a diverse group of people annotating your data. But bias also comes into play during data sourcing. For example, if you're building a model for a voice assistant like Siri or Alexa, your model needs to recognize all of the languages your customers will use, including various accents, dialects, and tones. But if most of your audio data is of English spoken in one part of the world, your model will perform much better for English speakers from that location, and a lot worse for just about anyone else. Instead, you need to collect a wide breadth of language data that covers speech for all of your potential end users.

Why You Need Separate Training and Validation Datasets

Typically, when you're building a model, you split your labeled dataset into training and validation sets (though, sometimes, your validation set may be unlabeled). And, of course, you train your algorithm on the former and validate its performance on the latter.

What happens when your validation set doesn't give you the results you're looking for? If you're like most folks, you'll update your weights, drop or add labels, try different approaches, and retrain your model. But when you do this, it's incredibly important to do it with your datasets split in the exact same way.

Why is that? It's the best way to evaluate success. You'll be able to see the labels and decisions it has improved on and where it's falling flat. Different training sets can lead to markedly different outcomes on the same algorithm, so when you're testing different models, you need to use the same training data to truly know if you're improving or not.

Protecting Your Data

Data security has to be factored into your training data management strategy. Why? Sometimes it's obvious, like when you're working on a healthcare model that uses personal health information that legally must be protected. Even in less obvious cases, though, you should still have security controls in place to keep valuable business information secure and/or to protect the privacy of the people you're sourcing data from or about.

You have a few options to get you started on better data security: use a secure data platform (either your own, or one from a vetted third-party), anonymize personal data if needed, and require your human annotators to sign confidentiality agreements in more sensitive use cases.

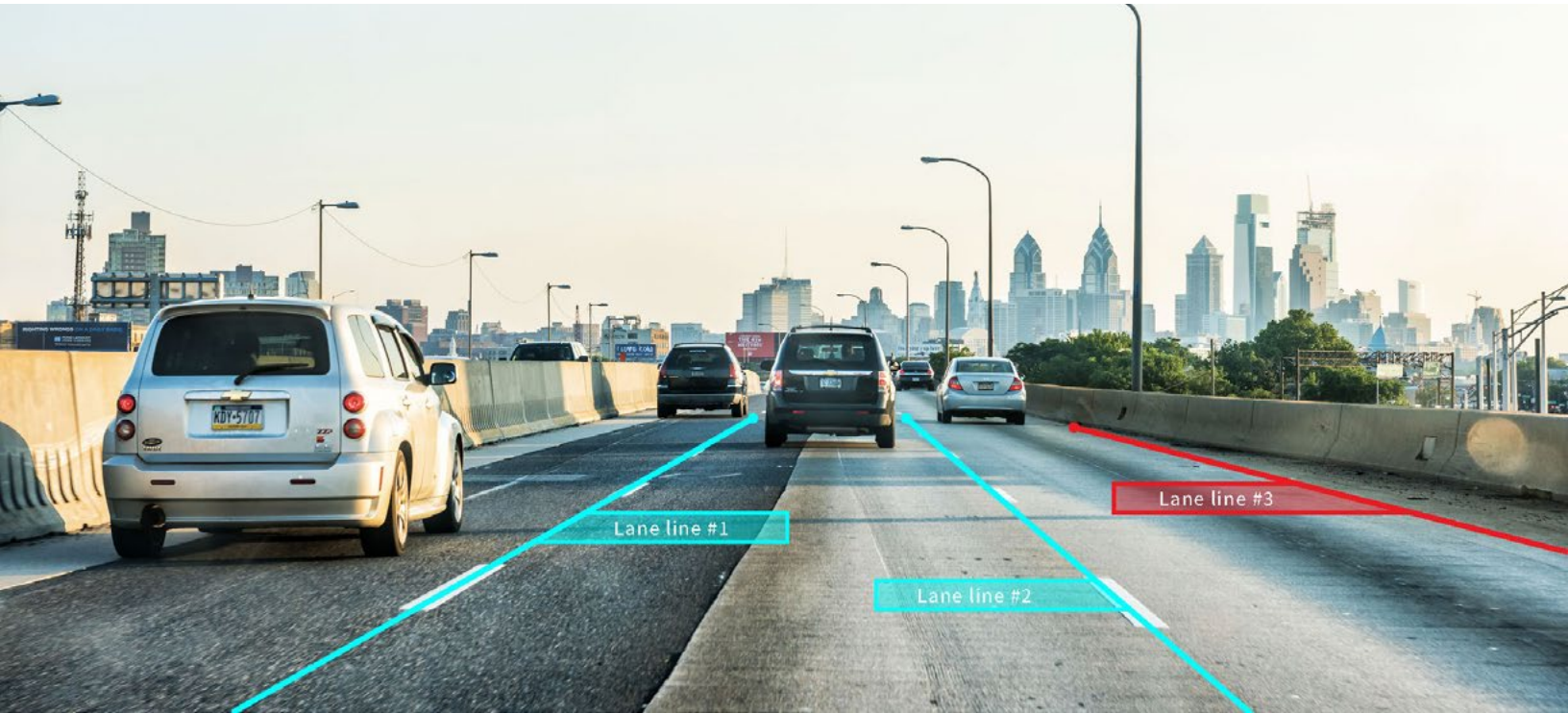
Evaluating and Maintaining Your Models

Your training data won't have equal amounts of every category you're hoping to identify. A sentiment classifier won't be split 33% positive, 33% neutral, and 33% negative. It's just not how randomly selected training datasets work. And the more of a particular label you have, the higher the chances it will perform well on that particular label. To use a simple example: if your computer vision algorithm sees 10,000 instances of a dog and only 5 of a cat, chances are, it's going to have trouble identifying cats. As we've covered, machines need training data as examples to learn from. Without them, it simply has trouble learning.

The important thing to keep in mind here is what success means for your model in the real world. If your classifier is really just trying to identify dogs, then it's less-than-stellar work on cats or fish identification is probably not a deal-breaker. But you're going to want to evaluate model success on the labels you'll need in production.

Now, what happens if you simply don't have enough information to reach your desired accuracy level? Chances are, you'll need more training data. Models built on a few thousand rows are generally not robust enough to be successful for large-scale business practices.

That's why building a strong data pipeline that's both repeatable and scalable is a key investment. You're going to need to retrain, test, and tune your model again and again to either get to your desired performance level or to teach your model new concepts that it encounters in the real world. With a training data pipeline in place, you can create high-quality data quickly to match your model's retraining needs.



How Training Data Can Solve for Overfitting

If you have too much of a certain label in your training set, you could run into problems with overfitting. Let's say you are creating a model that identifies cats and dogs, and you have 10,000 dog instances and a handful of cat labels out of, say, 12,000 data rows total. If you push your model into production, chances are, it's going to assume a lot of animals are dogs. That's because, for your model, somewhere close to 80% of its training data is dogs. It's going to see dogs everywhere.

A more even distribution of training data helps. So does increasing your training data, adding other labeled rows that aren't part of the overfit class. In other words, show your model more cats and it will start to learn the distinctions between a class it already understands and one it doesn't. With retraining and additional (or better) data, you can often solve your overfitting problem.

It's worth pointing out that active learning practices can help here. When your model is a bit overconfident about a certain class, using human judgments to correct it can be a big help. Remember: human-in-the-loop machine learning should never mean "just label some training data." You can also test and tune your algorithms with human judgments.

Want to jumpstart your AI project with a dataset? Visit Appen's Off-The-Shelf Dataset Library at appen.com for high-quality licensable datasets and our recommendations for open source datasets.

Our Capabilities

At Appen, we understand training data. We've labeled billions – with a "B" – of rows of data. We've annotated millions of images. We've supported hundreds of real-world machine learning projects. We can help prepare your data for whatever sort of initiative you want. What's more, we can evaluate your algorithms to help tune them with active learning.

Why Appen

Platform

Our platform collects and labels images, text, speech, audio, video, and sensor data to help you build, train, and continuously improve the most innovative artificial intelligence systems. In addition to specialized and precise tooling, we offer several machine learning assisted tools to enhance quality, accuracy, and annotation speed.

Crowd

To produce the volume of training data required to confidently deploy world-class models, you'll need an army of contributors and an experienced crowd management service to ensure annotators are identified and certified to your specifications. We are proud to offer a crowd of over one million contributors, in over 170 countries, and supporting over 235 different languages.

Expertise

With over 25 years of experience scoping and delivering more than 6,000 ML projects, we understand the complex needs of today's AI projects. Our solutions provide the quality, security, and speed used by leaders in technology, automotive, financial services, retail, manufacturing, and governments worldwide.

Types of Training Data

Text

Deploy text-based natural language processing with data that's collected, labeled, and validated in a wide array of languages.

Images

Add computer vision to your machine learning capabilities by collecting and understanding image classification, or leveraging pixel labeling semantic segmentation.

Audio

Build interfaces that process audio with data that is collected as utterances, time stamped, and categorized across more than 235 languages and dialects.

Video

Combine the best of audio and image annotation to process video and turn it into actionable training data for machine learning. Teach your model to understand video inputs, detect objects, and make decisions.

Sensor

Leverage even more data points by annotating data coming directly from sensors and enable machine learning models to make decisions on a variety of data sources including LiDAR and Point Cloud Annotation.

About Us

Appen collects and labels images, text, speech, audio, video, and other data used to build and continuously improve the world's most innovative artificial intelligence systems. Our expertise includes having a global crowd of over 1 million skilled contractors who speak over 235 languages, in over 70,000 locations and 170 countries, and the industry's most advanced AI-assisted data annotation platform. Our reliable training data gives leaders in technology, automotive, financial services, retail, healthcare, and governments the confidence to deploy world-class AI products. Founded in 1996, Appen has customers and offices globally.

- Experience working in **170+ countries**
- Expertise in **235+ languages**
- Over **1,125 employees** located in nine offices around the globe
- Access to a curated crowd of over **1 million** flexible contractors worldwide
- Nearly **1 billion judgments** made and **3 million images and videos** collected in 2020
- **25 years working** with leading global technology companies
- Over **7,500 AI projects** scoped and deployed to date

appen