

A practical guide to automated data labeling

Snorkel

Introduction

Today, the most significant blocker enterprises face in using AI is not models or infrastructure, but is the vast volumes of labeled "training data" that AI models learn from. For example, before using Snorkel, data science and legal teams at a top-3 US bank spent over six months labeling the data needed to train a machine learning model to automate loan document processing. The time and effort required to label data blocks even the largest organizations globally from scaling AI, especially when data is complex, fast-changing, or needs to be kept private.

Snorkel AI's mission is to help enterprises overcome this manual data labeling bottleneck with an automated approach called programmatic labeling. Snorkel AI's founding team developed this radically new method with over five years of research at the Stanford AI Lab. It has resulted in 50+ peer-reviewed publications, including course syllabi and textbooks. Snorkel's technology has been deployed by organizations like Google, Intel, Apple, Stanford Healthcare, and DARPA.

With Snorkel, users create labeling functions (e.g., looking for keywords or patterns in a loan document) to programmatically label millions of data points in minutes, rather than manually inspecting and labeling one data point at a time. Snorkel reduces the time, cost, and friction of labeling training data so enterprises can more easily build and scale AI models to deploy more meaningful applications. Incorporating human judgment into the AI process through subject-matter experts becomes more efficient and scalable and can lead to more ethical, responsible outcomes. For example, that same top-3 US bank now builds governable AI applications in 24-48 hours rather than six months.

Snorkel Flow, our end-to-end AI development platform based on programmatic labeling, is used by Chubb, BNY Mellon, and other Fortune 500 enterprises in biotech, insurance, and oil & gas sectors, and several government agencies to accelerate AI development by 10-100x speedups, unlock seven-to-eight figure ROI, and scale AI.

In this ebook, we discuss the following topics.

- Moving from manual labeling to programmatic labeling
- The key difference between manual and programmatic user inputs
- Manual labeling optimizations such as model-assisted labeling and active learning
- Automating data labeling with programmatic approaches
- Using Snorkel Flow to automate data labeling
- Accelerating AI with Snorkel Flow

Moving from manual to programmatic labeling

Labeling training data by hand is exhausting. It's tedious, slow, and expensive—the de facto bottleneck most AI/ML teams face today ¹. Eager to alleviate this pain point of AI development, machine learning practitioners have long sought ways to automate this labor-intensive labeling process (i.e., “automated data labeling”) ², and have reached for classic approaches like active learning or model-assisted labeling.

The Snorkel team has spent over half a decade focused on this problem, first at the Stanford AI lab and now at Snorkel AI—hosting workshops ³, working with collaborators in academia, industry, and government ⁴, and publishing 50+ of papers on the subject ⁵. During this time, we have used firsthand a wide variety of techniques (including all those mentioned in this article) that are aimed at automating parts of the labeling process. Through our experience, we have found these partial optimizations of manual labeling simply don't provide the magnitude of acceleration modern AI development needs to be practical. It's what drove us to develop programmatic labeling, an approach that makes automated data labeling a reality.

In this ebook, we'll outline techniques for optimizing manual labeling and compare them to programmatic labeling. In summary:

- **Manual labeling** optimizations such as active learning and model-assisted pre-labeling speed up the process of labeling individual examples incrementally. They are useful, classic approaches, but suffer from all the same pitfalls as vanilla manual annotation (lack of scalability, adaptability, governability, etc).
- **Programmatic labeling**, on the other hand, unlocks a much more scalable approach to automate data labeling, accelerating labeling 10-100x (instead of 10%) by using a fundamentally **more efficient and reusable type of user input than manual labels: labeling functions**. This enables teams to label far more data, adapt to changing needs, and inspect or correct bias in the labeling process in a much more practical way.
- **Snorkel Flow** is a data-centric platform for building AI applications where programmatic labeling, active learning, and many other modern ML techniques are integrated and made accessible. Because all these techniques are compatible with one another, data scientists and subject matter experts can utilize every automated data labeling advantage, big and small.

The key difference: manual vs programmatic user inputs

The primary difference between manual labeling (with or without optimizations) and programmatic labeling is the type of input that the user provides. With manual labeling, user input comes in the form of individual labels, created one-by-one. With programmatic labeling (a form of weak supervision described in greater detail later in this ebook), users instead create labeling functions, which capture labeling rationales and can be applied to vast amounts of unlabeled data to create individual labels automatically.

This difference in input type has huge implications for the scalability, adaptability, and governability of the labeling process.

Scalability

Programmatic labeling allows domain knowledge to be shared more directly and efficiently

Labeling data by hand is like sharing your knowledge via a game of twenty (thousand) questions ⁶: you ask experts to label data (using heuristics and knowledge they have in their heads) and then spend a bunch of additional time, energy, and money having an ML model try to statistically re-infer that same knowledge!

A major benefit of transferring knowledge via labeling functions is that it allows the user to incorporate that information directly. The result is a much more efficient transfer of knowledge, saving orders of magnitude in time and money in many cases ⁷, and allowing a domain expert to more directly affect the behavior of a model.

Adaptability

Programmatic labeling allows training labels to be easily updated or reproduced

When requirements change, data drifts, or new error modes are detected, training sets need to be relabeled. With a manual labeling process, this means manually reviewing each affected data point a second, third, or tenth time, multiplying the cost of both time and money to develop and maintain a high quality model.

When you produce your labels programmatically, on the other hand, **recreating all your training labels is as simple as adding or modifying a small, targeted number of labeling functions and re-executing them**, which can now occur at computer-speed, not human-speed.

Governability

Programmatic labeling makes the training set generation process easier to inspect and correct

In most manual labeling workflows, users leave no record of their thought process behind the labels they provide, making it difficult to audit what their labeling decisions were—both in general, and on individual examples. This presents a challenge for compliance, safety, and quality control.

With programmatic labeling, every training label can be traced back to specific inspectable functions. If bias or other undesirable behavior is detected in your model, you can trace that back to its source (one or more labeling functions) and improve or remove them, then regenerate your training set programmatically in minutes.

Automating the right things at the right level

While the goal of automated data labeling is clear (automate away the slow, tedious, expensive parts of labeling), the scope of what exactly can or should be automated is more nuanced. Various techniques have been proposed to make it slightly more pleasant or efficient to provide individual labels (smoother interfaces, model-assisted pre-labeling, active learning, etc.), but the fundamental limitations remain, because the input is the same: individual labels, collected one at a time, without rationale.

On the other hand, it's also possible to attempt to automate too much. The idea that training data labeling can be turned into a completely push-button process without any human input (essentially getting something from nothing) is another variant of the ill-fated Free energy machine ²⁰—which is to say: it won't work! The human cannot be pulled out of the loop entirely—they've got the domain knowledge the model needs.

The key to making automated data labeling work in practice is to raise the level of abstraction at which data scientists and domain experts label their data—in other words, the human is moved neither in (at the center of the labeling bottleneck) nor out (no longer guiding the labeling process), but *up*—**transferring domain knowledge via higher-level and fundamentally more scalable inputs than individual labels: labeling functions that capture labeling rationales.**

With this important distinction in mind—let's now take a closer look at a few manual labeling techniques in more detail.

Manual labeling workflow optimizations

Manual labeling

Let's begin with the baseline: vanilla manual labeling with no automation.

Lots of great things have been accomplished with manually labeled training sets! But some of those datasets have also taken person-years of time to collect (looking at you, ImageNet ⁸), and the vast majority of machine learning projects don't have that kind of time or budget.

On top of being slow (and expensive), manual labeling also scales poorly. With the exception of small gains from becoming more familiar with the dataset and annotation guidelines, **labels are collected at a rate approximately proportional to the amount of human effort you throw at the problem**—in other words, your millionth label costs about as much as your hundredth. And on top of that, any non-trivial change to your input data or output objectives (e.g., a change in annotation instructions, data distribution, model requirements, etc.) generally requires a complete relabeling of your training dataset from scratch, at the same cost as the original effort ⁹.

Additionally, privacy and subject matter expertise requirements often limit who is able to contribute. While some manual labeling tasks are simple enough that they can be outsourced to crowdworkers, most enterprise-relevant labeling tasks require some level of domain knowledge. On top of that, most organizations are hesitant to expose internal data to external eyes. As a result, the number of people who are qualified to label data for a critical task can often be counted on one hand, exacerbating the bottleneck even further.

A great labeling UI can alleviate some of the pain. Easy project setup, keyboard shortcuts, auto-advancing, point-and-click options (see our annotation workspace in [Snorkel Flow](#))—**these little things can add up to make the labeling process marginally more efficient... but they won't make it 10x more efficient.** A human with domain expertise (and permission to view the data) still needs to view each and every example, so the fundamental bottleneck remains. And the labeling process remains opaque and difficult to reproduce.



With manual labeling, a human provides labels for examples one-by-one, presenting a scalability challenge.

Model-assisted labeling

Model-assisted labeling (also known as “pre-labeling”) is just what it sounds like: using a current model (or an existing model from some other task) to propose an initial label for your data points that a human can then approve or reject manually.

This can be seen as another type of improved UI for manual labeling. For some tasks—particularly those where collecting an individual label takes a long time or a lot of button clicks (e.g., image segmentation)—this can save the time that it would have taken to make those clicks yourself. **But the human is still viewing and verifying each example that receives a label, creating the labels one-by-one without associated functions for adaptability or governance, making it another marginal improvement of manual labeling** ¹⁹.

Importantly, even with a human confirming each label, there is risk in using your model to suggest labeling decisions ahead of time: it has the potential to bias labelers into reinforcing the status quo—existing error modes and all. This phenomenon was demonstrated in an MIT study ²² published at CHI 2021, the most heavily cited ²¹ annual conference for human-computer interaction. The study found that:

“...when presented with fully pre-populated suggestions, these expert users exhibit less agency: accepting improper mentions, and taking less initiative in creating additional annotations.”

In other words, this manual labeling add-on can actually end up doing more harm than good, lowering model quality in exchange for relatively minor efficiency improvements.



Model-assisted labeling can reduce the time to collect each individual label by suggesting a label for the user to confirm or modify, but they're still collected manually and one at a time.

Active learning

Active learning is a classic machine learning technique used and studied for decades ¹⁰. It aims to make human labelers more efficient by suggesting what order examples should be labeled in, so fewer examples can be labeled overall. For example, if constraints on time or money limit how many examples you can afford to label (and frankly, few projects have unlimited labeling budget), then you may prefer to label data points that are relatively dissimilar from one another (and therefore more likely to contribute novel information), or that your model is least confident about (and therefore likely to affect its current decision boundary).

In practice, this reordering of data points can boost manual labeling efficiency for some applications, particularly near the beginning of a project when a model has learned very little. As the model learns more, however, the problem of determining the most valuable points to label can be nearly as hard as the original problem, making gains over random sampling fairly marginal. Intuitively speaking: if you knew exactly where the decision boundary between classes was, you wouldn't need to train a classifier to do exactly that.

Finally, because the traditional active learning approach is still operating on the level of individual labels collected one at a time without associated labeling rationale, it has the same issues around indirect knowledge transfer, adaptability, and governance as the other members of the manual labeling family. **In general, active learning is a classic, very standard, and very sensible way to make manual labeling more efficient—but it's not a step change.**

Active Learning (Labeling)



With active learning, an algorithm attempts to select the data point that will be most useful to label next (still one-by-one).

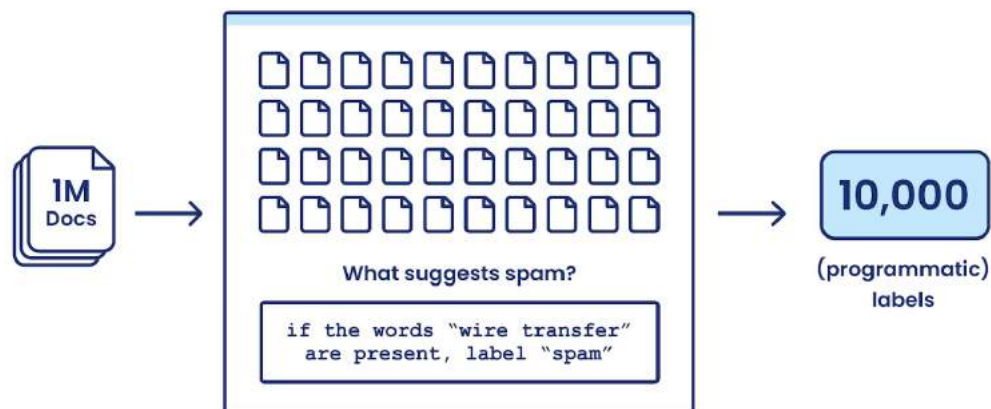
Programmatic labeling—a robust and practical way to automate data labeling

With the programmatic labeling approach to machine learning (also known as “data programming” or the “Snorkel” approach), the primary input from humans is not individual labels, but labeling functions (LFs) that capture labeling rationale. These functions are then applied to massive amounts of unlabeled data and aggregated to auto-label large training sets.

For example, if you’re building a spam classifier, you may have a number of relevant existing resources available (e.g., dictionaries of spammy words, a spam classifier from another domain, maybe even existing crowdworker labels). You may also be able to express a number of heuristics or “rules of thumb” that you or your domain experts would use if you were labeling by hand (e.g., formal but generic greetings like “dear sir or madam”, a high typo rate, links to wire transfer services, etc.).

You can express all of these resources and heuristics as labeling functions, which are then applied at scale to vast amounts of unlabeled data programmatically. **Importantly, this means that once you write a labeling function, no additional human effort is required to label your data, be it thousands or millions of data points.** And the same is true for when it comes time to update your model—for example, you can add a new labeling function addressing an error mode you’ve observed in your model, and then regenerate the entire training set programmatically, without having to review each data point individually.

Programmatic Labeling

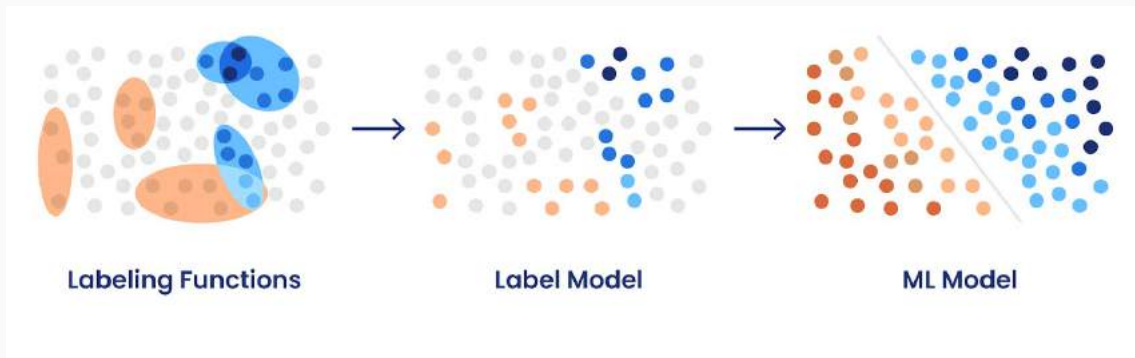


With programmatic labeling, the user conveys domain knowledge via labeling functions instead of individual labels, which are then used to label thousands of data points in seconds, creating large training sets that can be easily regenerated or updated as needed, presenting a robust and practical way to automate data labeling.

The labeling functions you write may vary in their coverages, accuracies, and correlations with one another, and they will almost certainly disagree with each other some portion of the time. But the Snorkel team and other researchers have developed many algorithmic techniques over the years for automatically learning how much to weigh these different sources of supervision during that aggregation process based on their learned accuracies and correlations. Many early versions (approx. 2016 – 2019) of these aggregation strategies have been published in peer-reviewed papers [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#), and all of them—plus many others—are available today in [Snorkel Flow](#).

The programmatic labeling approach to machine learning has four important properties:

1. **The labeling process is now truly automated**, providing orders of magnitude of efficiency improvements. This is in stark contrast to manual labeling with minor efficiency improvements from automation around the edges.
2. Even though the labeling process is now automatic, **the labeling decisions are still very much driven by the human**, who uses their domain expertise to decide what resources or heuristics to apply to the problem.
3. **The label creation process is now governable**—for every training label that your model is learning from, you can identify exactly what contributed to that label. If a bias is detected in your dataset, you can trace the source of that bias back directly and remove it.
4. **The efficiency gains of using programmatic labeling increase over the lifecycle of the model**. As inevitable changes in the problem definition or data distribution occur, adapting is easy because relabeling is now as easy as updating or adding labeling functions, rather than relabeling everything again by hand. And when more unlabeled data is acquired, it can be labeled and added to the training set without any additional manual effort.



Each labeling function suggests training labels for multiple unlabeled data points, based on human-provided subject matter expertise. A label model (Snorkel Flow includes multiple variants optimized for different problem types) aggregates those weak labels into one training label per data point to create a training set. The ML model is trained on that training set and learns to generalize beyond just those data points that were labeled by labeling functions.

And the kicker: programmatic labeling is fully compatible with the manual labeling workflow optimizations described above. Active learning can be used to suggest one or more points to label next. Existing models can show what the current predictions would be for those points ⁸. And the user can then write a labeling function based on those suggestions, updating the labels for many more data points at a time than in a manual workflow.

For more information on programmatic labeling, see this informative article on [How Snorkel Works](#), this amusing one on the [origins of the Snorkel project](#), and the [Snorkel AI website](#) for links to case studies and peer-reviewed papers of applying this approach to real-world problems.

Automate data labeling with Snorkel Flow

Snorkel Flow is the data-centric platform for building AI applications. It was built to maximize the end-to-end power of programmatic labeling. In Snorkel Flow, modern automated data labeling is made accessible, guided, and performant as users:

- Explore their data at varying granularities (e.g., individually or as search results, embedding clusters, etc.)
- Write no-code Labeling Functions (LFs) using templates in a GUI or custom code LFs in an integrated notebook environment
- Auto-generate LFs based on small labeled data samples
- Use programmatic active learning to write new LFs for unlabeled or low-confidence data point clusters
- Receive prescriptive feedback and recommendations to improve existing LFs
- Execute LFs at massive scale over unlabeled data to auto-generate weak labels
- Auto-apply best-in-class label aggregation strategies intelligently selected from a suite of available algorithms based on your dataset's properties
- Train out-of-the-box industry standard models over the resulting training sets with one click in platform, or incorporate custom models via Python SDK
- Perform AutoML searches over hyperparameters and advanced training options
- Engage in guided and iterative error analysis across both model and data to improve model performance
- Deploy final models as part of larger applications using your chosen production serving infrastructure
- Monitor model performance overall and on specific dataset slices of interest
- Adapt easily to new requirements or data distribution shifts by adjusting labeling functions and regenerating a training set in minutes.

While no hand-labeled data is required in Snorkel Flow, many teams find that labeling a small subset of ground truth data is helpful for evaluation or error analysis. **Snorkel Flow provides a workspace for hand-labeling to this end that includes the manual labeling workflow optimization tools described above**—keyboard shortcuts, active learning, and model-assisted labeling—as well as other features such as project dashboards, annotator roles, reviewer workflows, user comments, etc.

Finally, Snorkel Flow supports additional state-of-the-art machine learning techniques that aren't mentioned in this post, but are fully compatible with programmatic labeling. For example, users can:

- Apply **semi-supervised learning** with labeling functions and label aggregation strategies that factor in both labeled and unlabeled data
- Apply **transfer learning** by fine-tuning rich, pre-trained foundation models from industry standard model zoos made available in the platform
- Apply **few-shot** and **zero-shot learning** to create labeling functions from rich contextual models that do not have sufficient quality out of the box, but do have valuable signal

Developing AI solutions should be a partnership between human and machine—humans supply the intelligence and machines provide automation to scalably capture and encode the intelligence in a model. With programmatic labeling and other techniques made available in Snorkel Flow, we've made this ideal partnership a reality.

To learn more about how automated data labeling can take your AI development to the next level, request a [demo](#) or visit [our platform](#). We encourage you to subscribe to receive updates or follow us on [Twitter](#), [Linkedin](#), or [Youtube](#).

-
1. "Data preparation and [data] engineering tasks represent over 80% of the time consumed in most AI and Machine Learning projects." Data Preparation & Labeling for AI 2020, Cognilytica.
 2. This topic is particularly of interest to those who subscribe to a data-centric view of AI development. <https://github.com/HazyResearch/data-centric-ai>.
 3. "June 2019 Workshop". 2022. Snorkel.Org. <https://www.snorkel.org/blog/june-workshop>.
 4. Snorkel AI collaborates with Google, Intel, Stanford Medicine, DARPA, and customers such as Chubb, BNY Mellon, several Fortune 500 organizations, and government agencies
 5. "Technology". 2022. Snorkel AI. <https://snorkel.ai/technology/#research>.
 6. "Twenty Questions – Wikipedia". 2022. En.Wikipedia.Org. https://en.wikipedia.org/wiki/Twenty_questions.
 7. "Case Studies". 2022. Snorkel AI. <https://snorkel.ai/case-studies/>.
 8. Gershgorn, Dave. 2017. "The Data That Transformed AI Research—And Possibly The World". Quartz. <https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/>.
 9. Bach, Stephen H., Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, and Souvik Sen et al. 2018. "Snorkel Drybell: A Case Study In Deploying Weak Supervision At Industrial Scale". Arxiv.Org. <https://arxiv.org/abs/1812.00417>.
 10. Burr Settles' technical report first published in 1995 and updated in 2009 is one of the most cited surveys of the technique. 2022. Minds.Wisconsin.Edu. <https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf>

11. Ratner, Alexander J., Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. "Data Programming: Creating Large Training Sets, Quickly". *Advances In Neural Information Processing Systems* 29. <https://proceedings.neurips.cc/paper/2016/hash/6709e8d64a5f47269ed5cea9f625f7ab-Abstract.html>.
12. Ratner, Alexander, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. "Snorkel". *Proceedings Of The VLDB Endowment* 11 (3): 269-282. doi:10.14778/3157794.3157797.
13. Stephen H. Bach, Christopher Ré. 2017. "Learning The Structure Of Generative Models Without Labeled Data". *Proceedings Of Machine Learning Research* 70: 273. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6417840/>.
14. Ratner, A., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., & Ré, C. (2019). Training Complex Models with Multi-Task Weak Supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 4763-4771. <https://doi.org/10.1609/aaai.v33i01.33014763>.
15. Varma, Paroma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. 2019. "Learning Dependency Structures For Weak Supervision Models". *Arxiv.Org*. <https://arxiv.org/abs/1903.05844>.
16. Varma, Paroma, Frederic Sala, Shiori Sagawa, Jason Fries, Daniel Fu, Saelig Khattar, and Ashwini Ramamoorthy et al. 2019. "Multi-Resolution Weak Supervision For Sequential Data". *Advances In Neural Information Processing Systems* 32. <https://proceedings.neurips.cc/paper/2019/hash/93db85ed909c13838ff95ccfa94cebd9-Abstract.html>.
17. Fu, Daniel Y., Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. "Fast And Three-Rious: Speeding Up Weak Supervision With Triplet Methods". *Arxiv.Org*. <https://arxiv.org/abs/2002.11955>.
18. Chen, Mayee F., Daniel Y. Fu, Frederic Sala, Sen Wu, Ravi Teja Mullapudi, Fait Poms, Kayvon Fatahalian, and Christopher Ré. 2020. "Train And You'll Miss It: Interactive Model Iteration With Weak Supervision And Pre-Trained Embeddings". *Arxiv.Org*. <https://arxiv.org/abs/2006.15168>.
19. Some techniques in the semi-supervised learning family (see <https://link.springer.com/article/10.1007/s10994-019-05855-6> for a nice summary) take model-assisted labeling one step further and actually do apply the label without human review. This is made feasible by making certain assumptions about the distribution of the data and/or limiting the amount of "automatic" labeling that is allowed to occur in an attempt to limit semantic drift. However, these techniques also suffer from a lack of governability and the inability to transfer domain knowledge in a more direct form to the model than individual labels.
20. "Perpetual Motion – Wikipedia". 2010. *En.Wikipedia.Org*. https://en.wikipedia.org/wiki/Perpetual_motion.
21. Human computer interaction – google scholar metrics. (n.d.). Retrieved February 4, 2022, from https://scholar.google.es/citations?view_op=top_venues&hl=en&vq=eng_humancomputerinteraction.
22. "Assessing The Impact Of Automated Suggestions On Decision Making: Domain Experts Mediate Model Errors But Take Less Initiative | Proceedings Of The 2021 CHI Conference On Human Factors In Computing Systems". 2022. *DI.Acm.Org*. <https://dl.acm.org/doi/10.1145/3411764.3445522>.

Accelerate AI with Snorkel Flow

Adopting Snorkel's data-centric approach can be instrumental to not only automating the labeling process (while keeping human-in-the-loop) but also to accelerating AI development.

With Snorkel Flow, Fortune 500 organizations such as Chubb, BNY Mellon, and more have built accurate and adaptable AI applications fast by putting the power of programmatic labeling to use.

Beyond realizing Snorkel's data-centric approach, Snorkel Flow has many other features that elevate it from a training set or model creation tool to a complete enterprise platform for building AI applications. Some of those features include:

- **Application studio:** Support for multiple data modalities (unstructured text, structured text, time series, etc.), problem types (NER, info extraction, multi-label classification, etc.), and custom application graphs combining multiple models, pre-/post-processors, and custom business logic
- **Programmatic labeling IDE:** no-code LF templates, auto-suggest and auto-tuning of LFs, interactive data visualization, automated management and versioning of LFs, etc.
- **Training data management:** advanced label model algorithms, automated parallelization, and hyperparameter selection, dataset versioning and comparisons, etc.
- **Model training and serving:** one-click model training or fine-tuning, hyperparameter search, endpoint creation for model/application serving, export for serving at scale, etc.
- **Deployment and security:** REST API, monitoring services, and managed workers for job execution; encryption, authentication, and role-based access control (RBAC), managed SSO integration; support for hosted, hybrid, or on-premises deployments, etc.

To see Snorkel Flow in action: [Request a demo](#).

To stay in touch, follow us on [Twitter](#), [LinkedIn](#), [Youtube](#).