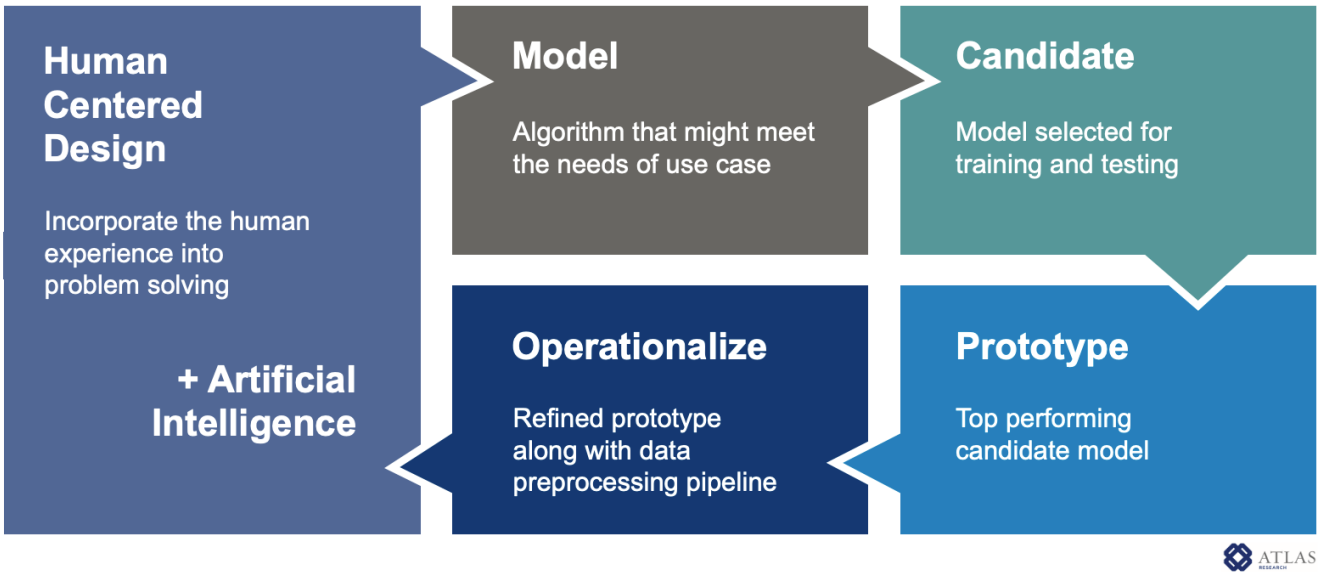
Comprehensive Guide to Model Selection

Nicole Janeway Bills

Here's a systematic approach to picking the right algorithm.



To help navigate the abundant options for creating a machine learning model, we recommend a five step process that results in a useful data product.

We recommend starting off with **Human Centered Design**-empowered approach. HCD focuses on the challenges faced by the end user and uses a framework to make decisions up-front that will guide the remainder of the model selection process. This helps keep the data scientist working toward resolving the business problem — not getting mired in technical difficulties.

With the learnings from the HCD phase in mind, the data scientist next conducts a survey of the landscape of **models** that could tackle the

business challenge.

Based on understanding of the needs of the end user and research into available models, the data scientist narrows the search to a subset of **candidates**. The data scientist then trains and tests these models, iterating over hyperparameters.

The top performing 1–2 candidates become **prototypes**. Their performance is judged based on a pre-agreed rubric of qualitative and quantitative considerations.

Finally, the team **operationalizes** the selected prototype. Continued maintenance is required over time to ensure model security, prevent drift, and address potential sources of bias.

Following these steps will ensure you end up with an effective result.

Background

In the recent past, data science involved choosing between methodologies based in classical statistics, such as regression and decision trees. Alone or combined into an ensemble, these models are capable of producing quite sophisticated results that approximate real world phenomenon. The use of these techniques remains widespread across academic and enterprise settings.

Today's data scientist has the opportunity to build upon these tried-and-true methods through machine learning. This refers to the application of complex algorithms, often called neural networks, that improve their ability to model the real world through experience of a training dataset.

Machine learning projects take many forms:

- Exploring an **unlabeled dataset through unsupervised learning** to produce categories or clusters

- Using **labeled data for supervised learning** to produce classifications or predictions
- Undertaking a **blended combination** of the above through **semi-structured learning**
- Informing a **series of decisions with reinforcement learning**

Common use cases for machine learning include:

- **Natural language processing (NLP)** — text analytics, including unsupervised topic modeling and natural language generation as well as supervised text classification and text regression
- **Computer vision** — including image recognition, detection, and classification

Across these categories and use cases of machine learning, the imperative to take a systematic approach to model selection remains.

If anything, additional structured research and decision-making processes are required when looking to implement a machine learning solution. This field has grown remarkably since the early 2010s. Many of the ground-breaking techniques are available to all data science practitioners through open source distribution.

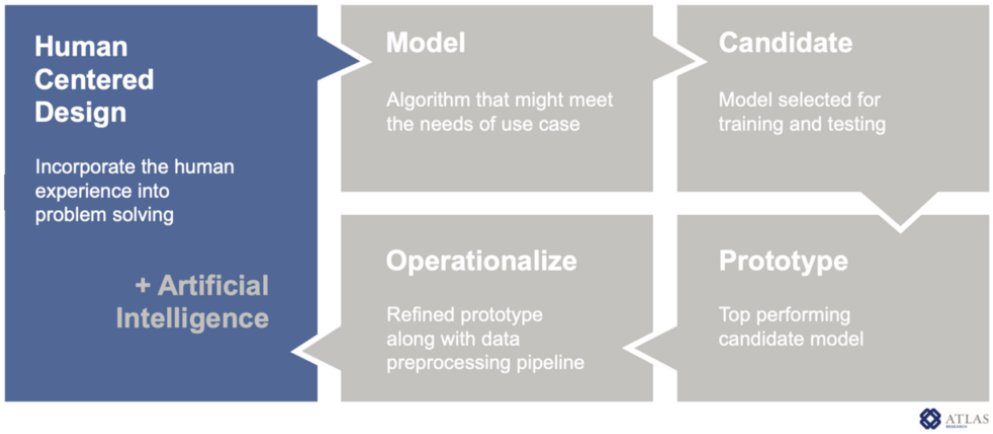
This advantage of machine learning also gives rise to many of the challenges that this article aims to resolve:

1. Keeping in mind the needs of the end user as discovered through an **HCD**-empowered exploration, the data scientist avoids losing valuable time to a fruitless path of exploration
2. Conducting thorough research of the full landscape of available **models**, the data scientist ensures discovery of many promising options for further investigation
3. Selecting several **candidates**, the data scientist explores a variety of capabilities

4. Evaluating **prototypes** using a predefined framework, the data scientist arrives at the top performing model
5. **Operationalizing** the model based on best practices of DevSecOps, the data scientist ensures the organization is set up to succeed with a pipeline that is robust to potential failure points

These five steps comprise the model selection roadmap outlined in this article.

Human Centered Design



Human Centered Design (HCD) is a flexible method of incorporating the human experience into problem solving. The goal is to learn directly from the people experiencing a given issue, then iteratively test and design a solution. Frequent communication ensures the solution 1) actually offers resolution of the end user's challenge, 2) is technologically feasible for the data scientist to create, and 3) is viable for the organization to adopt.

Here is non-exhaustive a list of questions that should be asked in these conversations:

- Is the challenge that you're trying to solve a use case for data science? Is it **ad hoc or reoccurring**?
- Are you looking to **inform or automate** decision-making?
- How will the end user **trust** the results of the model?

- What is the **level of expertise** of the team that will be deploying and supporting this solution in the future?

Using human-centric design principles from the outset of the model development process is critical to ensure adoption and mitigate risks associated with the deployment of a machine learning based system. Without HCD, artificial intelligence runs the risk of pushing consumers further to the extreme of their information bubble, encoding societal biases, and deepening the status quo.

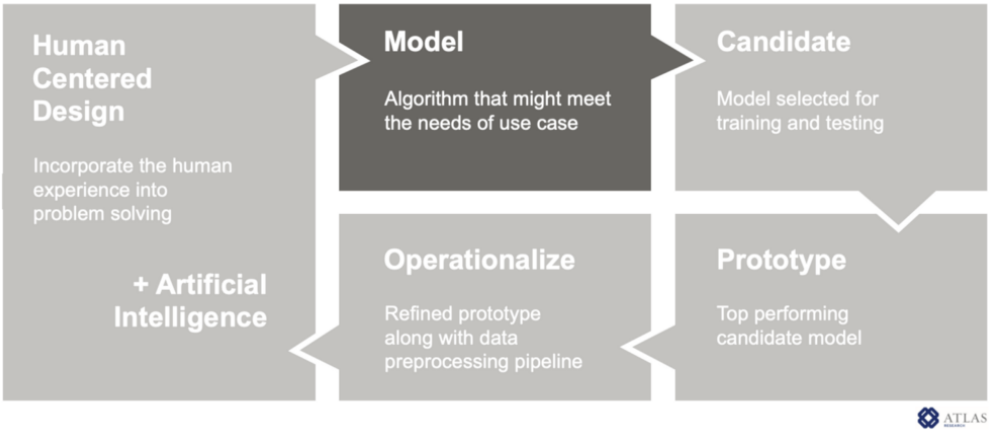
Here are three examples of those risks:

1. **YouTube's recommendation engine** quickly drives viewers toward increasingly extreme content. According to sociologist : "Videos about vegetarianism led to videos about veganism. Videos about jogging led to videos about running ultramarathons. It seems as if you are never 'hard core' enough for YouTube's recommendation algorithm." A similar effect can be seen across social media and content delivery networks. This pattern of segmentation is beneficial to the algorithms underlying this technology as it leads to wider decision boundaries between classes — but it is detrimental to society, increasing the struggle to find common ground. *Read more via* .
2. In November 2019, **Apple Card** was investigated by NY Department of Financial Services for gender bias in its algorithms. Female applicants were offered credit limits 20x smaller than those of their male counterparts. *Read more via* .
3. Courts across the U.S. have deployed algorithms to **predict recidivism**. This application of machine learning risks perpetuating existing inequalities into a well-trodden feedback loop. While a defendant's race might be explicitly left out of the feature set, features highly correlated to race must also be eliminated in order to reduce disparities in the judgement of AI-based systems. *Read more*

via .

Organizations can identify and eliminate these sources of potential AI-malpractice through Human Centered Design. Bottom-line: a machine learning model does not possess understanding. HCD enables data-driven organizations to take into account the environment into which their solution will be placed — an environment the model will reflect in how it forms its output. The HCD process links AI to business objectives, putting human goals at the center of development work. *Read more*.

Model



In our roadmap, the term **model** is used to refer to any algorithm that might meet the needs of the organization's use case. If the model selection process is looking for a needle in a haystack, then you could think of the term "model" as referring to any haystack that you could be searching.

As implied by that metaphor, the data scientist has abundant options when it comes to model section. Many state-of-the-art models, capable of attaining superhuman performance on a specific task, are freely available through open source distribution.

For NLP, there are currently 2,276 models in the Transformers library for text analytics. These models build upon state-of-the art breakthroughs

like Google's BERT. There are at least +150 popular models for image classification, segmentation, and detection.

The torrent of preprints enabled by arXiv has contributed to a culture of innovation and rapid speed to insight within the field of machine learning research. Photo by Rostyslav Savchyn on Unsplash.

We recommend a four step process for the *model phase* of model selection:

1. **Evaluate** whether the model solves the problem at hand
2. Assess **popularity / "ground-breaking-ness"**
3. Review potential **issues**
4. Analyze **documentation and preprints**

We'll illustrate this process with a quick example. On a recent project, our team at Atlas Research was tasked with developing a tool for named entity recognition (NER), a subtask within the field of natural language processing.

NER requires the machine learning model to pick out relevant snippets (i.e. entities) from a larger body of text. As you can imagine, there are a number of approaches to tackle this challenge. We used these questions to guide our research in the *model phase*.

Step #1 — Does the model address the use case?

We applied learnings from the HCD phase to ensure the needs of the end user were kept top of mind. Also, it was important for us to think through the capabilities and bandwidth of the team that would ultimately be responsible for the deployment and maintenance of the model. This made sure we didn't extend beyond the level of complexity that our client could support over the long term.

At this early stage of research, we needed to make sure our NER model was capable of identifying relevant entities. Our text corpus was from the medical domain, so we needed to look for models capable of parsing text containing clinical jargon.

Step #2 — How does this model fair in terms of its popularity? How about its "ground-breaking-ness"?

Popularity can be assessed by number of GitHub stars or Hugging Face "number of downloads in the past 30 days" metric.

Popularity can be a good indicator that the model will work well for a similar use case. It's worth keeping in mind that machine learning solutions are typically very narrow in scope. We made sure to evaluate the popularity of a model by looking across comparators in the biomedical domain.

Step #3 — What potential issues are associated with this model?

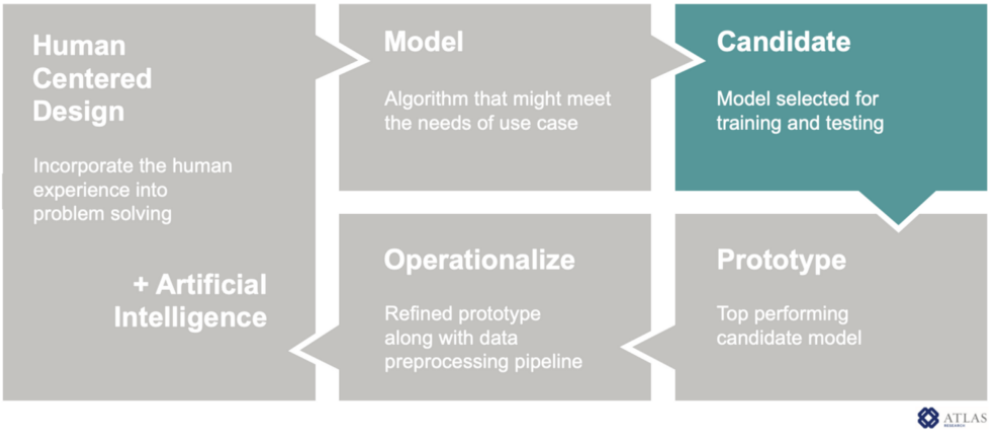
The obvious place to start is the model's GitHub Issues page. High volume can indicate a backlog, lack of developer support, and/or buggy software.

By reviewing GitHub issues, we found that one potential model had lost the support of its developer team. A good developer team is responsive to addressing issues quickly. We thought that this lack of support could represent a major barrier when handing off the model to our client counterparts, which we flagged as a risk.

Step #4 — What can we learn from the model documentation? What about papers on arXiv?

We advanced our understanding of the models we were investigating by reading through documentation and related arXiv papers. Also, we found this helpful package that allowed us to easily test a variety of candidates without changing our coding interface too much.

Candidate



At this phase of the model selection roadmap, you'd want to train, validate, and test multiple algorithms ranging in complexity. These are your **candidates**.

The **No Free Lunch theorem** of data science (Wolpert 1996; Wolpert and Macready 1997) tells us that:

> *Learning algorithms cannot be universally good.*

There does not exist a singular best model that solves a variety of challenges. Instead, choosing the right model for the job requires a harmonious fit between the **task, dataset, and constraints** such as team expertise and available compute resources.

Given the No Free Lunch theorem, it's useful to pick candidate models that range in level of complexity (also referred to as **capacity**).

Choosing a **linear regression model** for text classification would be like bringing a knife to a gunfight. On the other hand, **Google's BERT model**, with 110 million trainable parameters, might be the equivalent of bringing a water cannon to the Super Soaker themed birthday party of your baby cousin.

Fine-tuning BERT (i.e., using the model for transfer learning) requires significant time and compute resources. And the resulting **"BERT boost"** over the accuracy of less computationally intensive models may be just **1–3 points of accuracy**.

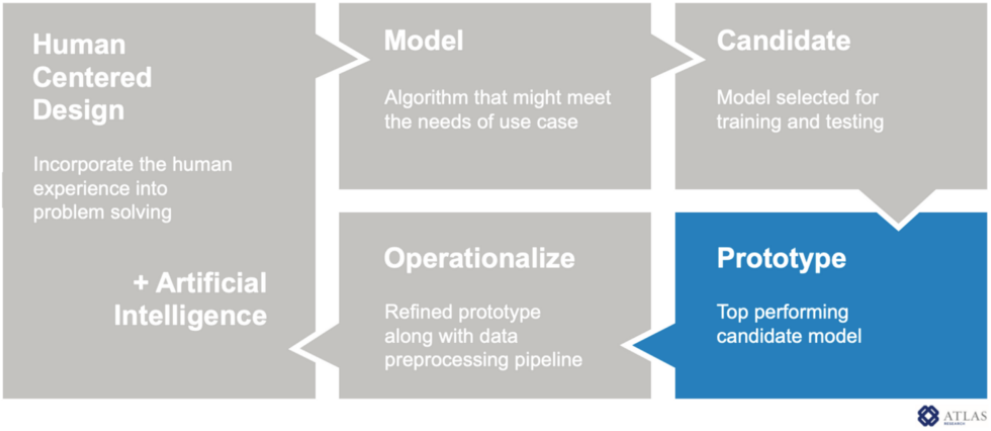
BiLSTM with CRF could represent a good middle ground between these options. We might call it the "Goldilocks model" — it's complex enough that it allows us to take advantage of the ability of deep learning to capture nuanced relationships in the underlying data, but not so complex that the model will overfit.

The goal of the *candidate phase* is to take the intuition displayed in the example above and apply it to **select models at a range of capabilities:**

- **Classic statistical models** — e.g., a boosted random forest or SVM
- **Foundational neural nets** — e.g., BiLSTM with CRF
- **Current state-of-the-art** — e.g., BERT, DistillBERT, GPT-2, etc. (if your environment supports it)

Next, you'll train these various models and compare results.

Prototype



In our roadmap, **prototype** refers to the top performers among the candidates.

Attaining a set of top performers is possible only after extensive hyperparameter tuning. Brute force grid search is your enemy — instead, there are heuristics that you can apply to narrow the search space based on early training runs.

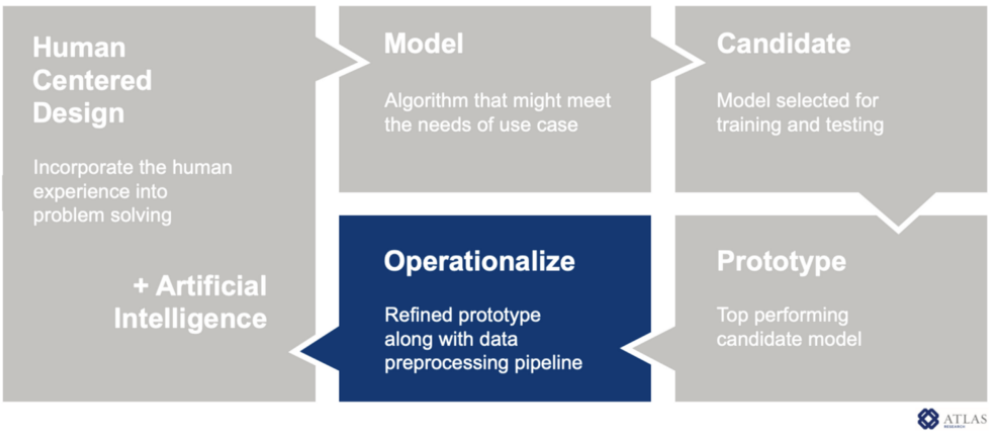
During the *HCD phase* of the roadmap, you should have worked with the stakeholder group to decide your rubric for evaluating models.

> *Increasingly designers are going to have to be concerned not just with human behavior and decision-making, but also fundamental needs and values. Read more.*

Candidate evaluation should take into account environmental constraints such as available compute resources and deployment expertise. Candidates should also be assessed against a slate of pre-selected quantitative metrics, such as accuracy, f-1 score, Matthew's Correlation Coefficient, etc.

After completing the *prototype phase* by tuning models, employing a custom framework for evaluation, and picking out the top performers, it's time to deploy the solution.

Operationalize



When putting the chosen model into production, we aim to create a data product — an application that derives its value from data, and in so doing, creates more useful data as a result.

Let's review some considerations for this phase:

- Seek opportunities to **streamline** — e.g., through pruning and quantization — before putting the model into production
- Outline **requirements** for successful deployment, which is typically conducted by a DevSecOps team
- Put into place consistent monitoring against the risk of **"data drift"** — i.e., when features are no longer measuring the real-world conditions that they were initially measuring when the model was developed
- Conduct a thorough investigation of potential **sources of bias** that may enter the dataset
- Assess robustness of the **pipeline** that enables consistent flow of high-quality data to the model
- Protect against **adversarial attacks** and **reverse engineering**
- Plan for the **computational requirements** of inference

Summary

In this article, we discussed a roadmap for model selection.

As a critical starting point, **Human Centered Design** enables stakeholders and technical teams to align on the functional requirements of the machine learning empowered tool. Through the HCD process, stakeholder voices are integrated into the design. This approach helps avoid pitfalls associated with mindlessly deploying an “artificially intelligent” solution.

During the **model** phase, the data scientist researches and evaluates the search space of possible solutions.

In the **candidate** phase, the data scientist narrows selection to the subset of models most apt for training and testing.

In the **prototype** phase, the top performing model or models are identified using rubric of pre-selected metrics.

In the **operationalization** phase, the selected prototype goes into production based on best practices of DevSecOps.