CHAPTER 4

A Collaborative Process

The relationship between humans and machines is evolving rapidly. Designing the parameters of this relationship—meaningfully and ethically—is complex, requiring the application of a human-centric approach that prioritizes our needs and goals.

However, this is not always the case. Often the process of designing and developing ML systems is determined by the technological limitations and is guided by what a model can provide. Designing learning systems in a human-centered way requires careful consideration of the ultimate learning goal, feedback mechanisms, and strong collaboration between designers and AI practitioners. Currently, that is still missing—or is not as strong as it should be.

In my experience, the researching, envisioning, and designing of learning systems should not overlook effective collaboration between designers and AI practitioners. Such a collaboration can make a difference in the definition of goals by working forward from the needs of users.

In this chapter, I cover questions such as "How can designers and AI practitioners collaborate?" and "How might we start with a human end goal in mind?"

Designing backwards

When it comes to AI, the focus often leans more toward the algorithm and model, with the learning outcome for the machine defined by technical opportunities and constraints. In such cases, the end user is penalized because what reaches the market isn't the result of a well-defined plan pursuing an end-learning goal that enriches the product or service usage, but rather simply mar-

keting what was feasible in the short-term.

To build models that really fulfill human needs, the starting point must be the human end goal. Defining this is necessary to later define what the model needs to learn and the mechanisms that will lead to that learning goal.

Reimagining ML systems as subjects of learning, rather than static entities, can shift the mindset regarding the approach needed to build a system. As developed by Grant Wiggins, Backward Design is an educational framework that emphasizes the importance of starting with clearly defined learning outcomes—and only then designing assessments, experiences, and feedback mechanisms that will support the attainment of these outcomes.

As in human education, it is only when the end learning outcome is clear that we can build effective mechanisms for establishing user interactions and measuring learning progress. Instead of designing a product or service with a fixed output, ML should be considered an evolving entity that learns, over time, to reach a learning goal.

The Backward Design framework involves three main steps:[15]

1. Identifying the desired results.

2. Determining acceptable evidence.
3. Planning learning experiences and instructions.

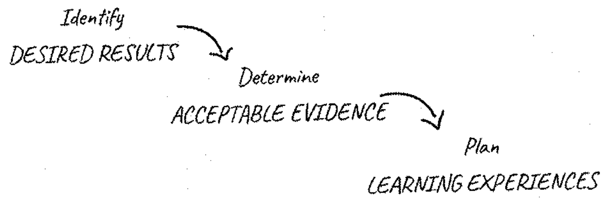

Figure 4.1: Stages of Backward Design

Framing an end goal is important; it should determine how the entire system is built. Consider a music streaming service such as Spotify. While using the application, a user's end goal is just to listen to music, while the desired learning outcome for the machine could be framed as finding old favorites and new songs to recommend to the user. Indeed, what Spotify does is provide users with a stream of music based on previously listened-to songs, along with songs that similar users have listened to.

When the learning outcome is clearly defined, based on the needs of users, then we can identify

the acceptable evidence that the goal is achieved.

Acceptance criteria shouldn't just include what we can measure about the model, but also gauging how well the chosen model is performing in helping humans reach their end goal. This is the reason behind the learning goal. Spotify tracks the time spent on the platform, how many recommended songs were listened to, and how many were skipped.

The third step of defining the experiences through which the learning happens involves creating mechanisms to enable it. For example, interactive training interfaces that allow users to provide explicit feedback on the system's predictions, use usage as an implicit way to improve the model, or use other technical methods to improve the model's performance.

The Backward Design approach ensures that we are framing the right learning goal at the beginning, measuring the performance correctly, and enabling the best-optimized mechanisms for that goal.

This three-step process requires strong collaboration between designers and AI experts, the kind that is only possible by combining the design process and engineering processes carried out by data scientists and ML engineers.

Two misaligned processes

Working backward from the desired output means working backward from user needs. Even if there is a growing recognition of the importance of designing AI systems with the user in mind, sometimes the collaboration between designers and AI experts is not very strong—or at least not well-formed.

The role of designers in the field is not well-defined either, so expectations and contributions from them are highly varied and sometimes restricted to what they routinely do on other projects. However, there is significant value designers can provide toward more human-centric ML-based product and service design. Enabling this comes down to aligning the ML and design processes, which until now have been maintained as separate and misaligned tasks and activities.

A typical ML process includes data collection, data cleaning, features engineering, model selection, model training, model evaluation, and deployment.

The data collection step is about gathering the relevant data for the problem. This may involve scraping from the web, data collection from sen-

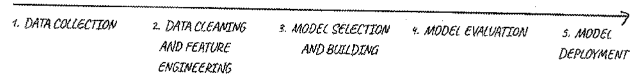

Figure 4.2: The ML Process

sors, or mining databases.

Once collected, it is time for data preprocessing, which involves cleaning and transforming the "raw" data into a dataset suitable for use in an ML model. This includes removing missing values, scaling, and transforming the material into a format that can be used by the ML algorithm.

Feature engineering involves selecting and creating features that are relevant to the problem being solved, either by combining or transforming existing features or creating new ones.

Model selection is choosing an appropriate ML model suitable to the problem being solved. This may involve choosing between classification and regression models or selecting from various types of neural networks.

At this stage, ML engineers and scientists can define the model parameters for the training, then proceed to training the model.

The final step is evaluating the model according to chosen metrics to ensure it is accurate and reliable before deployment. Then, the model is ready for implementation in the real world.

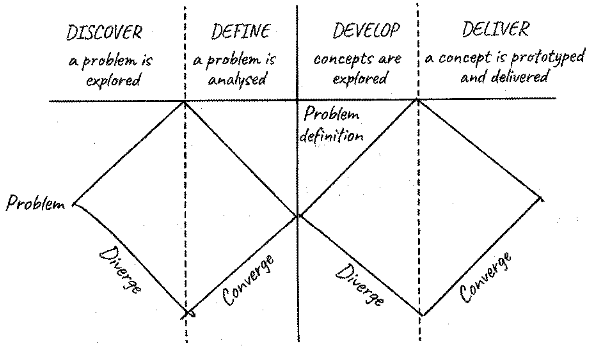

Figure 4.3: The Double Diamond Model

Looking at this process, it seems that it works somewhat opposite to the design process, which starts from real scenarios and the needs of potential users and then determines the problem and a definition for the direction of the product and service.

Typically, the design process is represented through the double diamond, a framework for problem-solving widely used in the design industry. It was developed by the UK Design Council and consists of four phases: discover, define, develop, deliver.

The discovery phase is all about understanding the problem—and the people affected by it. This involves research, analysis, and empathy building

in order to identify user needs, pain points, and goals.

The second part of the first diamond is the define phase, in which the research is synthesized by creating user personas or archetypes and prioritizing user needs.

Once the problem is defined, the development phase is about generating ideas and prototyping solutions. This involves ideation sessions, sketching, wireframing, rapid prototyping that can be tested by users, and iterating possible solutions. By testing solutions, designers can discover new scenarios that have not been covered and, in some cases, discover that the solution needs to consider other aspects not taken into consideration in the initial research. This enables designers to find design gaps or usability issues that otherwise would have been delivered to users.

The last step is, indeed, delivering a product to customers. This is when chosen and refined solutions are actually developed and put into the hands of users.

The Double Diamond process is iterative, meaning that designers move back and forth, as needed, between phases to refine their understanding of the problem and generate effective solutions. It is also human-centered, meaning that the needs and experiences of users are central to the process at all stages.

There are many frameworks for both design and ML processes—and many visual representations of such processes—but the main point is that whatever design or ML framework is taken into consideration, they are not integrated, they start from a different perspective and have different activities.

To enable a human-centered way of designing and developing these systems, a shared process is needed.

A shared process

A collaborative approach is needed—among designers, data scientists, and ML engineers—to ensure that the human experience is at the forefront of design decisions. This is done by working backward from human needs.

A collaborative process would see designers involved before, and during, data collection to best conduct user research and identify user needs.

By sharing the insights coming from the research, designers would inform the definition of the learning goal. This is the point from which to

work backward and define the data that is needed to train the model and key features.

Data collection is usually considered a data scientist activity; however, designers play a crucial role in enabling data collection by defining the best methods and tools to collect it and by defining the best-suited interaction patterns.* Interactions greatly impact the way we collect data—and which data we can collect—and this is a theme covered in the next two chapters.

ML practitioners are usually responsible for defining the type of AI to be used. But have you ever wondered about the impact of using one AI technique over another? The technique chosen has a huge impact on the outcomes and experiences provided. Different AI techniques have unique strengths, limitations, and approaches to problem-solving.†

Even within the subfield of ML, techniques are designed to tackle specific types of prob-

---

*According to the Interaction Design Foundation, an interaction design pattern is a generally repeatable solution to a commonly-occurring usability problem in interface design or interaction design.

†For example, deep learning neural networks excel at making predictions from vast amounts of data, while rule-based systems are effective for capturing explicit knowledge and making logical inferences.

lems. Convolutional neural networks (CNNs) are a popular technique for image classification tasks and are designed to handle spatial data, such as images. They are particularly effective at capturing local patterns and hierarchical representations. Other techniques, such as support vector machines (SVMs), are powerful tools used for classification and regression problems.

By selecting the most appropriate model for a particular problem, we can maximize the chances of achieving accurate and meaningful results for users.

Testing—before any implementation with final users—about if and how the model tackles the problem provides great advantages. By collecting user feedback on the output, designers could test with prototypes to make better decisions regarding the interface provided through a specific model. Such prototypes can also help in understanding the usefulness, usability, and desirability of the model output. In doing so, measuring the model is not only done on technical metrics, such as accuracy, but also experience metrics like perceived transparency and trust.

Finally, by collaborating during deployment and maintenance, designers can work closely with data scientists and ML engineers to mon-

itor the system's performance to ensure that it remains fair, transparent, and accurate while maintaining coherence with human needs. How the machine should evolve via usage over time can also be envisioned. More on this in Chapter 6 on "Designing With Loops".

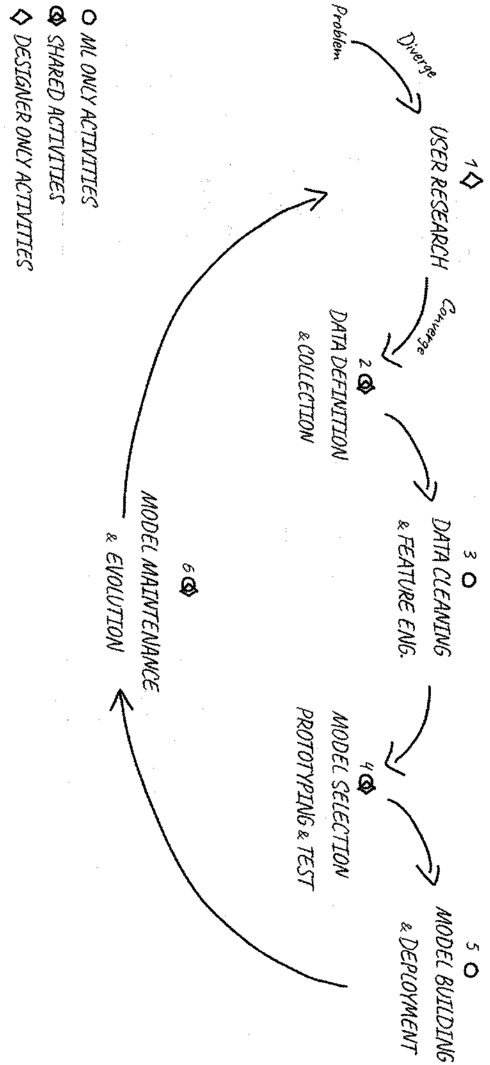

Figure 4.4: Shared Process between Design and ML Practitioners (continues on next page).

Figure 4.4:

Shows the following steps and activities:

1. In the divergence stage, UX works backward from the user needs.
2. In the convergence stage, UX and ML (practitioners) work together to define the ML learning goal designing backward from research results. UX facilitates the definition of the dataset according to the learning goal. UX defines the best data-collection tools and interaction types to build a representative dataset. ML builds the dataset.
3. ML cleans the dataset and works on feature engineering.
4. ML selects the model according to performance while UX helps define the appropriate model for the learning goal by prototyping, testing with users, and validating with human-centered metrics.
5. ML builds, evaluates, and deploys the model.
6. Together, ML and UX define how the model should evolve and track how new incoming data affects the model and user experience.

The first and the second steps of the process presented in Figure 4.4—user research and data definition and collection—are where the Backward Design framework should be embedded.

The activities coming from the Backward Design framework, are:

1. *Identifying the desired results for the machine according to human goals.* The main questions we want to answer at this stage are: "What is the human need we need to cover? What are the learning outcomes for the machine that will cover the human need?"

2. *Determining acceptable evidence that the outcome is reached.* The main questions we want to answer at this stage are: "What are the criteria to determine if we have satisfied the human need? What is the evidence that we have reached the machine's learning outcome?"

3. *Planning learning experiences and instructions for the machine to learn*, the learning events that enable the machine to collect new data and feedback. This requires answering the questions: "What data does the machine need to collect? What are the skills

needed to be acquired? How do we design for events enabling learning?"

Once the product/service is released, the team can collaboratively work on iterating the system by envisioning its evolution and measuring machine learning goal achievement not only via machine-centered metrics, but also via human-centered metrics.

The following three chapters will dive deeper into this process. Methods and tools to design backward from human needs—to answer the technical and ethical challenges ML and AI bring generally to the design of products and services—will be presented.