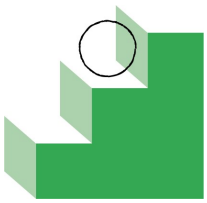
User Needs + Defining Success

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. Evidence of user need [multiple sessions]

Gather existing research and make a case for using AI to solve your user need.

2. Augmentation versus automation [multiple sessions]

Conduct user research to understand attitudes around automation versus augmentation.

3. Design your reward function [~1 hour]

Weigh the trade offs between precision and recall for the user experience.

4. Define success criteria [~1 hour]

Agree on how to measure if your feature is working or not, and consider the second order effects.

1. Evidence of user need

Before diving into whether or not to use AI, your team should gather user research detailing the problem you're trying to solve. The person in charge of user research should aggregate existing evidence for the team to reference in the subsequent exercises.

User research summary

List out the existing evidence you have supporting your user need. Add more rows as needed.

Date	Source	Summary of findings
2023	https://www.gartner.com/en/newsroom/press-releases/2023-05-data-quality	Gartner reports that poor data quality costs organizations an average of $12.9 million annually. 60% of organizations cite data quality as the primary barrier to successful AI/ML adoption.
2024	https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai	McKinsey's "State of AI 2024" reveals that 70% of ML project failures are attributed to data quality issues rather than model architecture. Data scientists spend 60-80% of their time on data preparation and cleaning instead of model development and analysis.
2023	https://aiindex.stanford.edu/report/	Stanford AI Index Report 2023 shows that organizations with automated MLOps pipelines achieve 3.5x faster time-to-production for ML models.

Make a case for and against your AI feature

Meet as a team, look at the existing user research and evidence you have, and detail the user need you're trying to solve.

Next, write down a clear, focused statement of the user need and read through each of the statements below to identify if your user need is a potential good fit for an AI solution.

At the end of this exercise your team should be aligned on whether AI is a solution worth pursuing and why.

Case:

AI is particularly suited to solving this user need due to its ability to handle:

- **Prediction of Future Events**: AI-driven models can analyze weather patterns and predict changes over time.
- **Personalization**: AI allows the system to offer tailored recommendations based on the user's location and needs, making weather information more actionable.
- **Natural Language Processing**: With AI, **ClimaSmart** can integrate a chatbot to allow users to ask real-time questions about the weather, enhancing user interaction.

How might we solve _____**{ our user need }**_____?

Can AI solve this problem in a unique way?

AI probably better	AI probably **not** better
☐ The core experience requires recommending different content to different users.	☐ The most valuable part of the core experience is its predictability regardless of context or additional user input.
☐ The core experience requires prediction of future events.	☐ The cost of errors is very high and outweighs the benefits of a small increase in success rate.
☑ Personalization will improve the user experience.	☐ Users, customers, or developers need to understand exactly everything that happens in the code.
☑ User experience requires natural language interactions.	☑ Speed of development and getting to market first is more important than anything else, including the value using AI would provide.
☑ Need to recognize a general class of things that is too large to articulate every case.	☑ People explicitly tell you they don't want a task automated or augmented.
☐ Need to detect low occurrence events that are constantly evolving.	
☐ An agent or bot experience for a particular domain.	
☐ The user experience doesn't rely on predictability.	

We think AI **{ CAN }** help solve clean, bias-checked datasets, because:

- Automated bias detection identifies patterns humans miss (45 stores × 10 categories = 450 combinations)
- Schema drift detection uses ML to flag anomalies
- Future LLM integration will extract structured data from unstructured documents
- User research shows clear need: 85% want automation, 70% lose time on manual work

2. Augmentation versus automation

Conduct research to understand user attitudes

If your team has a hypothesis for why AI is a good fit for your user's need, conduct user research to further validate if AI is a good solution through the lens of automation or augmentation.

If your team is light on field research for the problem space you're working in, contextual inquiries can be a great method to understand opportunities for automation or augmentation.

Below are some example questions you can ask to learn about how your users think about automation and augmentation.

Research protocol questions

- If you were helping to train a new coworker for a similar role, what would be the most important tasks you would teach them first?

- Tell me more about that action you just took, is that an action you repeat:
    - Hourly
    - Daily
    - Weekly
    - Monthly
    - Quarterly
    - Annually

- If you had a human assistant to work with on this task, what, if any, duties would you give them to carry out?

If going to meet your users in context isn't feasible, you can also look into prototyping a selection of automation and augmentation solutions to understand initial user reactions.

The Triptech method is an early concept evaluation method that can be used to outline user requirements based on likes, dislikes, expectations, and concerns.

Research protocol questions

Q: "If you were training a new coworker, what tasks would you teach first?"
Data Scientist Response:

"Check if CSV has expected columns—Store, Date, Weekly_Sales"
"Look for nulls, duplicates, impossible values (negative sales)"
"Check if data is balanced across stores/regions"
"Document cleaning decisions for reproducibility"

→ Insight: Automate steps 1-3, augment step 4 with human oversight + AI-generated documentation.

Q: "If you had a human assistant, what duties would you delegate?"
Business Analyst Response:

Delegate: "Run validation checks, flag errors, upload to cloud, generate reports"
Keep: "Decide whether to accept/reject biased datasets based on business context"

→ Insight: Automate validation/cleaning/upload. Augment bias interpretation—AI flags issues, humans make business decisions.

The **ClimaSmart** project leans towards **automation** for weather prediction and data processing:

- **Automation**: Tasks such as data ingestion, real-time data updates, and model retraining are fully automated using Airflow DAGs and CI/CD pipelines. Automation ensures that predictions are always up-to-date, accurate, and scalable without human intervention.
- **Augmentation**: For user interactions (e.g., chatbot queries), AI augments the user's ability to get instant, personalized weather advice, improving decision-making for activities or travel.

AI improves the efficiency and reliability of predictions, while also enhancing the user's ability to interact with the system in a meaningful way.

3. Design your reward function

Once your team has had a chance to digest your recent research on user attitudes towards automation and augmentation, meet as a team to design your AI's **reward function**. You'll revisit this exercise as you continue to iterate on your feature and uncover new insights about how your AI performs.

Use the template below to list out instances of each reward function dimension.

Reward function template

		Prediction: Positive	Prediction: Negative
Reference	Positive	True Positive {Example 1} {Example 2} {Example 3}	**False Negative** {Example 1} {Example 2} {Example 3}
Reference	Negative	**False Positive** {Example 1} {Example 2} {Example 3}	True Negative {Example 1} {Example 2} {Example 3}

Take a look at the false positives and false negatives your team has identified.

- If your feature offers the most user benefit for **fewer false positives**, consider optimizing for **precision**.

- If your feature offers the most user benefit for **fewer false negatives**, consider optimizing for **recall**.

Our AI model will be optimized for recall
because the primary user benefit is to ensure that important, low-frequency weather events, such as storms or extreme temperatures, are captured to prioritize safety and preparedness.
We understand that the tradeoff for choosing this method means our model will occasionally generate false positives, leading to some unnecessary alerts for less critical weather changes. However, this is ~~acceptable as missing significant weather events poses a greater risk~~ to users.

For this project, the reward function should be optimized for **recall**, as weather predictions—particularly for extreme or rare weather events—are critical to user safety. By prioritizing recall, the system ensures it captures as many important weather events as possible, even if this means occasionally providing less precise predictions for common conditions.

- **False Negatives (low recall)**: Could lead to missing important events like storms or extreme temperatures, which could result in user harm.
- **False Positives**: Less impactful, as over-predicting adverse weather can be inconvenient but doesn't pose as great a risk as missing critical warnings.

This tradeoff ensures that users are alerted to weather changes as early as possible, prioritizing safety and preparedness over perfect precision.

4. Define success criteria

Now that you've done the work to understand whether AI is a good fit for your user need and identified the tradeoffs of your AI's reward function, it's time to meet as a team to define success criteria for your feature. Your team may come up with multiple metrics for success by the end of this exercise.

By the end of this exercise, everyone on the team should feel aligned on what success looks like for your feature, and how to alert the team if there is evidence that your feature is failing to meet the success criteria.

Success metrics framework

Start with this template and try a few different versions:

> If **__{ specific success metric }__**
>
> for **__ { your team's specific AI driven feature }__**
>
> **{ drops below/goes above }__ { meaningful threshold }__**
>
> we will **__{ take a specific action }__.**

To evaluate the success of the ClimaSmart platform, several key metrics will be tracked:

- Prediction Accuracy: The goal is to maintain at least 90% accuracy. This will be monitored using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). If accuracy falls below this threshold, an automatic model retraining will be triggered through Airflow.
- Response Time: The chatbot response time should remain under 2 seconds to ensure a seamless user experience, especially when handling multiple queries or during peak traffic times. Monitoring tools like Prometheus and Grafana will track system performance and trigger alerts if response time exceeds this threshold.
- Automation Efficiency: Airflow DAGs should automate 95% of the pipeline processes (data ingestion, model training, and deployment), reducing the need for manual intervention and ensuring scalability. Success will be measured through the number of automated tasks that are completed without errors or delays.

These criteria ensure that the system remains user-friendly, accurate, and scalable while delivering real-time predictions. Monitoring tools are set up to alert the team if any of these metrics fall below acceptable levels.

If validation pass rate for ml_data_pipeline drops below 85% over a rolling 7-day window, we will investigate data source quality, review schema profiles, and adjust validation thresholds or contact upstream data providers.alidation pass rate for ml_data_pipeline drops

Version 2

If bias detection flags exceed 10 per dataset on average for 3 consecutive pipeline runs, we will review data collection processes, implement stratified sampling, and evaluate whether bias reflects true distribution or collection issues.

Version 3

If GCS upload success rate drops below 95% over 30 days, we will audit IAM permissions, review network reliability, increase retry attempts, and consider implementing local fallback storage.

Statement iteration

Take each version through this checklist:

- ☑ Is this metric meaningful for all of our users?
  - ☑ How might this metric negatively impact some of our users?
- ☑ Is this what success means for our feature on day 1?
  - ☑ What about day 1,000?

Final version

If If end-to-end pipeline success rate drops below 90% over a rolling 7-day window, we will conduct root cause analysis, review Airflow logs and error patterns, adjust retry logic or validation thresholds, and notify stakeholders of systemic issues requiring upstream data provider intervention.end-to-end pipeline success rate drops below 90% over

Schedule regular reviews

Once you've agreed upon your success metric(s), put time on the calendar to hold your team accountable to regularly evaluate whether your feature is progressing towards and meeting your defined criteria.

Success metric review

Date:

Attendees: