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
2025	Bluebikes Trip Data Analysis	70.5% member usage indicates commuter dependency; peak demand 8am/5-6pm creates critical availability windows
2020	NACTO Shared Micromobility Report	Industry average of 1.7 rebalancing trips per bike per month; operational costs 20-30% of revenue
2015	NYC Citi Bike Analysis (Singhvi et al.)	Predictive models achieved 75-85% accuracy for station demand; reduced rebalancing costs
2014	Weather Impact Study (Gebhart & Noland)	Temperature below 32°F reduces ridership by 30%; precipitation reduces by 25%
2024	Station Analysis	Top 10 stations handle 35% of trips; MIT/Central Square are consistent high-demand locations
2018	Caggiani et al. Dynamic Management Study	Optimization reduces operational costs by 10-35% compared to fixed schedules

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.

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 core experience requires prediction of future events. ☒ Personalization will improve the user experience. ☐ User experience requires natural language interactions. ☒ Need to recognize a general class of things that is too large to articulate every case. ☒ 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.	☐ The most valuable part of the core experience is its predictability regardless of context or additional user input. ☐ The cost of errors is very high and outweighs the benefits of a small increase in success rate. ☐ Users, customers, or developers need to understand exactly everything that happens in the code. ☐ Speed of development and getting to market first is more important than anything else, including the value using AI would provide. ☐ People explicitly tell you they don’t want a task automated or augmented.

We think AI **{ can / can not }** help solve _______**{ user need }**________, because

The problem requires predicting future bike demand 1-6 hours ahead based on complex patterns (temporal, spatial, weather, and user-type interactions) that are impossible to capture with rule-based programming.

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

- Describe your first impression of this feature.
- How often do you encounter the following problem: [insert problem/need statement here]?
    - Daily
    - Often (a few times a week)
    - Sometimes (a few times a month)
    - Rarely (a few times a year)
    - Never
- How important is it to address this need or problem?
    - Not at all important
    - Somewhat important
    - Moderately important
    - Very important
    - Extremely important

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

Reference		Positive	Negative
	Positive	**True Positive** **Example 1:** Predict 80 bikes needed at North Station during morning rush hour; actual demand is 75 bikes — stations remain stocked, customers satisfied. **Example 2:** Forecast surge near TD Garden for Celtics game; 120 bikes requested as predicted — proactive rebalancing prevents empty stations. **Example 3:** Model predicts rainy day low demand (20 bikes); actual usage is 18 bikes — efficient resource allocation, no wasted rebalancing.	**False Negative** **Example 1:** Predict 30 bikes needed at Back Bay station; actual demand is 70 bikes — station empties by 8 AM, 40 customers unable to rent, revenue lost. **Example 2:** Miss Boston Marathon demand spike; predict normal weekend traffic but thousands of spectators need bikes — system-wide shortages, brand damage. **Example 3:** Underestimate sunny Saturday demand at Charles River stations; predict 50 bikes, need 120 — recreational riders frustrated, negative social media posts.
	Negative	**False Positive** **Example 1:** Predict 100 bikes needed for anticipated Red Sox game that gets rained out; only 40 used — wasted rebalancing effort but no customer harm. **Example 2:** Over-stock Harvard Square expecting exam week surge; predict 80 bikes, only 55 rented — operational inefficiency, bikes idle but available. **Example 3:** Forecast high commuter demand on Monday; predict 90 bikes, actual need is 60 due to unexpected holiday — minor resource misallocation.	**True Negative** **Example 1:** Predict low demand at suburban station during midday weekday; 5 bikes needed, 7 used — efficient allocation, no over-stocking. **Example 2:** Forecast minimal demand during winter snowstorm; predict 10 bikes system-wide, actual usage 12 bikes — appropriate conservative stocking. **Example 3:** Model predicts overnight low demand (2 AM - 5 AM); minimal bikes allocated, minimal usage — optimal resource distribution during off-peak.

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______**{ precision / recall }**________

because_____________**{ user benefit }**_____________________________

Higher recall means we'll predict demand generously, ensuring bikes are available when users need them. This prevents the critical failure mode of empty stations, which directly impacts revenue and customer satisfaction. We'd rather occasionally over-stock a station than leave customers unable to find a bike.

This prevents the critical failure mode of empty stations, which directly impacts revenue and customer satisfaction. We'd rather occasionally over-stock a station than leave customers unable to find a bike.

We understand that the tradeoff for choosing this method means our model will ______**{ user impact }**____________________________________

Occasionally over-predict demand, resulting in wasted rebalancing effort and bikes sitting idle at over-stocked stations. However, this operational inefficiency is preferable to the alternative: customers arriving at empty stations, causing immediate revenue loss, frustration, and damage to Bluebikes' reputation.

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 }__.**

Version 1

If prediction accuracy (MAPE) for station-level hourly demand forecasts drops below 85% for three consecutive days, we will halt automated rebalancing recommendations, switch to manual oversight mode, and immediately investigate data pipeline integrity and model drift.

Version 2

If customer complaints about empty/full stations for our ML-driven bike distribution system goes above 50 complaints per week (20% increase from baseline), we will temporarily increase our recall threshold to prioritize bike availability over operational efficiency and conduct urgent user research with affected stations.

Version 3

If unnecessary bike rebalancing trips (moves to stations that remain >80% full/empty) for our demand prediction system goes above 30% of total rebalancing operations, we will recalibrate our precision-recall balance, add station capacity constraints to the model, and retrain with weighted penalties for over-predictions.

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 prediction accuracy (MAPE) for station-level hourly demand forecasts drops below 85% for three consecutive days, we will investigate root causes (data quality, seasonal shifts, external events), notify the operations team of reduced confidence predictions, and retrain the model with recent data while reverting to conservative historical demand patterns as a fallback.

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: 10/28/2025

Attendees: Pranav, Ananya