B
MHCI

BLOOMINEERS

SPRING RESEARCH REPORT

Carnegie Mellon University

Bloomberg

IMPROVING THE MACHINE LEARNING MANAGEMENT WORKFLOW BY REDUCING COMPLEXITY

EXECUTIVE SUMMARY

01 Context

Through its Terminal, Bloomberg L.P. provides financial software tools, such as an analytics and equity trading platform, data services and news, to financial companies. As technological demands accelerate, data science and machine learning have begun to play an increasingly important role at Bloomberg. Machine learning is being applied to nearly every facet of the business – from data acquisition through analysis, to real-time news alerts and story generation.

However, applying machine learning at scale has created friction around experiment management, performance tracking, and collaboration, thus driving the need to manage some of this complexity. Bloomberg brought on our five-person team to work with machine learning practitioners and engineers who design the classification, prediction, and annotation systems in order to understand existing systems, culture, and processes for experiment management across various teams.

Our goal is to design a platform for managing machine learning experiments and tracking performance, parameters, and other metadata to enhance reproducibility and knowledge sharing. Ultimately, this will simplify the machine learning model training process, allowing for higher success rates of experiments and the delivery of increasingly sophisticated products powered by machine learning.

02 Process

Our process began by building domain knowledge through a literature review. We used this as an opportunity to discover potential pain points and varying perspectives on the problem space. With a better understanding of common practices, we then analyzed the key features and capabilities of four machine learning management software.

The research phase began with an onsite visit to Bloomberg's headquarters, where we interviewed our primary users, machine learning practitioners. We then began conducting remote contextual inquiries to better understand the various workflows across six different teams.

We ran several other research methods in parallel, including a survey to help quantify our research results, and a design thinking activity to better identify how machine learning engineers feel about their current system. This prepared us to develop a master workflow for design iterations.

03 Insights

- **Tracking, Discoverability, Documentation**

  Ineffective tracking leads to further issues in documentation and discoverability.

- **Three Components of Tracking**

  The machine learning workflow is comprised of three interdependent components such as data, code, and results, which are all reliant on effective tracking.

- **Workarounds as Substitutes**

  Because of system limitations, machine learning engineers resort to developing their own workarounds to overcome workflow challenges.

04 Next Steps

- During the summer, we will use our research findings to inform the design of our solution. Having already developed some preliminary visions, we will begin by ideating further to hone in on the value of experiment tracking, facilitating better documentation and discoverability. Together, these drive specific downstream benefits such as simplified model training, higher success rates of experiments, and increased collaboration amongst stakeholders.

We plan to then iterate upon this design by creating lo-fidelity prototypes and performing targeted usability studies. For our final deliverable, we will develop a working prototype, which will be presented to Bloomberg in our summer presentation.

TABLE OF CONTENTS

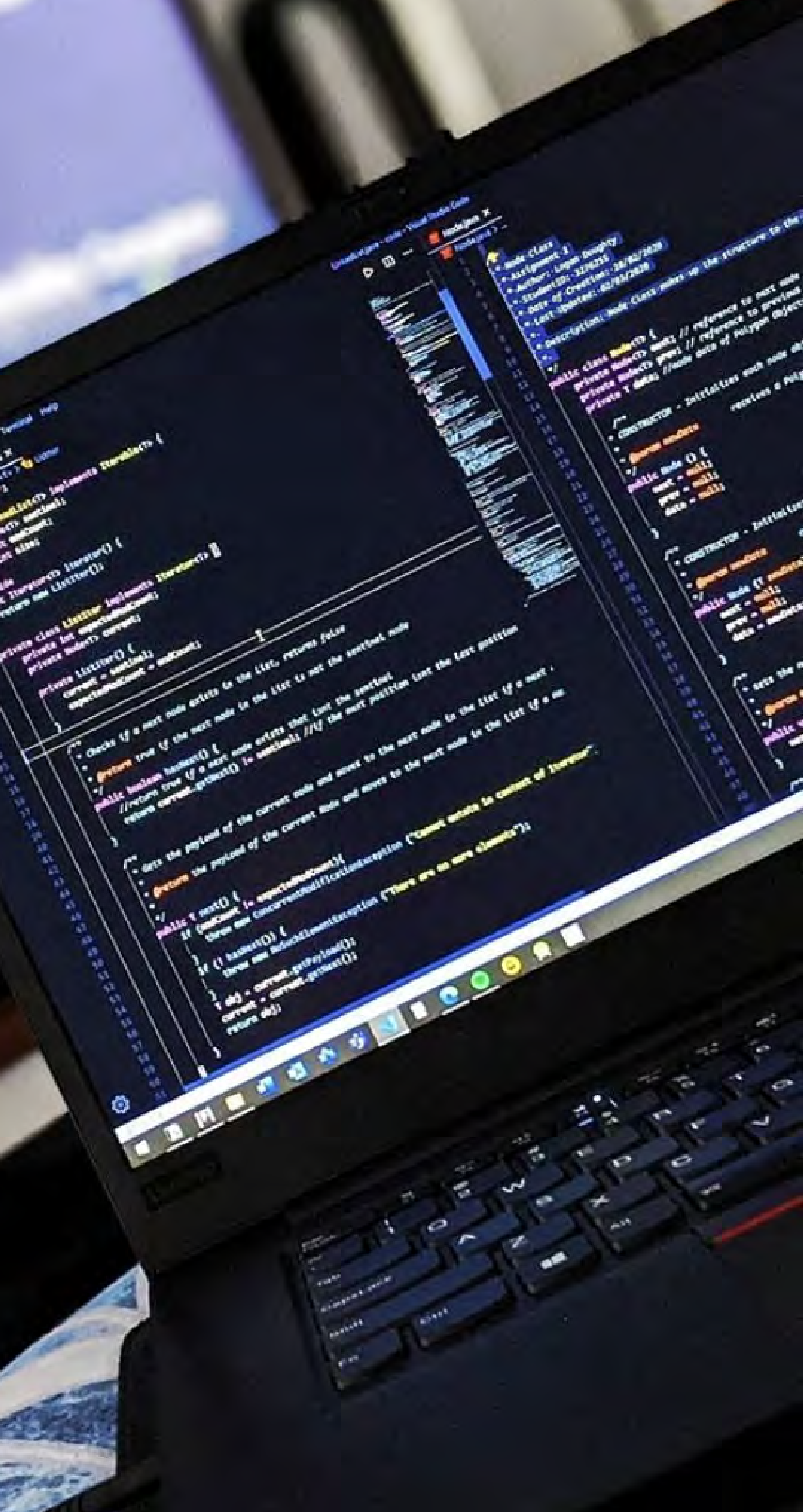
INTRO

RESEARCH

SYNTHESIS

DESIGN

SUMMER



01 Project Scope

Bloomberg has tasked our interdisciplinary team to design a solution for machine learning engineers at the organization within the timeframe of 28 weeks. The final working prototype will not only enable enhanced organization and visibility into their experiments, but allow the engineers to more seamlessly track, share and reproduce experiments.

To gain an understanding of the problem space, we chose to use a number of methods, including contextual inquiry, participatory design, storyboarding, and surveys.

From initial research findings, we've identified numerous challenges both within teams and across the organization.

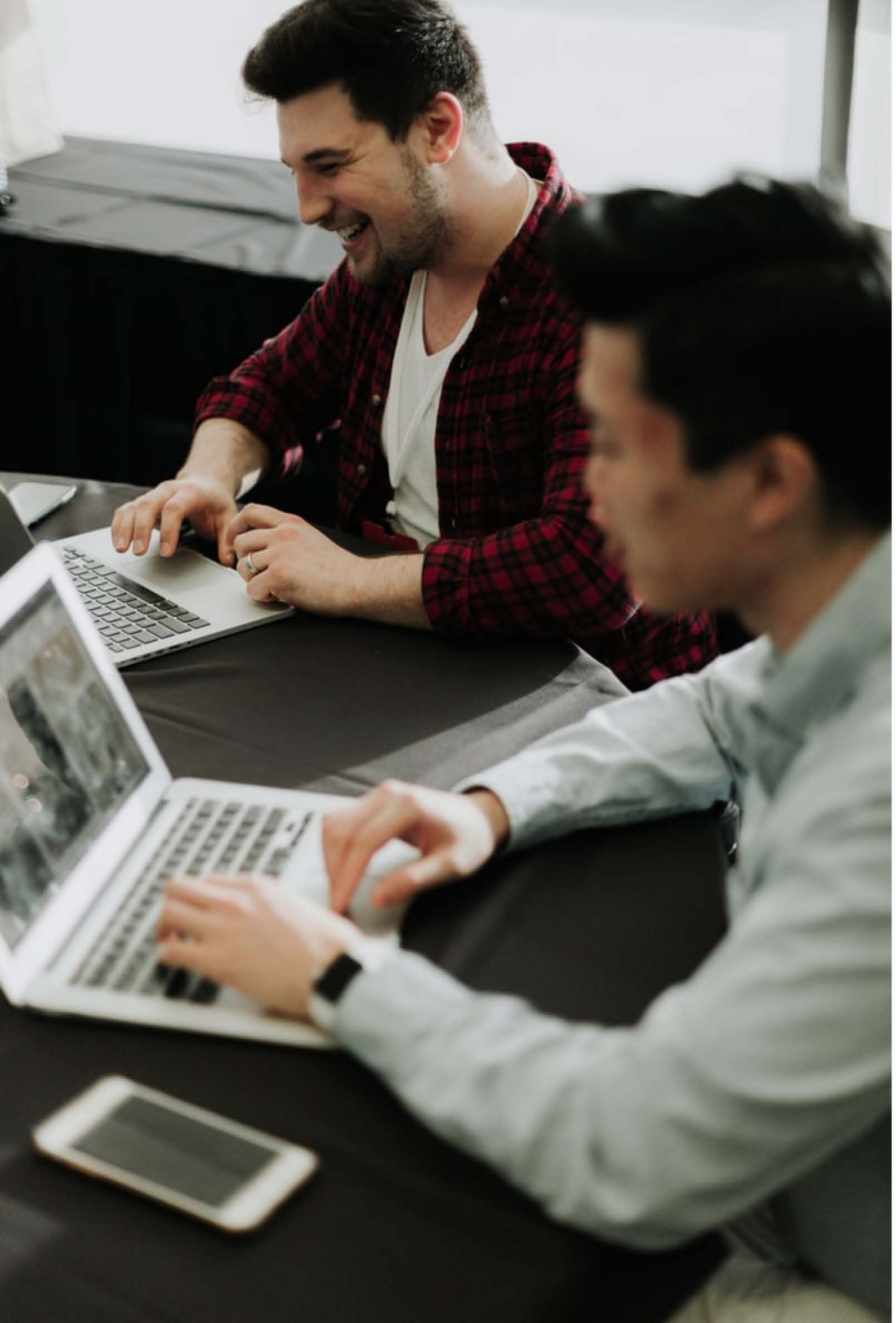
Our goal is to take opportunities from the research findings and use them to generate design ideas. In the summer semester, our design process will begin by creating prototypes for the experiment management platform, which we'll validate through usability for the creation of enhanced workflows.

MEET THE INSIGHTS

01 Ineffective tracking leads to further issues in documentation and discoverability.

02 The machine learning workflow is comprised of three interdependent components such as data, code, and results, which are all reliant on effective tracking.

03 Because of system limitations, machine learning engineers resort to developing their own workarounds to substitute workflow challenges.



03 User Spotlight

▶ Founded in 1981 by Michael Bloomberg, Bloomberg is a 39 year-old financial technology giant. Bloomberg manages massive amounts of data in a real-time environment, making machine learning an increasingly important area for the company to focus on.

Within the AI Engineering department, machine learning practitioners and engineers are developing the classification, prediction, and annotation systems represent global markets.

There are six divisions under AI Engineering, each focusing on different functions, such a text enrichment or natural language processing. For our research, we interviewed one representative team from each division.

RESEARCH GOALS

01 Understand the problem space

Gain an in-depth understanding of the realm of machine learning through primary research, secondary research, and taking online courses in machine learning.

02 Visualize Bloomberg ML engineers’ workflows across different teams

Map out a master workflow– the “happy path”– and identify where different teams diverge from this.

03 Identify stakeholders’ pain points and areas of opportunity

Pinpoint and quantify pain points in order to target areas where we could have the most impact for product managers and ML engineers.

04 Research through design

Test out our assumptions and reframe the direction of the project through storyboarding and visual storytelling.

THE BLOOMINEERS

The team is comprised of five inter-disciplinary members who come from a diverse set of backgrounds and skills. The team is carefully selected by the department faculty and advisor at Carnegie Mellon University. Together, each with their unique set of skills, the Bloomineers balance the core areas in techno-logy, design, and research to bring success to this project.

DANIELLE SHOSHANI

PRODUCT MANAGER

Danielle studied International Relations and Communications at University of California, Santa Barbara, and worked as a Global Communications Manager for five years at a B2B tech com-pany developing market research strategies and product awareness.

NEHA CHOPADE

DESIGN RESEARCHER

Neha is one of our research leads. During her work as real estate-based UX researcher, she conducted ethnographic research into the unregulated real-estate sector of India and produced roadmaps and insights for the design and management teams.

NORMAN KANG
UX ENGINEER

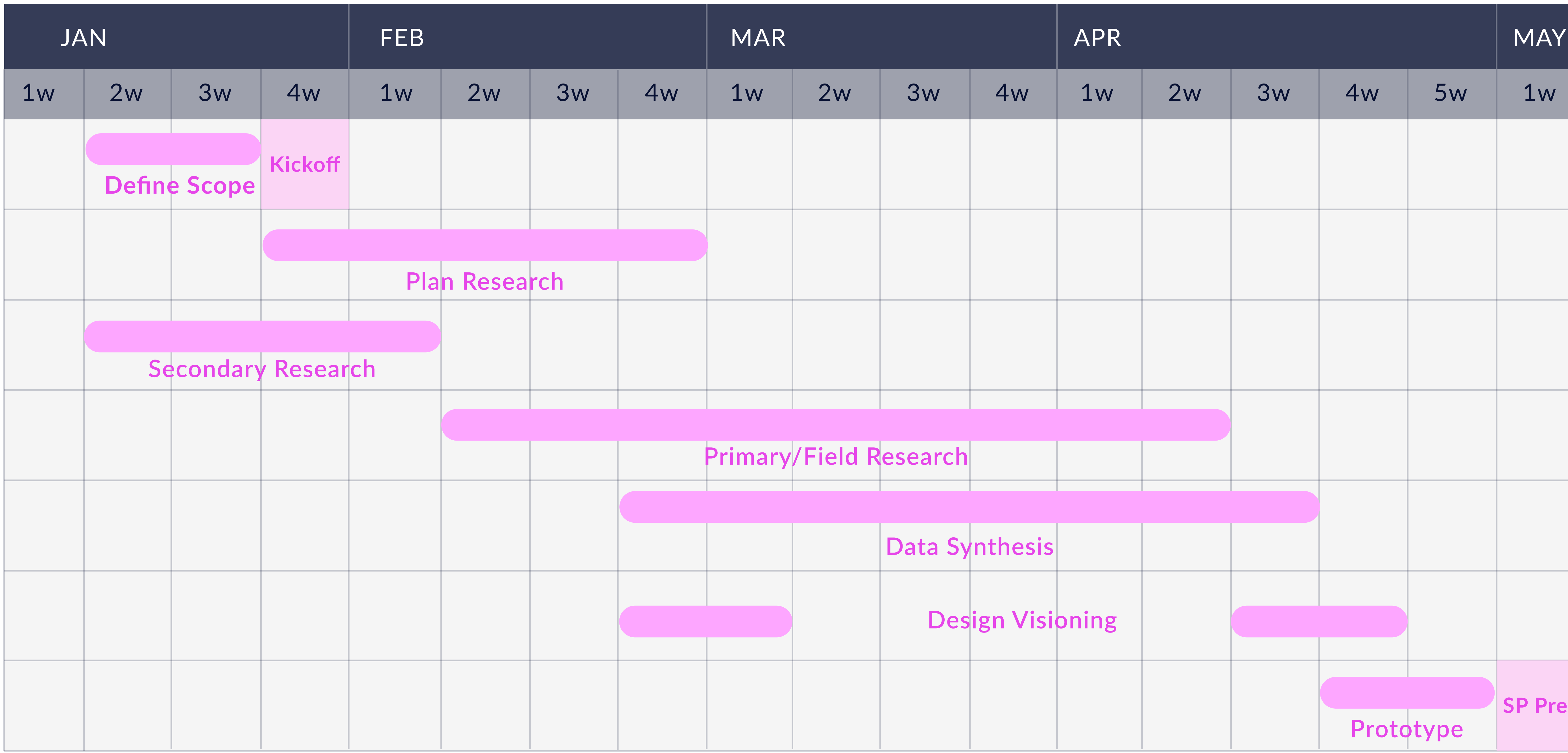
Norman is our Technical Lead and a self-taught front-end developer. He studied Bio-chemistry at UC San Diego and worked at a nationally recognized web development agency for 2 years. Norman brings his analytical skills along with a passion for design.

CHI HUANG
DESIGN RESEARCHER

Chi is one of our research leads. She studied Psychology and Communications at University of Washington. Her research work contributed to the study of gender discrepancy in STEM fields to fuel evidence-based interventions.

AMY LU
PRODUCT DESIGNER

Amy studied fashion design at Parsons School of Design and worked as a fashion designer for five plus years. As the design lead, Amy connects her under-standing of customer research from her work experience with human-centered design.


JAN
FEB
MAR
APR
MAY
1w
2w
3w
4w
1w
2w
3w
4w
1w
2w
3w
4w
1w
2w
3w
4w
5w
1w
Kickoff
Define Scope
Plan Research
Secondary Research
Primary/Field Research
Data Synthesis
Design Visioning
SP Pres
Prototype

RESEARCH

RESEARCH

Since machine learning was uncharted territory for all team members when we first began, research is an especially critical phase– it's the pathway for us to understand the problem space, empathize with users, and uncover unmet needs.

In order to submerge ourselves in this space, we conducted in-depth primary and secondary research before meeting with our stakeholders at Bloomberg. This included literature reviews and interviewing data scientists, ML engineers, and project managers. With a shared ML language, we then conducted various research methods better understand and visualize where the biggest pain points were for our stakeholders and where we could bring the most value.

Our approach was to have research and design run in parallel, as one informs the other, in this iterative process of identifying pain points, testing out our assumptions, and adjusting our designs.

ENTERING THE DOMAIN

01 Literature Review

02 Interviews with Faculty & Model Users

03 Competitive Analysis

01 Literature Review

We researched academic papers on machine learning experiment management to gain domain knowledge and better understand the problem space. The literature review was also used to discover existing pain points and varying perspectives.

In addition, we reviewed literature from adjacent problem spaces that deal with tracking multiple moving parts of a complex system, such as healthcare. This was done to help keep our perspective open and create ideas outside of common practice. The literature review also informed and reinforced our proposed pain points of the ML process.

There was one paper that was particularly influential in our process. Its web-based dashboard summarized model performance in a bar chart as well as a scatterplot that displayed the relationships between hyper-parameters and performance metrics.

Key Takeaways:

i It is recommended to utilize **different visualizations** such as charts and scatterplots to compare experiment results and provide different perspectives of the data.

ii **Hierarchy is key.** Organization of a project in file format allows for efficient access to results.

Runway: machine learning model experiment management tool

Jason Tsay, Todd Mummert, Norman Bobroff, Alan Braz, Peter Westerink, Martin Hirzel
IBM Research, Yorktown Heights, NY, USA
{jason.tsay,braz.alan}@ibm.com,{mummert,bobroff,peterw,hirzel}@us.ibm.com

ABSTRACT

Runway is a cloud-native tool for managing machine learning experiments and their associated models. The iterative nature of developing models results in a large number of experiments and models that are often managed in an ad hoc manner. Runway is a workflow and framework independent tool that centrally manages and maintains metadata and links to artifacts needed to reproduce models and experiments. Runway provides a web dashboard with multiple levels of visualizations to evaluate performance and enable side-by-side comparisons of models and experiments.

1 INTRODUCTION

Machine Learning (ML) models are increasingly at the core of applications and systems. The process around developing these models is highly iterative and experiment-driven [4]. The often non-linear and non-deterministic nature of implementing ML models [7] results in a large number of diverse models. Through interviews we find that data scientists tend to manage models using ad hoc methods such as notebooks, spreadsheets, file system folders, or PowerPoint slides. However, these ad hoc methods record the models themselves but not the higher-level experiment. For example, a data scientist developing a natural language classifier may wish to compare models from a support vector machine experiment to ones from neural networks. At best, extra effort must be spent to manage experiments and their models and at worst, effort is wasted on what one interviewee called "dead-end trials." Given the increasing complexity and required computational time for ML models, reducing effort on experiments may greatly improve the efficiency of data scientists' workflows. At the same time, data scientists also tend to work idiosyncratically with pipelines and workflows unique to the task at hand. This variety means that a "one size fits all" approach is insufficient.

To address these challenges, we introduce Runway, a prototype ML model experiment management tool with the following design goals: (1) multiple levels of model management, (2) workflow and framework independence, (3) visual tools for model evaluation and comparison, and (4) cloud-native architecture that allows for easy integration with existing platforms. Runway is currently internally available to data scientists at IBM. Runway's design is also informed by a series of interviews with 27 data scientists at IBM from a wide variety of domains and by iterative agile development with sponsor users.

2 RELATED WORK

We position our work in a burgeoning field of engineering that assists in developing ML models and applications. Kim et al. [4] find through interviews that data scientists fulfill multiple important engineering roles towards connecting software systems to "real-world" data. They find something that we confirm in our own interviews: the sheer variety of titles, backgrounds, domains, and tasks for data scientists. An important commonality however between data scientists is familiarity with experiment-driven work or, as one interviewee put it, "I am used to designing experiments." Patel et al. [8] find through interviews and studies with data scientists that the highly iterative and exploratory nature of developing ML models is a primary challenge. In particular, multiple aspects of the seemingly linear workflow interconnect and ML developers would waste time on dead-end experiments. They also find that for many tasks, evaluating performance is often more difficult than simply evaluating metrics.

Closely related to our work and model management tools are tools that support general ML engineering such as Gestalt [6] or TFX [1] and in particular model management tools such as ModelDB by Vartak et al. [9], ModelHub by Miao et al. [5], and MLModelScope[1]. Model management tools are concerned with indexing and tracking large numbers of ML models for future sharing, querying, and analysis. Such tools support data scientists in sensemaking and identifying insights for their models. Runway builds on model management tools by supporting multi-level management such as experiments and including visual evaluation tools.

3 IMPLEMENTATION

3.1 Architecture

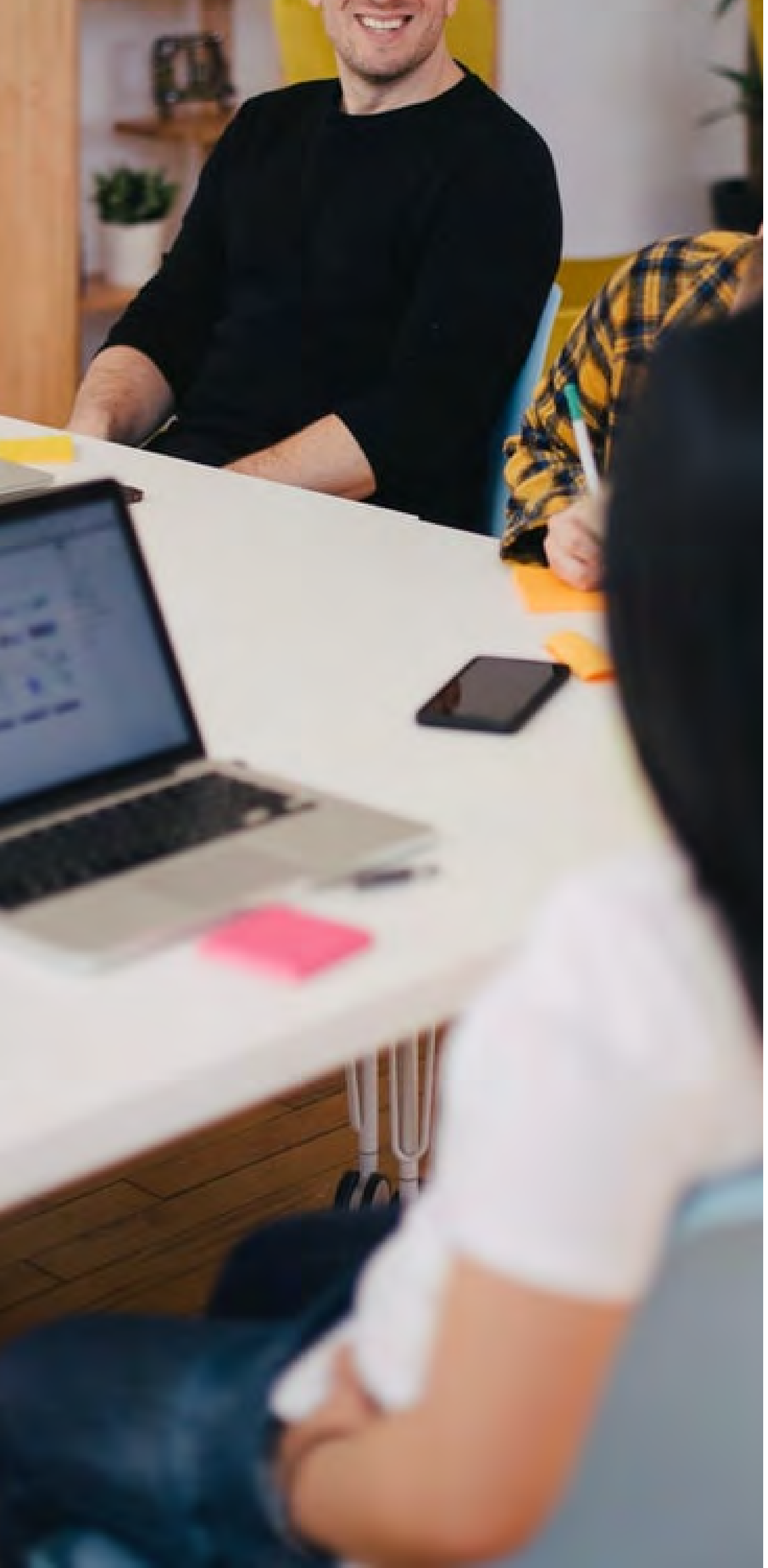
Figure 1 shows the high-level architecture of Runway, which consists of three key components: (1) a REST API backend which is the core of the architecture, (2) a Software Development Kit (SDK) that allows data scientists to instrument their own Python 3 scripts, and (3) a web-based dashboard interface. Runway is also designed to be cloud-native and integrates easily with other services such as cloud object storage[2] and the IBM Deep Learning (DL) Service [2].

Figure 1: Runway high-level architecture.

Runway stores and organizes metadata about ML models in a hierarchy of Projects, Experiments, and Runs. *Projects* represent

SYSML '18, February 2018, Stanford, CA, USA
2018.

Runway: machine learning model experiment management tool by Jason Tsay, Todd Mummert, Norman Bobroff, Alan Braz, Peter Westerink, Martin Hirzel



02 Faculty

We aimed to further our domain knowledge in both machine learning and experiment management software by interviewing faculty at Carnegie Mellon. We interviewed Professor Jason Hong to discuss topics on machine learning and Professor Majd Sakr to review a machine learning experiment management project from the Computer Science department.

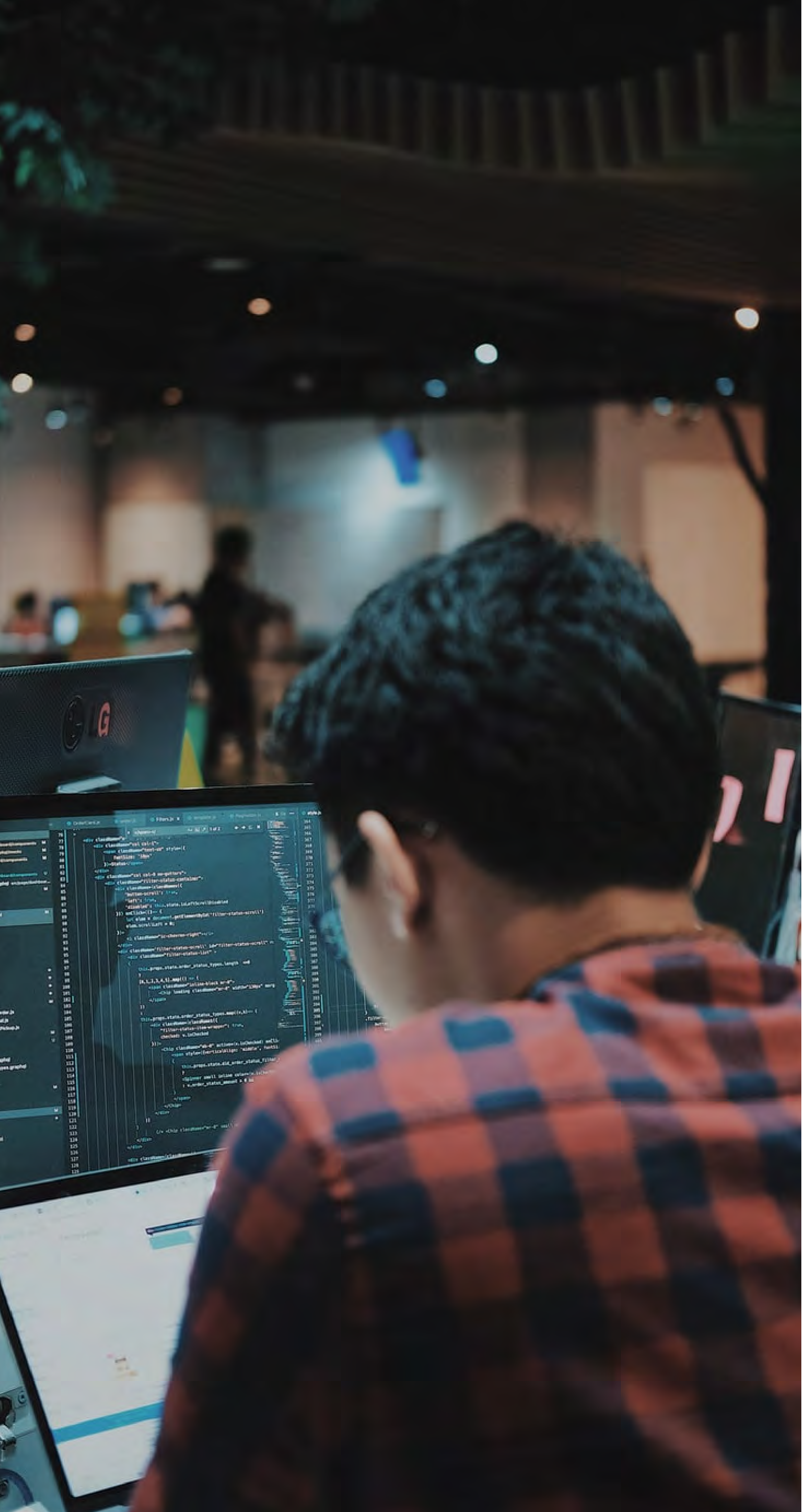
Professor Sakr provided us with tips he learned from his projects which we kept in mind during our research phase. Sakr found the experiment management process to be broad, so he recommended focusing on a specific area to provide strong targeted value. He also warned that our designs should be tested on the users who will be using the end product. These considerations helped us orient as we progressed into research.

Some findings from faculty interviews:

i Machine learning experiments are highly complex and nuanced. Each experiment can be a replication of the previous one. Instead of trying to design for the entire workflow, it may be more beneficial to zoom in on one specific use case.

ii Professor Hong challenged us to answer these questions in our research:

What do different teams do?
What do they want to see differently from each other?
Who manages the cluster?
Who has the priority?



02 Model Users

We then interviewed graduate students studying machine learning at Carnegie Mellon. These interviews provided us with a foundation early in the process for laying out the machine learning workflow.

We noticed there was a gap between machine learning in academia vs. industry. There wasn't a high demand for collaboration as students often worked in silos. Their experiment sprints were smaller in scale in terms of the amount of data collected.

Students also didn't have to worry about production of their models. While keeping these differences in mind, the model user interviews provided us with useful domain knowledge of the ML workflow process.

Some findings from student interviews:

- **i** In academic ML, there is less emphasis on building stable production models and more on exploration.
- **ii** Student experiments were generally on a smaller scale and would not require as many resources.
- **iii** A smaller scale equates to less requirements for tracking metrics.
- **iv** Academic machine learning workflows were similar to industry in the artifacts and tools they used for exploratory research.

03 Competitive Analysis

In this section we analyzed four machine learning management software and provided a breakdown of the key features/capabilities offered. Included in our analysis are guild.ai, comet.ml, neptune.ai, and mlflow.org.

Their features consist of:

i **Tracking** (metrics, artifacts, success rates, start/end times of experiments)

ii **Reproducibility** (ensuring experiments can be reproduced based on artifacts provided)

iii **Comparing experiments** (having a visual comparison what works and what doesn't)

iv **Collaboration amongst teams** (ability to see what members of the teams are currently working on to avoid duplication of efforts)

We found that there isn't a tool that effectively encompasses all the capabilities.

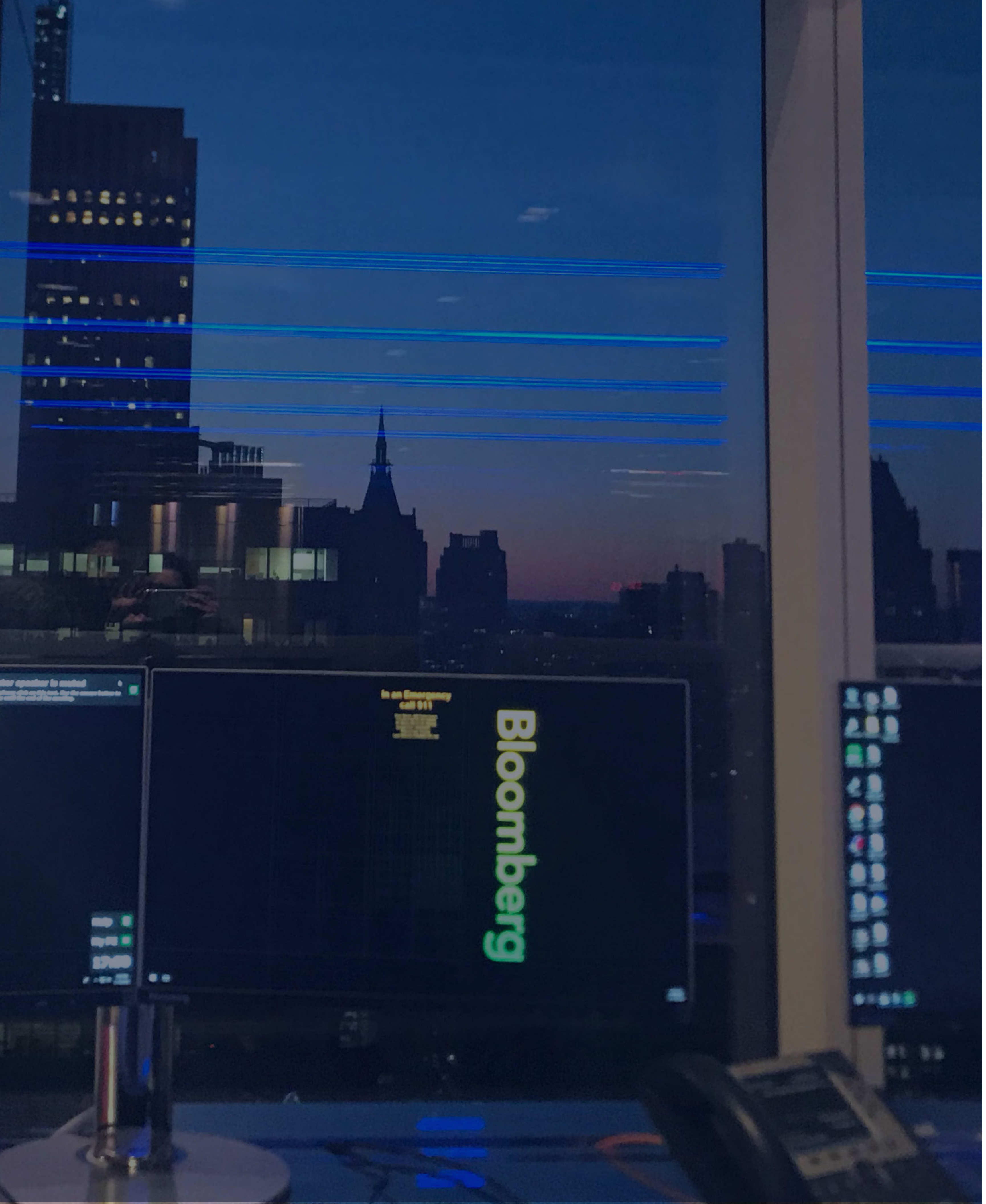
This raises two questions for our research in the space of experiment management at Bloomberg:

Are we designing into a white space or are we working on top of something pre-existing?

What workarounds do the engineers currently do to manage their experiments?

competitor management software

	Guild.ai	Comet.ml	Neptune.ai	Mlflow.org
01. TRACKING	✓		✓	✓
02. REPRODUCIBILITY	✓			✓
03. COMPARE EXPERIMENTS	✓	✓	✓	✓
04. COLLABORATION		✓	✓	



INTO THE FIELD

01 Overview

02 Sense Mapping

01 Overview

In order to answer the emerging questions, we planned a visit to Bloomberg's New York office to meet our primary users in person — the machine learning engineers, data scientists, and product managers.

We prepared an interview guide for two alternate research methods: contextual inquiry (building a master-apprentice relationship with our interviewees) and semi-structured interviews (directed narrative).

Bloomberg has six primary divisions under AI engineering. We were scheduled to meet with one representative team across all the divisions during the course of one working day.

After spending six hours onsite, we'd gained an in-depth view into the way each team worked. Although there were similarities in their overall process, we learned of quite a few differences in the way they were documenting and sharing experiments.

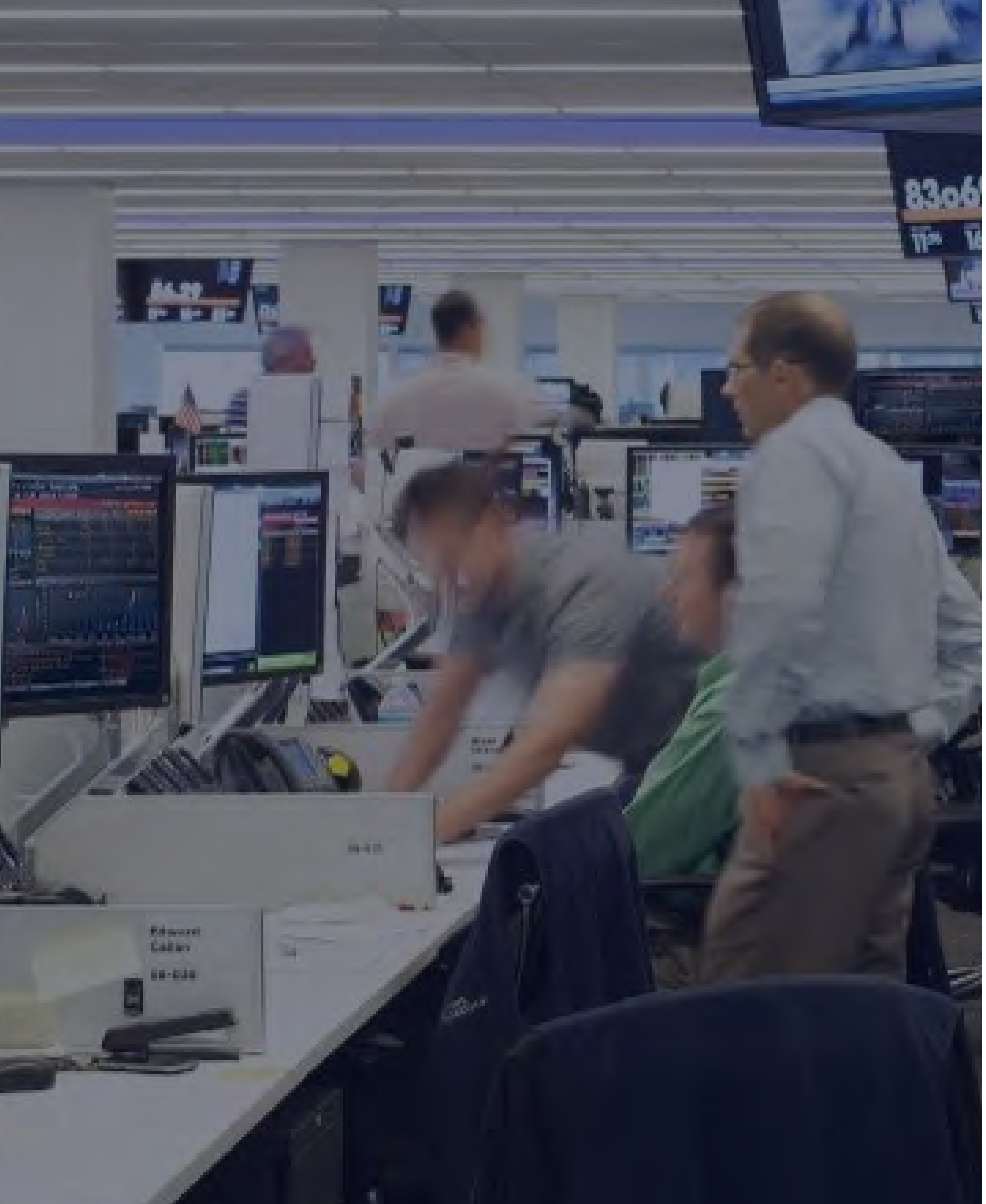
We left with an initial understanding of the machine learning workflow from the broader teams' perspectives, which would position us to later interview the ML engineers individually for more a in-depth explanation.

Neha at Bloomberg's New York office

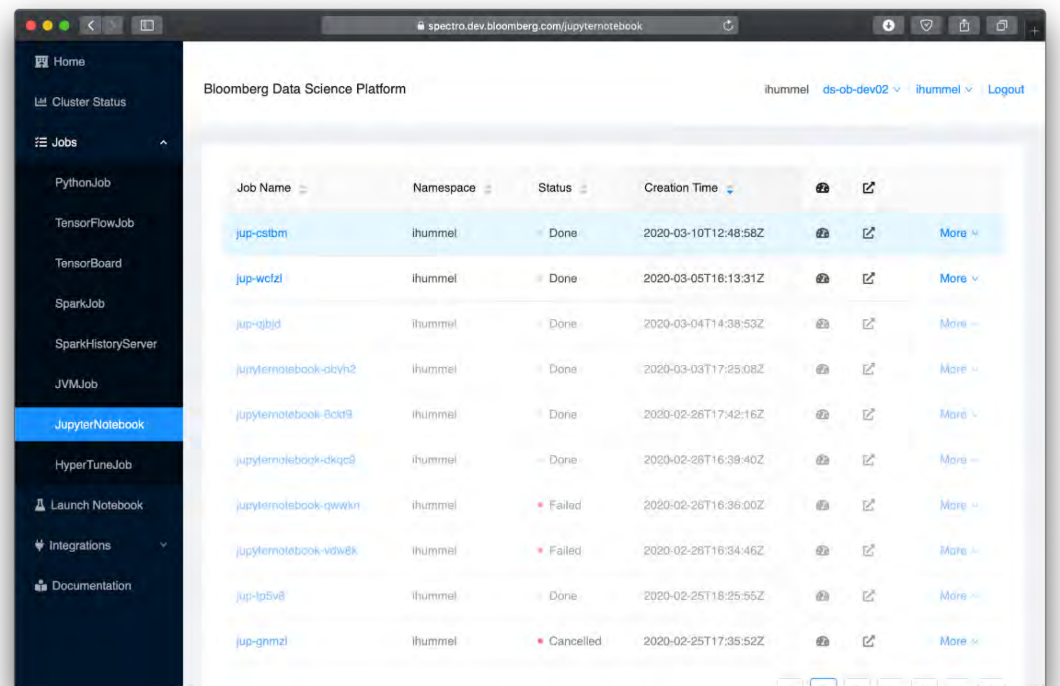
02 Sensemaking and Empathizing with the User

- We gathered our notes from the New York trip and color coded them across different teams on an Excel sheet. Using affinity clustering, we grouped these notes across need-based user statements.

  This exercise was pivotal for us in identifying the core framework for the machine learning workflow. The new framework helped us draft a sequence model workflow exercise for finding patterns during future research methods. We also uncovered gaps and missing information in the flow that needed further inquiry through semi-structured research in the form of a survey.



CURRENT STATE ANALYSIS



01 Overview

▶ The current state of machine learning management at Bloomberg is conducted on the Data Science Platform (DSP). The DSP's main function is to run ML experiments by connecting data and models with GPU, then displaying the results in logs.

We synthesized the research we received from semi-structured interviews, love letter/breakup letter, and the survey. The findings will drive our design for the future state, so we believe it will be critical to examine these more closely during the summer semester.

How do we allow for collaboration on this platform?

A Letter to DSP...

♥ Love

Please write a secret love letter/breakup letter to the DSP as if it were your romantic partner. Tell the DSP all your thoughts about it, everything you like and dislike, and why you're thinking and feeling that way.

Rest assured. Everything you write here is completely confidential and anonymous. We will not tell DSP your thoughts and opinions ☺.

Breakup

Dear DSP,

As I write this letter, our relationship has ended. I am deeply hurt, and my heart will undoubtedly be scarred because we can't be together. For example, sometimes it's just really hard to track all the experiments that I'm running because ....

Maybe someday our stars will align again.

With Deepest Love and Sympathy,

Michael

02 Love Letter/Breakup Letter

We asked the ML engineers to write either a love letter or breakup letter to the DSP– the machine learning management platform created internally. We wanted to understand how the tool currently available at Bloomberg assisted or hindered the productivity of their work.

By having them freely express their thoughts and feelings toward a software as if it were a romantic partner, we were able to understand in greater detail their current satisfactions, dissatisfactions, and unmet needs through rich qualitative data.

Unlike other research methods, the love letter/breakup letter gave us a glimpse into the emotions associated with working as a machine learning engineer at Bloomberg, uncensored.

It is hard due to a lot of random factors.

It is a chore.

It's the fun part!

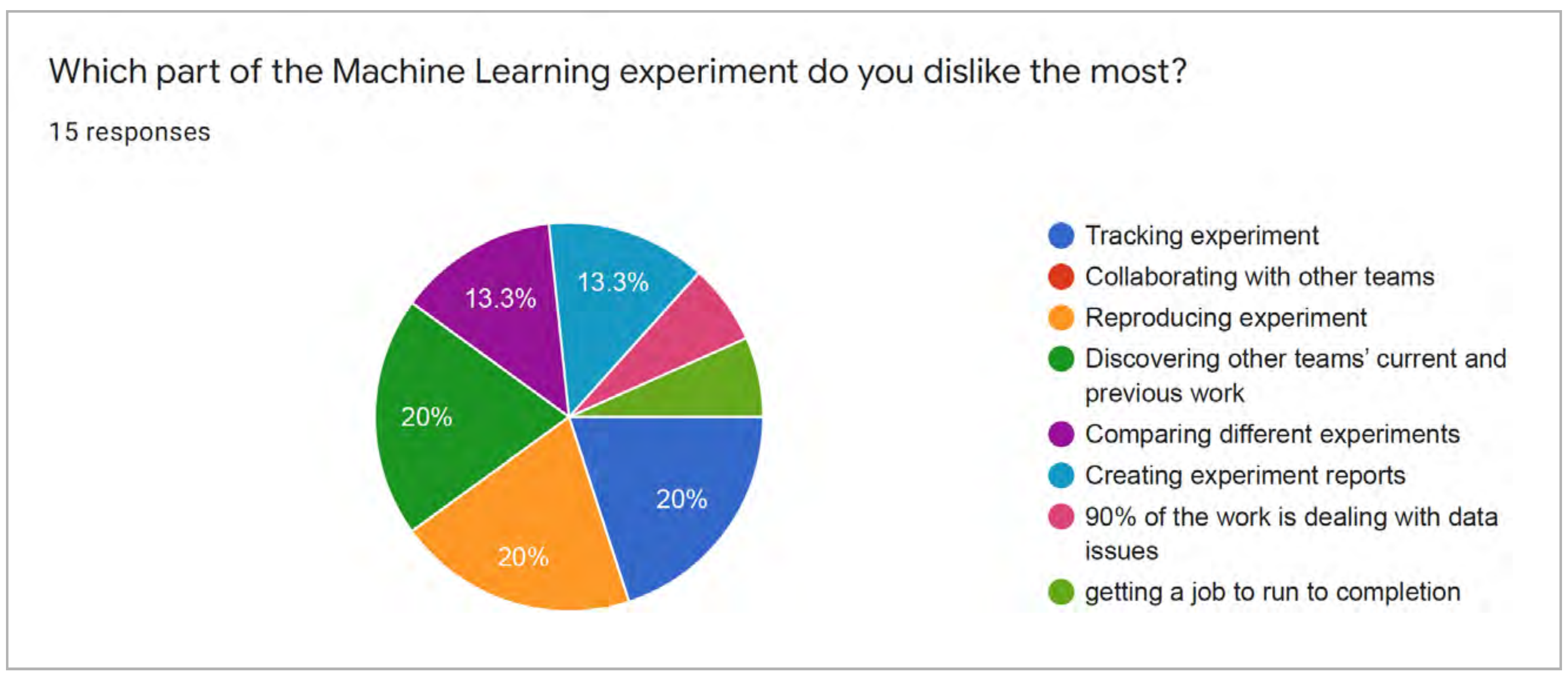
It is an interesting way to gain insights on problems.

Tracking Experiments

Comparing Experiments

It's tedious...but doesn't lead to tangible progress toward solving business problems.

I like to test my intuitions about what approaches work best for a task.



03 Surveys

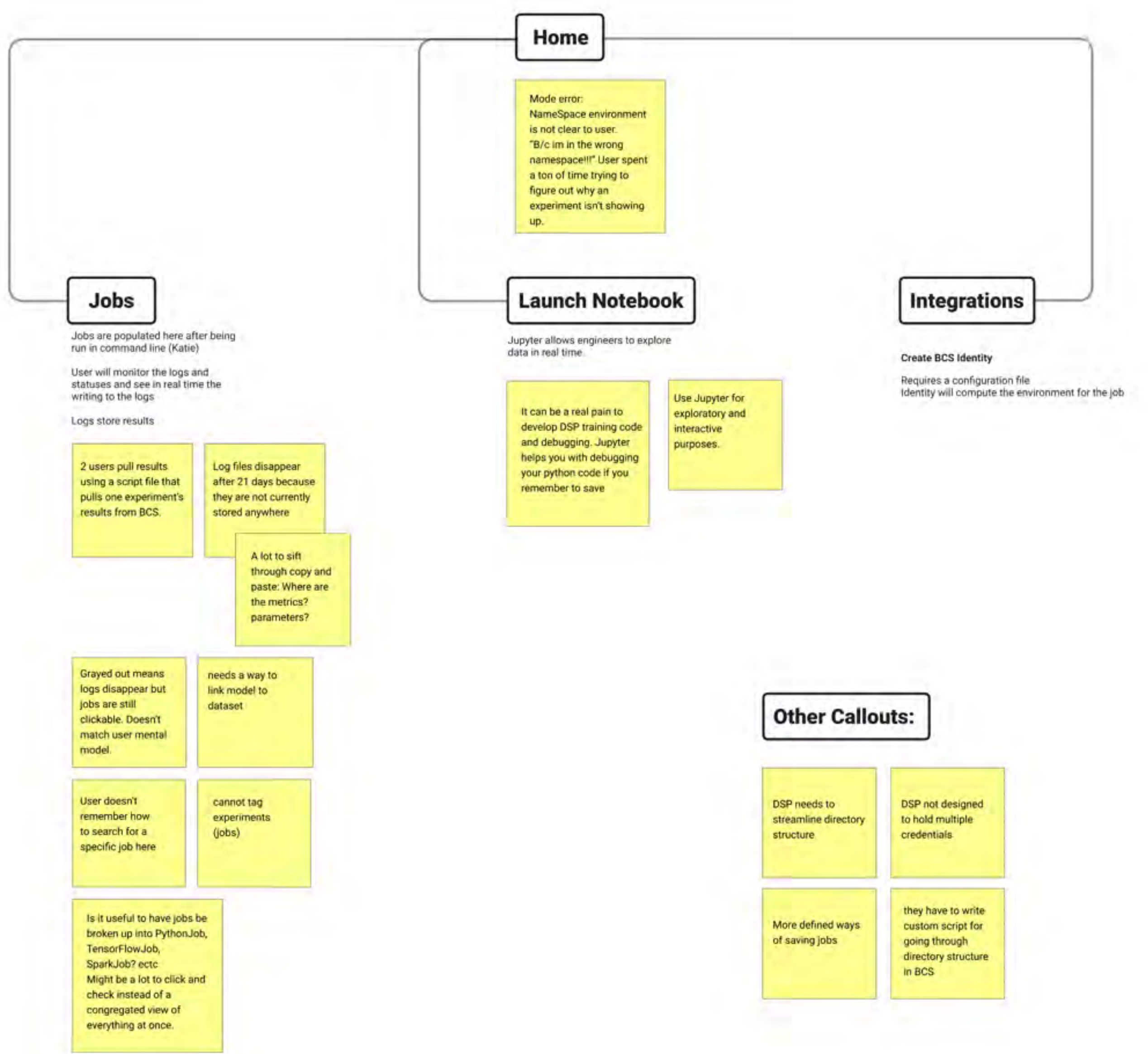
▶ We decided that developing and distributing a survey would be beneficial to:

i Quickly gain access to a larger user group.

ii Quantify the measures that were examined in the survey.

The overarching goal of our survey was to understand how ML engineers felt about each stage of their workflow. We wanted to specifically quantify which parts of their workflow they disliked and enjoyed the most and their reasoning behind their choice.

In the future, we plan to also use the results of this survey as a benchmark as we evaluate the effectiveness of our design.



04 Pain Points Identified:

i **Lack of experiment comparison.** Comparing experiments is ranked by the ML engineers as the most enjoyable part of their workflow, yet the existing tools make it very hard for them to compare experiments.

- The visualizations on the DSP that allow for hypertune job comparison are confusing, which leads ML engineers to manually copy and paste the performance metrics and compare the different runs using other tools.
- There is no streamlined way for them to aggregate and compare results in the DSP. Comparing experiments and different hypertune jobs require them to make many extra manual steps. For instance, one ML engineer stated that he/she needs to manually add boilerplate code to the different experiments, write them to HDFS, download them locally, and then analyze the results.

ii **Name Space environment is not clear to user.**

- Users spent quite a bit of time trying to figure out why an experiment wasn't showing up.
- *"Because I'm in the wrong Name Space!"*

iii The experiment homepage has pages and pages of experiments. **It's hard for user to know at a glance which experiment is which, or what each experiment was about/how it performed.**

- What were the metrics? Parameters? There's currently not a single way to figure out what are the best results for different metrics on the main page. There isn't an easy way to filter or search through them by time, hyperparameters, performance metrics either.
- *"I want to be able to filter experiments by application type, time period, hyperparameter and accuracy."*

Design opportunity:

Allowing ML engineers to tag an experiment or add a note next to an experiment on the main page could allow them to easily understand and parse through the experiments to find the ones they need.

iv **It is difficult for users to keep track of the experiment results** on Spectro because log files disappear after 21 days since they are not currently stored anywhere.

- Because the logs disappear after 21 days and there's no easy way to download the logs, ML engineers need to save the information in the logs in a roundabout way (e.g., copying and pasting).
- Currently, in order to save a log, they need to first try to delete a log, which would prompt them to save the log. This design is counterintuitive because if they want to delete a log, it means that they don't want it.
- Users are less inclined to use this feature knowing it will disappear after 21 days.
- *"My project might last longer than 21 days."*

iv **Currently, engineers are not able to share notebooks across Name Spaces, which hinders within team and cross-team collaboration.**

v **There have been several instances where DSP jobs get killed without notice, and the work that the job generated disappears.**

- If there's the need to run a big test on Jupyter Notebook that would last a couple days, ML engineers need to make sure to constantly back up the environment to S3 or HDFS because the session would automatically shut down after a certain time, and the work would get lost.

vi **There is no easy way to link the experiment to its appropriate Team<Go> page.**

- There's a disconnect between the experiments that were run and the summary of the experiments.

vii **The overhead and onboarding costs take up significant time and effort.**

- There are a lot of permissions that need to be granted, such as which server host, nameserver, and storage are needed.

viii **Sometimes when bugs about the DSP are reported, it takes a long time for the bugs to be addressed.**

ix **Capacities on memory, CPU, and GPU should be either expanded or better communicated.**

Design opportunity:

Instead of allowing the DSP to crash at the end, inform ML engineers beforehand when a job has exceeded the their allocated share of capacities on the DSP.

x **Currently, DSP only supports one optimization metric, but there are instances where the models that the ML engineers are using have more than one metric.**

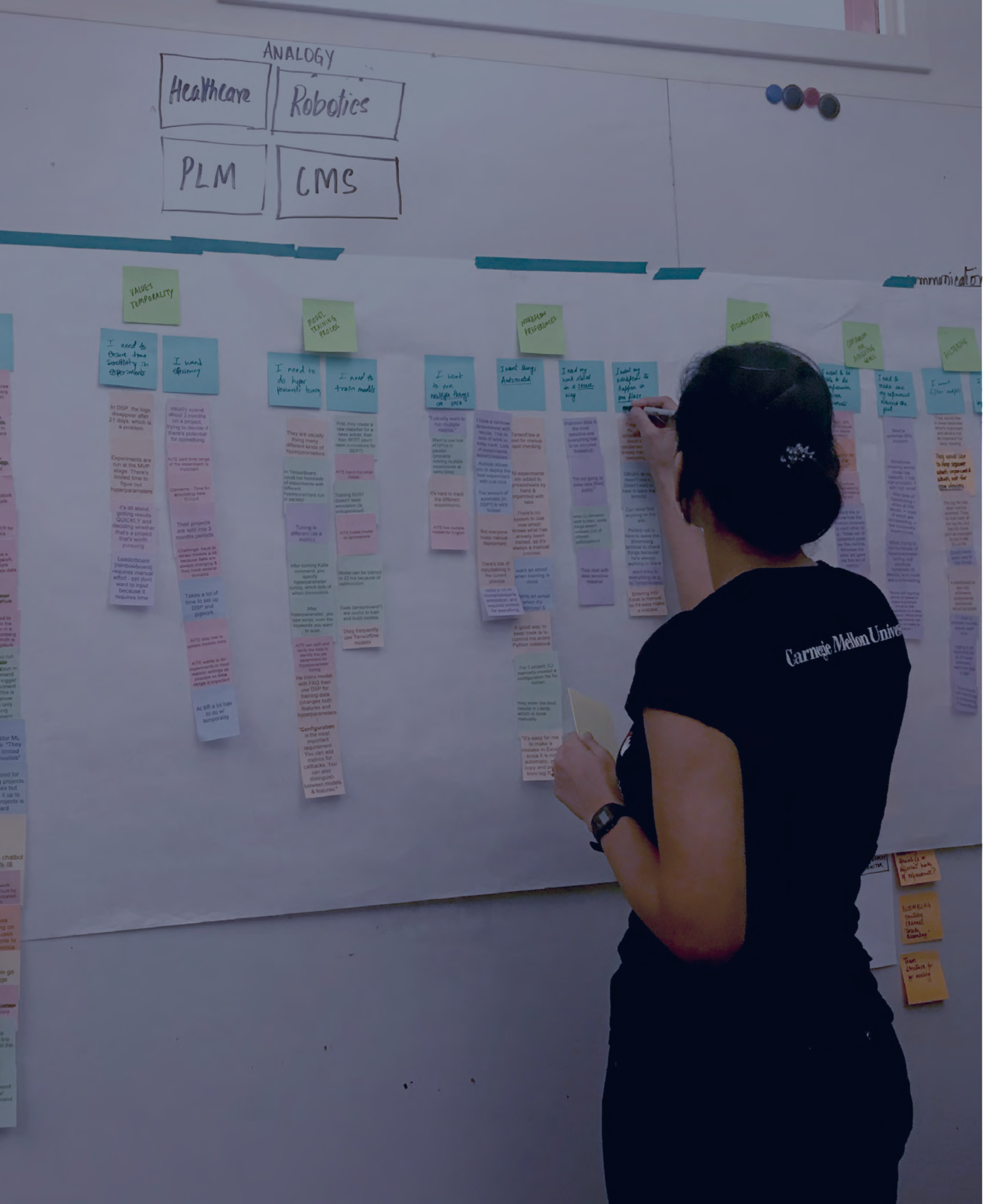
SYNTHESIS

SYNTHESIS

First, in “Tracking, Documentation, Discoverability,” we examine in detail how the three components are interdependent and why tracking serves as the foundation for effective documentation and discoverability.

Next, in “Three Components of Tracking,” we take an in-depth look at tracking and pinpoint the current challenges in each stage, along with the insights and their supporting evidence.

Lastly, in “Workarounds as Substitutes,” we present how ML engineers currently deal with the challenges of tracking, how their workarounds will inform our designs, and the insights along with their supporting evidence



SYNTHESIS

01 Tracking, Discoverability, Documentation

02 Three Components of Tracking

03 Workarounds as Substitutes

Insight 01

Tracking, Discoverability, Documentation

Ineffective tracking leads to further issues in documentation and discoverability.

Insight 02

Three Components of Tracking

The machine learning workflow is comprised of three interdependent components such as data, code, and results, which are all reliant on effective tracking.

Insight 03

Workarounds as Substitutes

Because of system limitations, machine learning engineers resort to developing their own workarounds to substitute workflow challenges.

Insight 01

Ineffective tracking leads to further issues in documentation and discoverability.

▶ Through affinity diagramming and workflow analysis, we've found that limitations of the current tools for tracking have negative, cascading effects on various aspects of ML engineers' workflows– namely documentation and discoverability.

Tracking is an extremely difficult task for two main reasons:

i It's heavily dependent on manual work.

ii It needs to take into account the iterative nature of machine learning.

After delving into the issues of tracking, we will further examine how it subsequently influences documentation and discoverability.

Challenges in Tracking

i **It's heavily dependent upon manual work.**

The engineers who we interviewed either manually produced scripts to scrape the information about experiments from logs, or manually copy/pasted information from logs into another artifact.

Because tracking is manual, it forces ML engineers to selectvely choose which experiments to track. Tracking every single experiment manually is impossible due to the sheer number of experiment jobs engineers run, and the long list of variables needs to be tracked to make an experiment reproducible. This cherry-picking strategy often leads to good experiments going undocumented.

ii **It needs to take into account the iterative nature of machine learning.**

This manual process is even harder in the context of machine learning because a core characteristic of machine learning is experimentation. ML engineers iterate on an experiment numerous times– trying out different datasets, models, and hyperparameters– to see which variation yields the best results. Yet they don't know which experiments will go into production (hence, which experiments they should track) until they've tried out other variations.

Effects of Tracking on Documentation

Tracking breeds inconsistencies in documentation.

Since there's no standardized way to track experiments, ML engineers are left to their own devices when deciding which artifacts to use for tracking, and subsequently, the devices used for creating documentation. Through interviews, we uncovered six different artifacts used for documentation: Github Issues, Team<Go>, Spreadsheets, Google Docs, Tutti, and Jupyter. This inconsistency in artifacts is not only prevalent across teams, but within teams as well.

The power of documentation.

An experiment often results in a finding or an insight that fuels a sequential decision on how to change it to yield better results. Documenting the intent behind an experiment is crucial because it's a way to document one's thought process. This is integral in allowing the ML engineer and others to revisit their work and understand at a glance why a specific decision was made.

Documentation is not merely used for internal purposes; it is also a crucial component for reporting progress to management, and it serves as a point of persuasion to resolve any differing expectations between parties. It holds the power to serve as strong supporting evidence to gain consensus from management and buy-ins from external stakeholders.

Effects of Tracking on Discoverability

ii **Inconsistencies in the tools used for documentation hinder the possibility of discoverbility.**

The current methods available for ML engineers at Bloombert to discover other teams' work include:

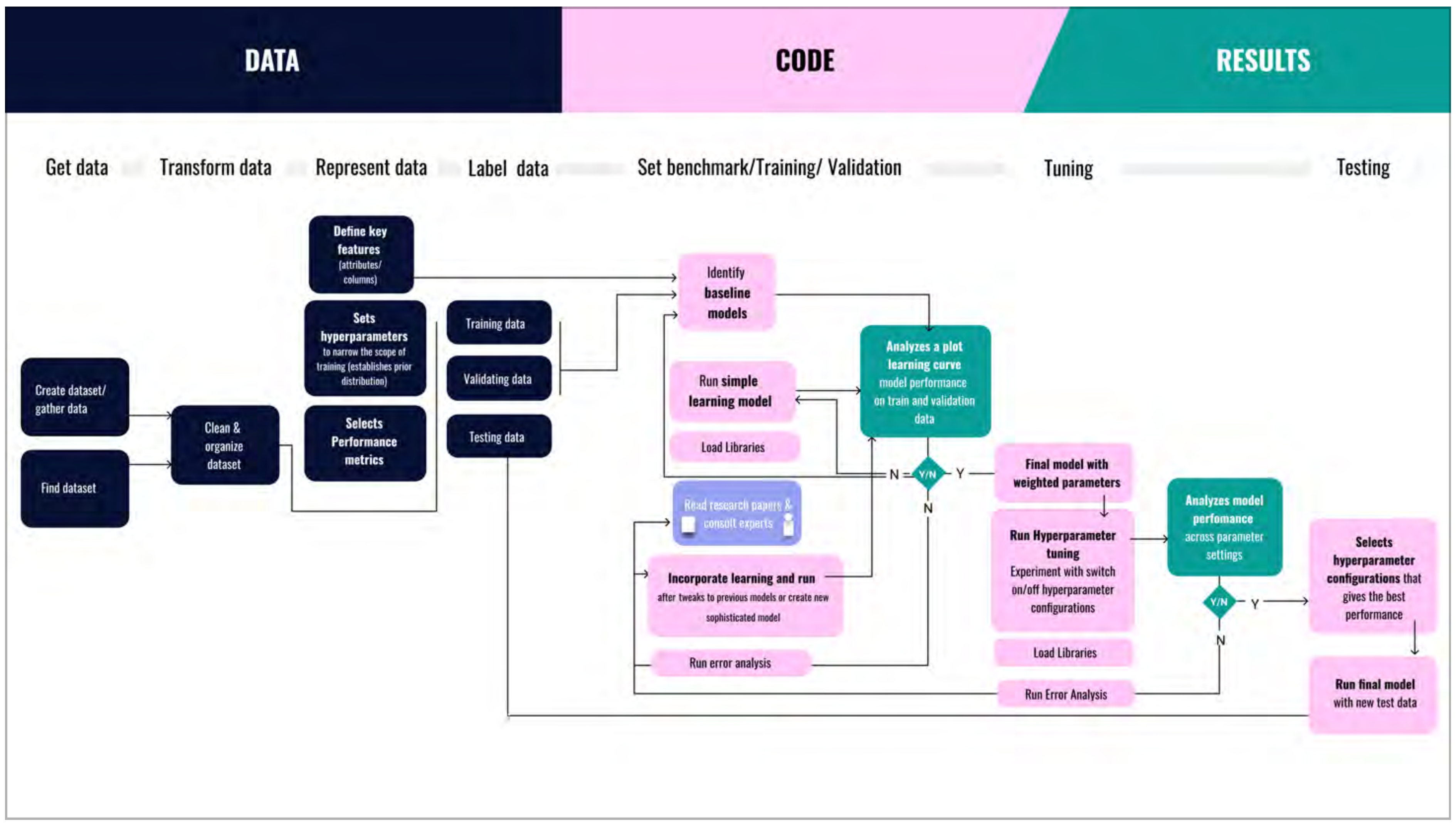
a Spring reviews
b Information sharing sessions
c Asking around
d Scheduling meetings

Because documentation is scattered in different locations, the possibility of having a more structured platform to discover others' work is prevented. Essentially, the way information is currently shared is mainly dependent on verbal means. Many ML engineers acknowledged that relevant experiments from other teams may have bypassed their attention, and this may not have occurred had they been able to discover other teams' work.

ii **The power of discoverability.**

If there were a centralized location in place for discoverability, it would decrease the possibility of duplicated work and open up immense opportunities for collaboration.

Increasing discoverability could allow engineers to collectively push the boundaries of the machine learning field and develop innovative solutions that could best serve Bloomberg and its clients.



Insight 02

The machine learning workflow is comprised of three interdependent components– data, code, and results– which are all reliant on effective tracking.

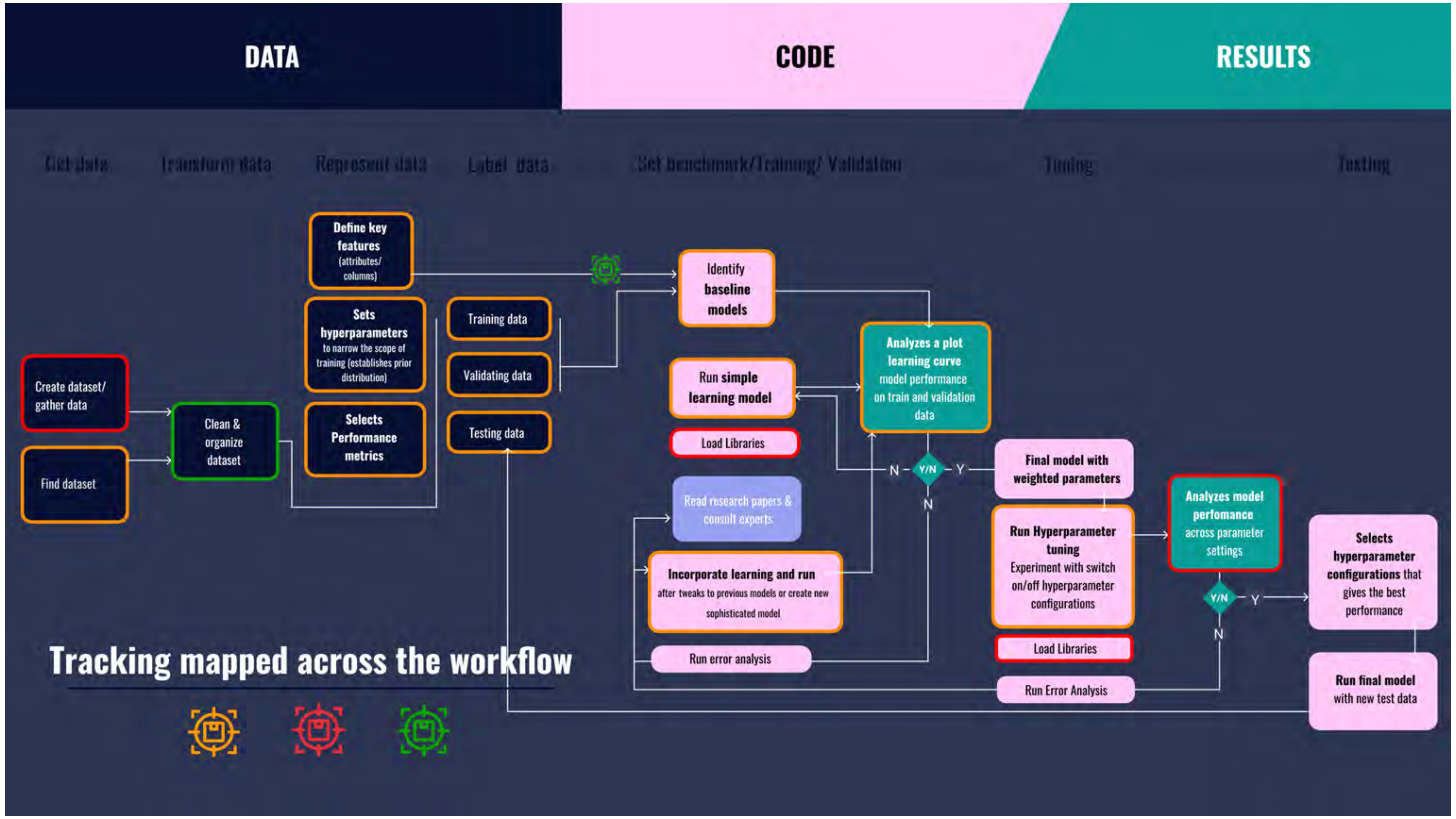
▶ We synthesized our interviews with the machine learning engineers to draft a model of their workflow. The design of this model is categorized into three distinct stages based on the three artifacts that machine learning engineers find most challenging.

After synthesis, we broke down each component into sub-categories with additional insights.

i Insights from Data

ii Insights from Code

iii Insights from Results



Workflow Introduction

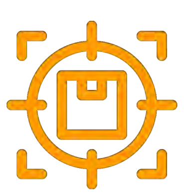
▶ In order to understand the location of tracking-related challenges in our workflow, we further codified tracking using three kinds of trackers.

Yellow: places where information is tracked, but incompletely due to the current practices.

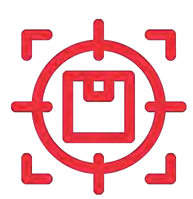
Red: places where tracking is tedious and might not be possible.

Green: places where tracking could generate new data points.

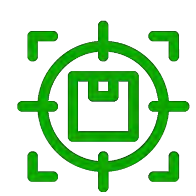
Mapping our trackers across the machine learning workflows helped us identify places where new tracking opportunities could complement current tracking practices.



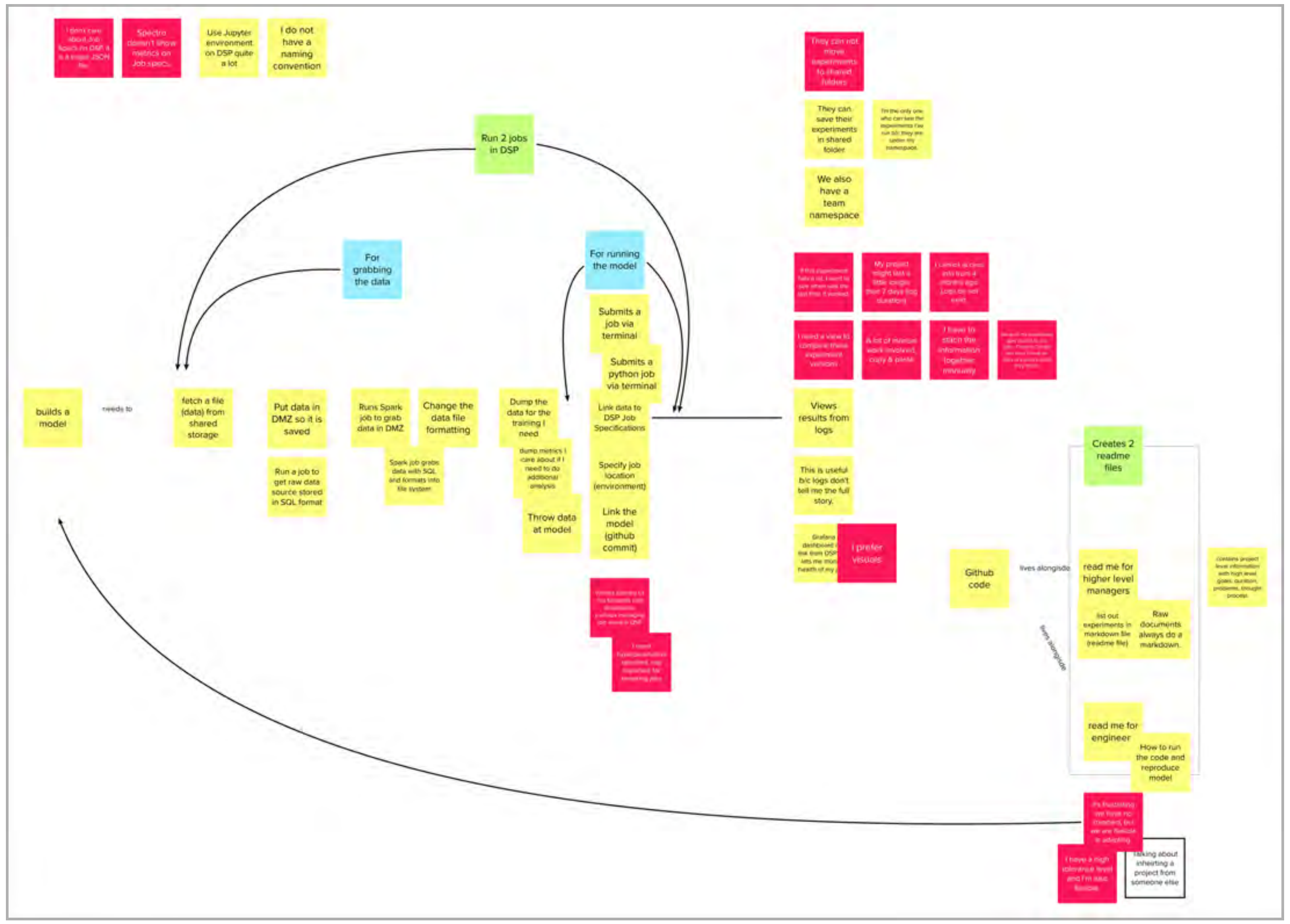
Incomplete Tracking



No Tracking



New tracking opportunity



Sequence Model Method

- We needed to develop a deeper understanding of the ML engineers' workflows and mental models, so we started exploring participatory design methods to supplement our research.

We landed on visual storytelling in an effort to create a more tangible, shared understanding of our users' processes for machine learning. Using an online tool called Mural, we created a template for our remote interviews, which allowed users to run through the workflow of a recent experiment they were working on while the interviewer captured and reflected feedback in real-time. With provided sticky notes and emojis, they could add various tools and software at different stages to narrate their own workflow story.

We found that the ML engineers were much more engaged when actively helping us design their narratives, which led to a better-shared understanding of their challenges.

Insights from **Data**

01

ML engineers could significantly decrease the amount of time pinpointing why an experiment wasn't reproducible if they were informed beforehand of alterations in the source dataset the model was trained on.

- When data is constantly being updated in a financial company such as Bloomberg, the issue is not so much tracking the data versioning, but rather tracking the obsolescence of data being used on the models.

02

Tracking the data sources, formats and systems make use of previously invested time and effort put into new experiments, allowing for efficiency.

- Some experiments require an engineer to collect data from scratch. Often times, multiple people need to be involved in order to pinpoint where the relevant dataset is. After finally getting the dataset, it's not uncommon for the data to be in the wrong format or need additional information. It may even be irrelevant.

  Tracking the location of the dataset, as well as its format and sources, could serve as a common resource, wherein everyone could benefit from prior work done to gather the dataset.

03

If a model performs well, the ability to identify its dataset filters and reuse them for future dataset cleaning could save ML engineers time and effort.

- Engineers use libraries to efficiently filter outliers. This is a critical decision-making process that identifies which information is relevant for the purpose of the project. Tracking these filters might help to know which filters provided better results on a specific dataset. These filters could then be shared with others in an effort to reproduce the same experiments.

Insights from **Code**

01

The need for an engineer to remember details dispersed across time leads to inconsistent tracking and documentation.

- Since engineers are using software systems, some details like the time stamp and username are automatically captured.

  However, important details such as metrics and hyperparameters require constant attention at different points throughout the workflow.

  This creates the possibility for important tracking details to be missed.

02

Tracking libraries is beyond an engineer's control; however, tracking updates could help choose versions better fit for the model.

- Engineers extensively use the predefined functions in many open source Python libraries. Since these are external elements beyond an engineer's scope of control, tracking can become very difficult here.

  However, being aware of library versions used and availability of newer versions gives the ML engineers an opportunity to choose a version that best serves their model.

02

Tracking knowledge from previous works' success and failures would inform future model selections in similar domain projects.

- Bloomberg has more than 250 engineers who are trying new ideas and learning from their own experiments.

  If there was a way to track which models performed well within a topic domain and the ability to search for these models, it would benefit other ML engineers working on the same topic domain. They would then be able to save time through the iterations by having predefined models as a basis to start with.

Insights from Results

01

Collating results from scattered platforms may enable engineers to visualize and compare results in multiple ways.

- Engineers currently choose their tools based on the skills and habits that they've previously formed. Therefore, it may be challenging to make someone use a new platform.

  But it is important to consider that using multiple platforms makes it difficult to share results and might possibly lead to different conclusions about the results.

02

Experiments without logic can't provide any insight into why an experiment should be tracked.

- During the hypertuning stage, engineers manage results in large volumes. Hence they often need to make educated guesses on which experiments should be tracked.

  Tracking the logic and rationale behind these guesses can provide evidence to support decisions over time.

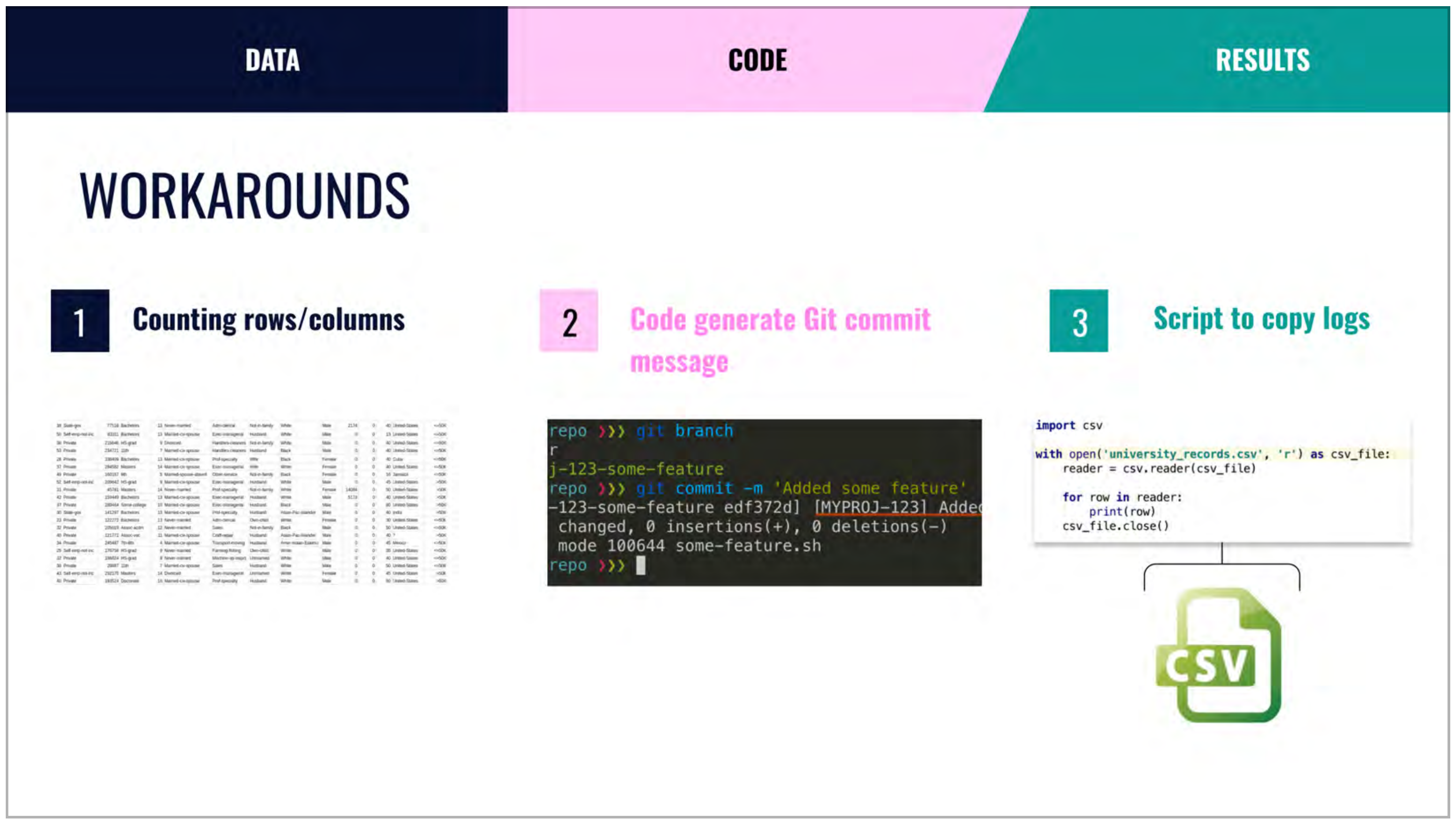
03

Tracking the amount of time it takes for an experiment can also inform future investments.

- All results are geared towards a specific business goal. The same can be said with machine learning experimentation as well.

  Over time, an understanding of the time cost along with model performance can illuminate a model's real value.

  Understanding the model value, which is the time invested vs. the business gain it produced, could be a determinant of an experiment's success. This could also inform future decisions for building models intelligently.

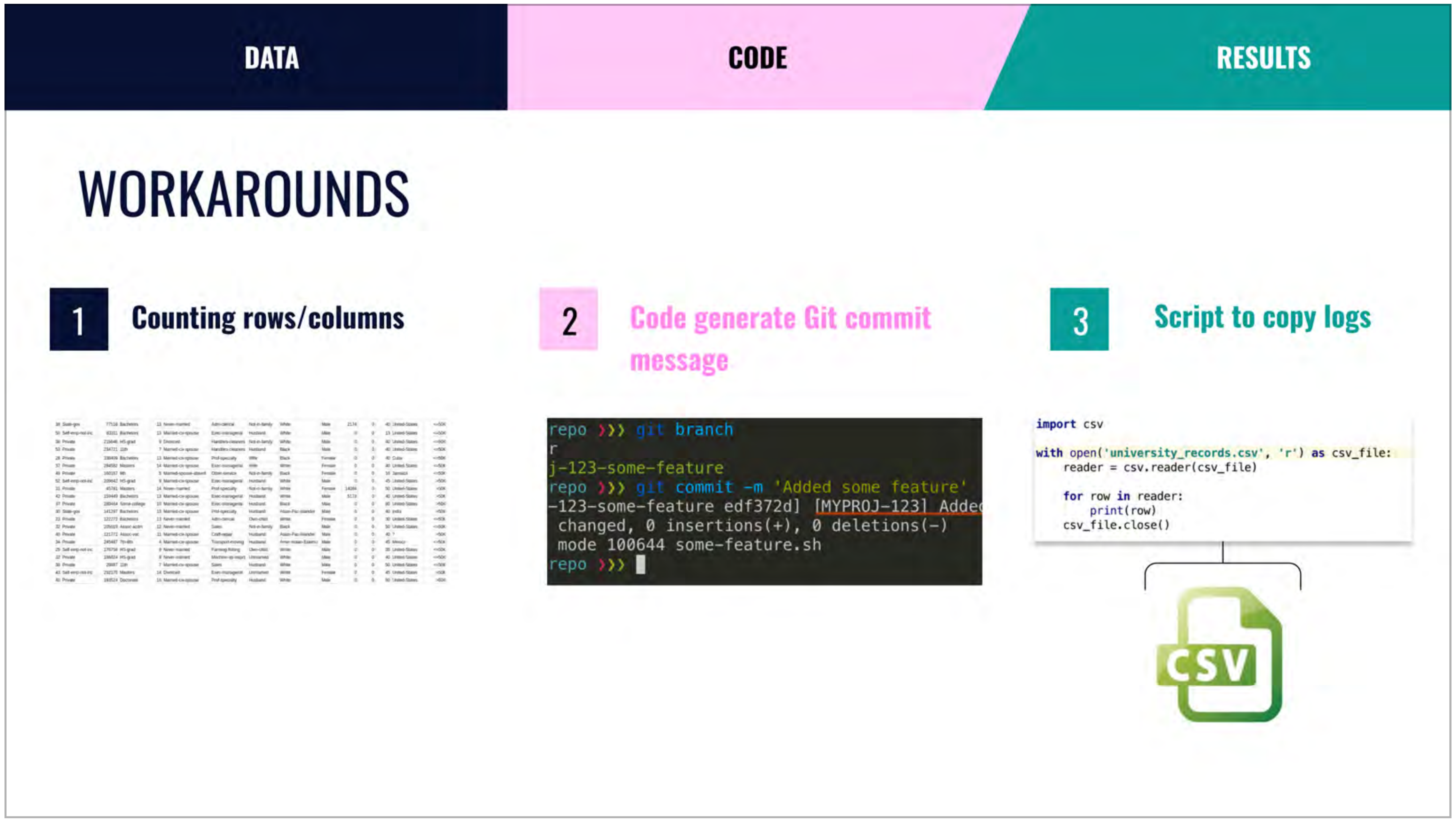


Insight 03

Because of system limitations, machine learning engineers resort to developing their own workarounds to substitute workflow challenges.

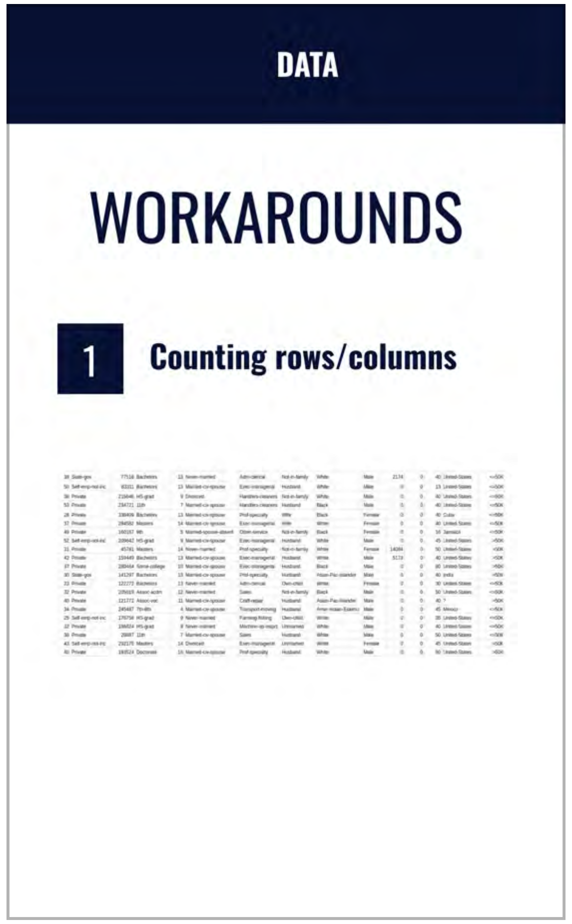
▶ As we continued to collaborate on our workflow with more and more engineers, a significant pattern emerged. Since they generally lack a set standard, ML engineers have resorted to developing their own workarounds, making it difficult to collaborate across teams. Yet workarounds have become so internalized at Bloomberg that they almost seem like an inherent part of the process.

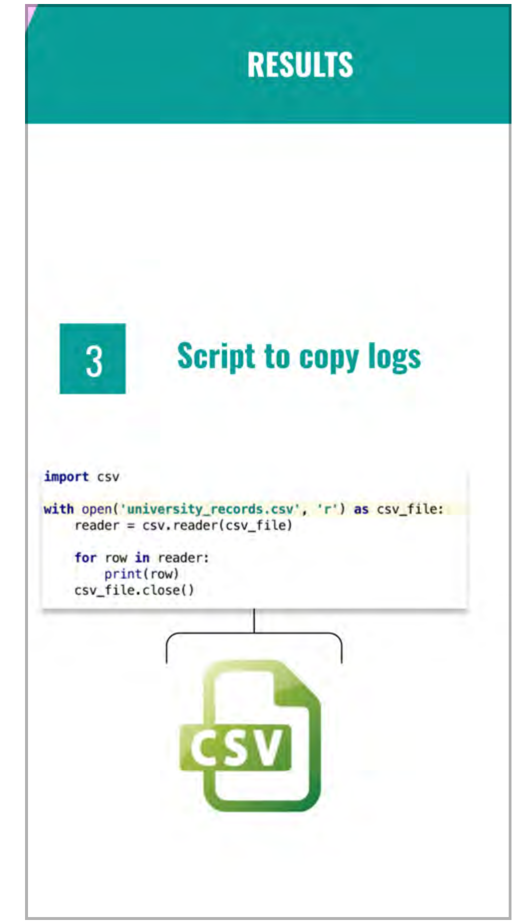
We started noting where these workarounds were happening and identified which were the most prevalent within in the three phases of tracking: data, code, and results.



Workaround Categorization

- Using the model which we'd created from synthesized interviews with machine learning engineers, we grouped the workarounds into each of the three phases: data, code, results. From here, the purpose of the workarounds became more evident, and we could start correlating them with potential designs.





Data

We discovered that in the data phase, there are numerous workarounds Bloomberg engineers use to track dataset versions.

One of the most consistent is manually **tracking the attributes of a dataset by counting rows and columns.** This allows the engineer to know if there have been any changes made. Albeit labor intensive, this method gets the job done.

Code

In the coding phase, one of the most common workarounds we found was using code to automatically generate a ready-to-use commit message.

From one of the engineers interviewed, we learned that this not only allows him to track metadata efficiently, but also serves as a reminder to commit the code. In every run, he captures the meta-data with a commit message. These **commit messages then allow him to see the evolution of the experiment** in terms of changes made.

Results

Lastly, in the results phase, we observed engineers **using Python code to automatically move outputs into a csv file.**

As a workaround, this method is prone to fewer errors, and if incorporated into our design, could help automate logging of results in a centralized place, offering efficiency and reducing cognitive load for engineers.

DESIGN

DESIGN

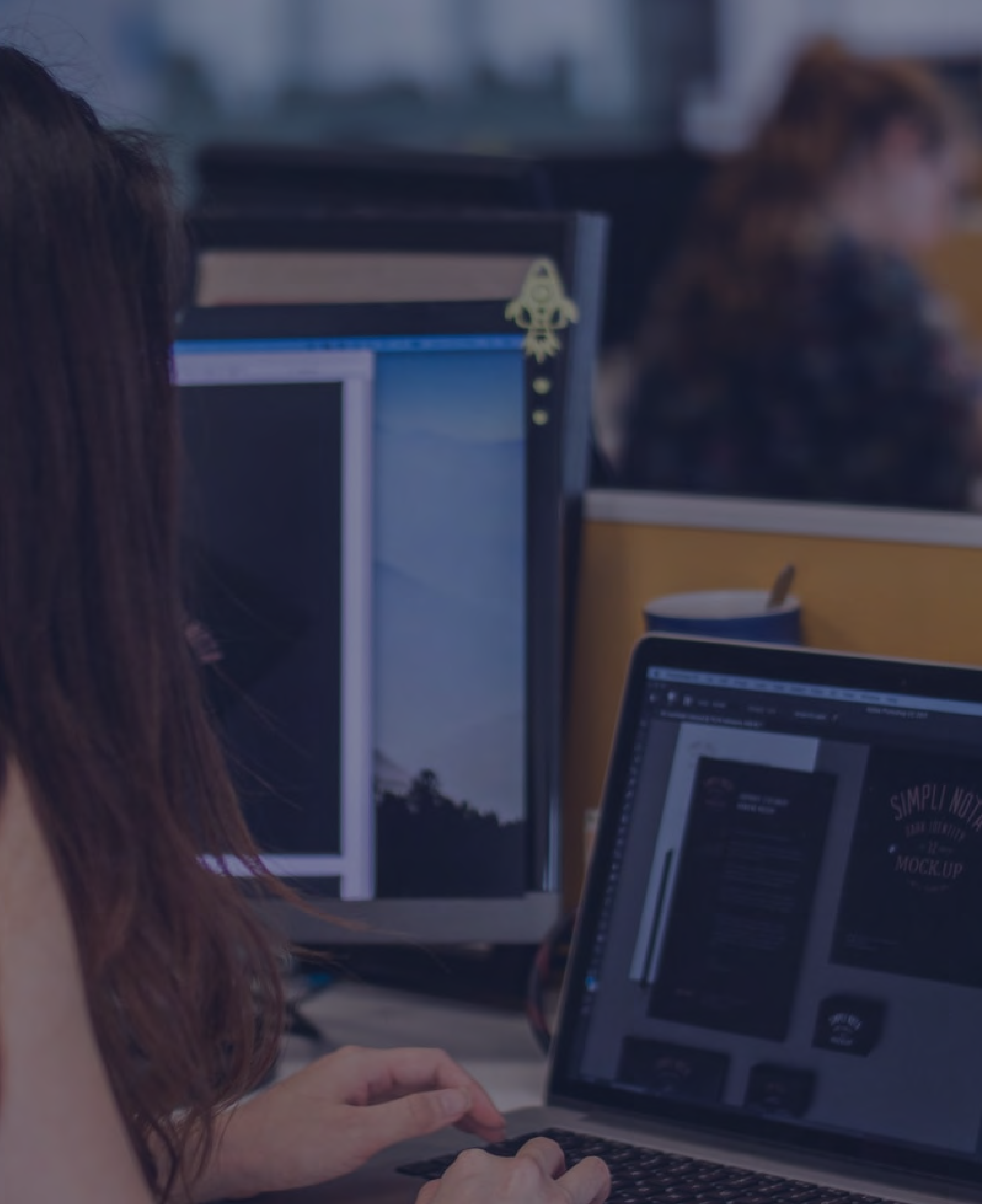
Based on our research findings and insights, we generated storyboards and low-fidelity mockups to visualize what the solutions could look like. In addition, designing in parallel helps us to evaluate our design assumptions as we move forward in our research.

Here in this section we will present our three approaches and the evolution of prototypes based on the findings discovered.

They are:

- i Storyboards
- ii Conceptual Pretotype
- iii Dashboard Prototype

We will also discuss our process in arriving at our designs.



DESIGN

01 Storyboards

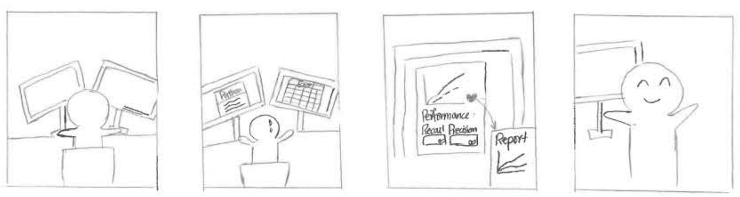
During our initial research phase, we individually synthesized visions and concepts for making the ML experiment management process simple. To outline some of those initial directions, we created several storyboards which would be used for speed dating.

Our three storyboard ideas included:

- i How might we help ML engineers keep a record of their experiment results?
- ii How might we increase transparency and collaboration for ML experiments?
- iii How might we make it easier to capture a ML experiment and reuse/reproduce it?

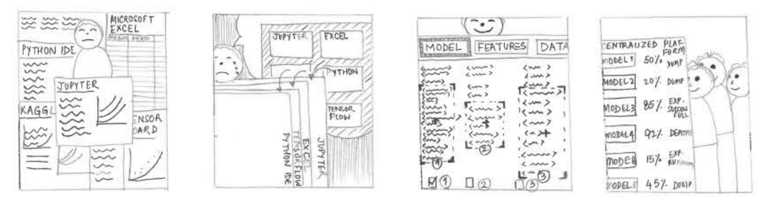
01 Storyboards Overview

i How might we help ML engineers record their experiment results?



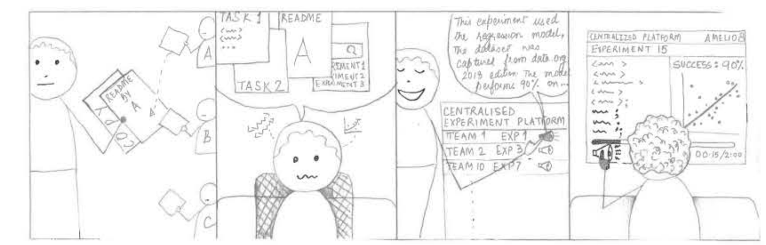
ML engineers at Bloomberg often make errors while copying/pasting different aspects of an experiment to generate their reports. Our design harnesses a like/save feature across all platforms which selectively allows the engineers to identify elements that are worth cross-referencing later on. The identified "liked" elements could also be used to quickly pull together a report.

ii How might we improve transparency and collaboration for ML experiments?



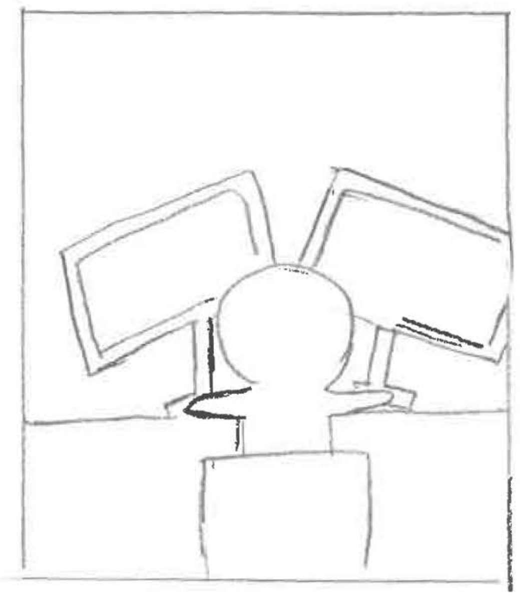
Since the ML engineers at Bloomberg use multiple softwares to save, compile and visualize datasets and models, a screen-based toolbar that captures information across all platforms could allow users to choose their software. This would enable easy searchability in future.

iii How might we make it easier to reproduce an ML experiment?

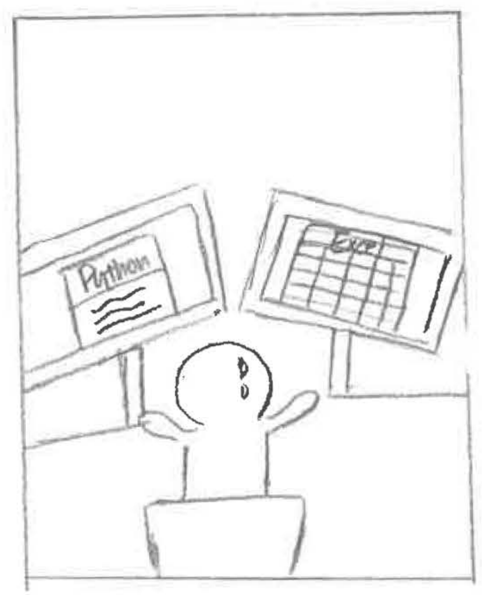


Task switching bears a cognitive load on users. This solution seeks to validate the use of voice as a modality to capture the report summary along the way. Auto-translation of the audio file could create a readme and have placeholders to attach visualizations and experiment results.

How might we help ML engineers keep a record of their experiment results?



Justin is a ML engineer who works at a big tech company.

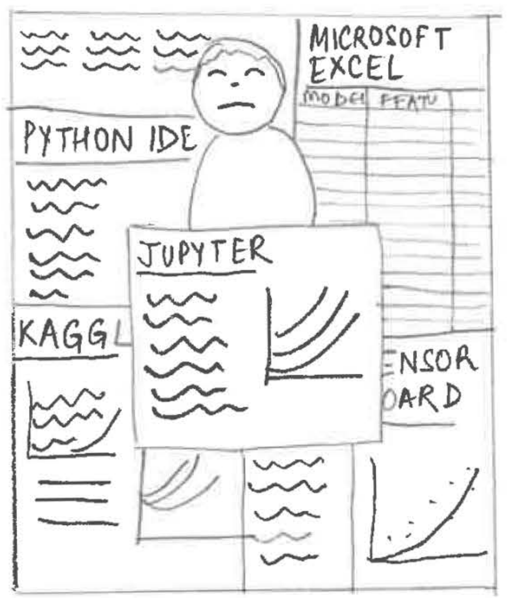


Every time Justin needs to record the performance of a model, he needs to manually enter the performance metrics in an Excel sheet. This process of manually entering information is tedious, and Justin often makes mistakes.

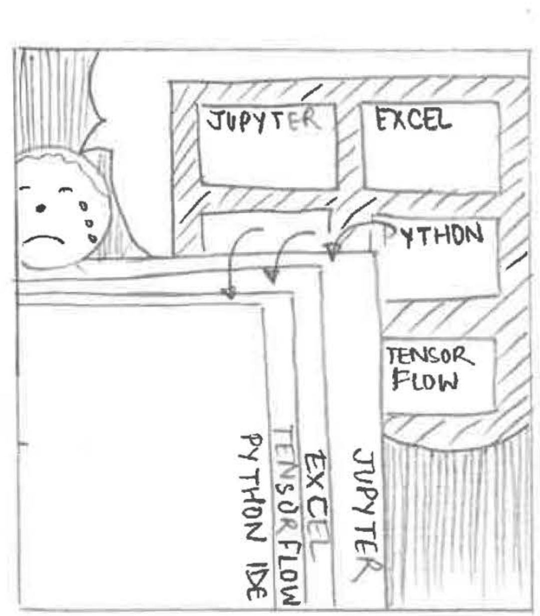
Justin then runs a model on a platform and can “like” anything he wishes to record on a report--whether that is the hyperparameter that he used, the model’s performance metrics, or etc. The information he “liked” is automatically recorded on a report.

Justin is now happy because he doesn’t have to manually record his models on an Excel sheet anymore.

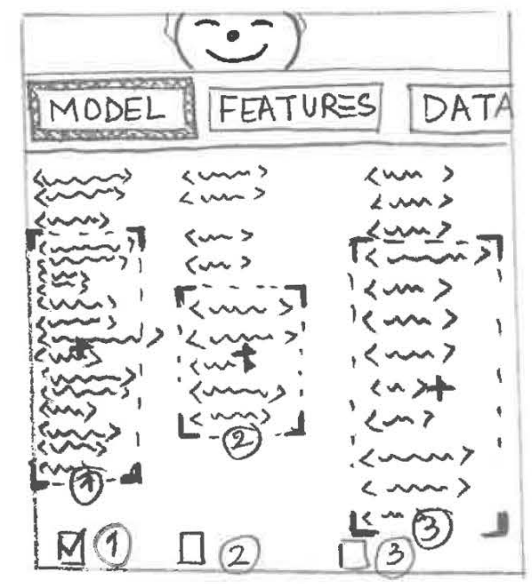
How might we increase transparency and collaboration for ML experiments?



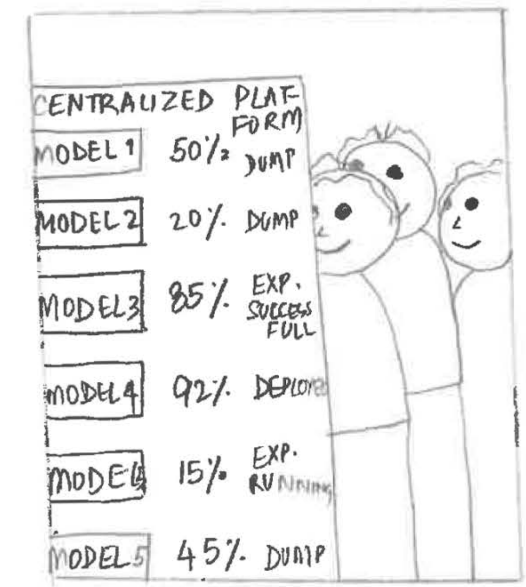
ML engineers use Python, Jupyter, Kaggle, Excel sheet, and Tensorboard for managing ML experiments.



There is no unified coding environment for ML engineers.

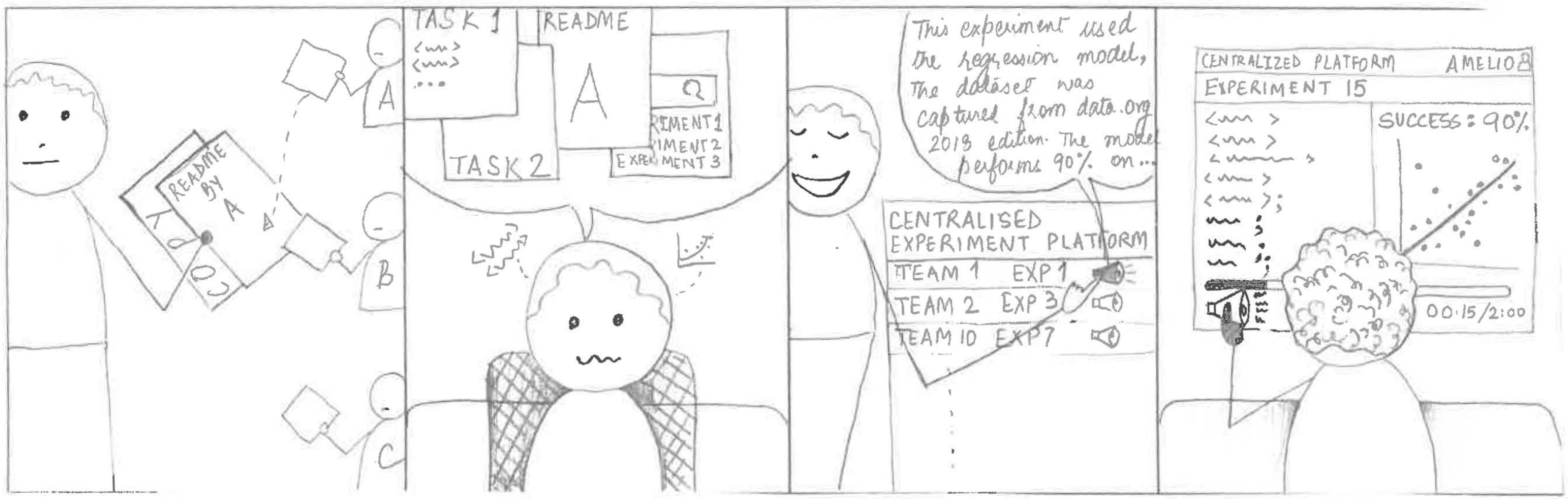


The introduction of screen capture allows for automatic report generation.



ML engineers are able to easily check the status, progress, and performance of models on one centralized platform.

How might we make it easier to capture a ML experiment and reuse/reproduce it?



Amelio wants to replicate an experiment that was previously produced by another ML engineer.

He has other things to do. He finds it difficult to digest all the complicated information on the Readme file.

He uses the new audio-friendly platform which has all the experiments categorized and allows him to playback a 2-minute podcast version of the Readme file.

He gets all the information he needs in an engaging way and feels inspired to create his own for other benefits as well. His audio file gets converted into a Readme file automatically using the NLP technology.

02 Conceptual Pretotype

A pretotype lies between "an abstract idea and proper prototype".

It is different than that of a prototype in its ability to test ideas and their market potential cheaply and easily.

Instead of answering questions such as: "can we build it?" or "will it work as expected?", a pretotype focuses on answering core need questions such as "should we build it at all?"

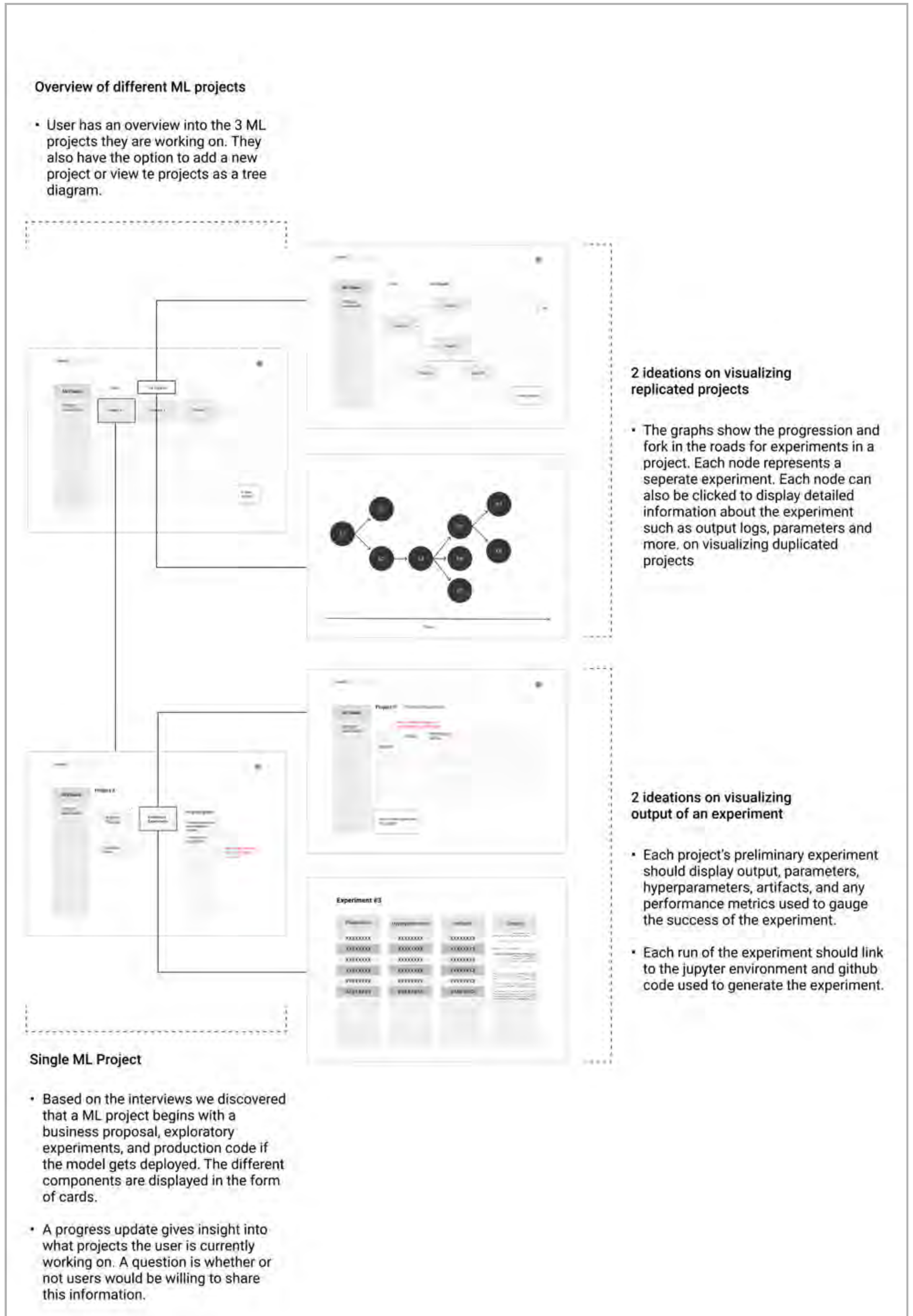
With this in mind, we approached the pretotype testing core need questions below:

i. Does a progress report give members of the team ability to view the status of a project? How do they feel about transparency into what others are working on?

ii. How do engineers feel about the ability to view duplicated experiments as nodes in a tree branch?

iii. If an output of an experiment displays all the artifacts involved, such as parameters, hyperparameters, artifacts, metrics, and results all on one page, is this information useful or not?

iv. Do users want to view all of their current projects on one page? (Revealing detailed information as each project is clicked).



We designed four components that make up the pretotype addressing our previous questions:

i **An overview of ML projects**

- Helps engineers manage their experiments as they freely move the cards to prioritize or archive certain ones

ii **Visualizing history of an experiment**

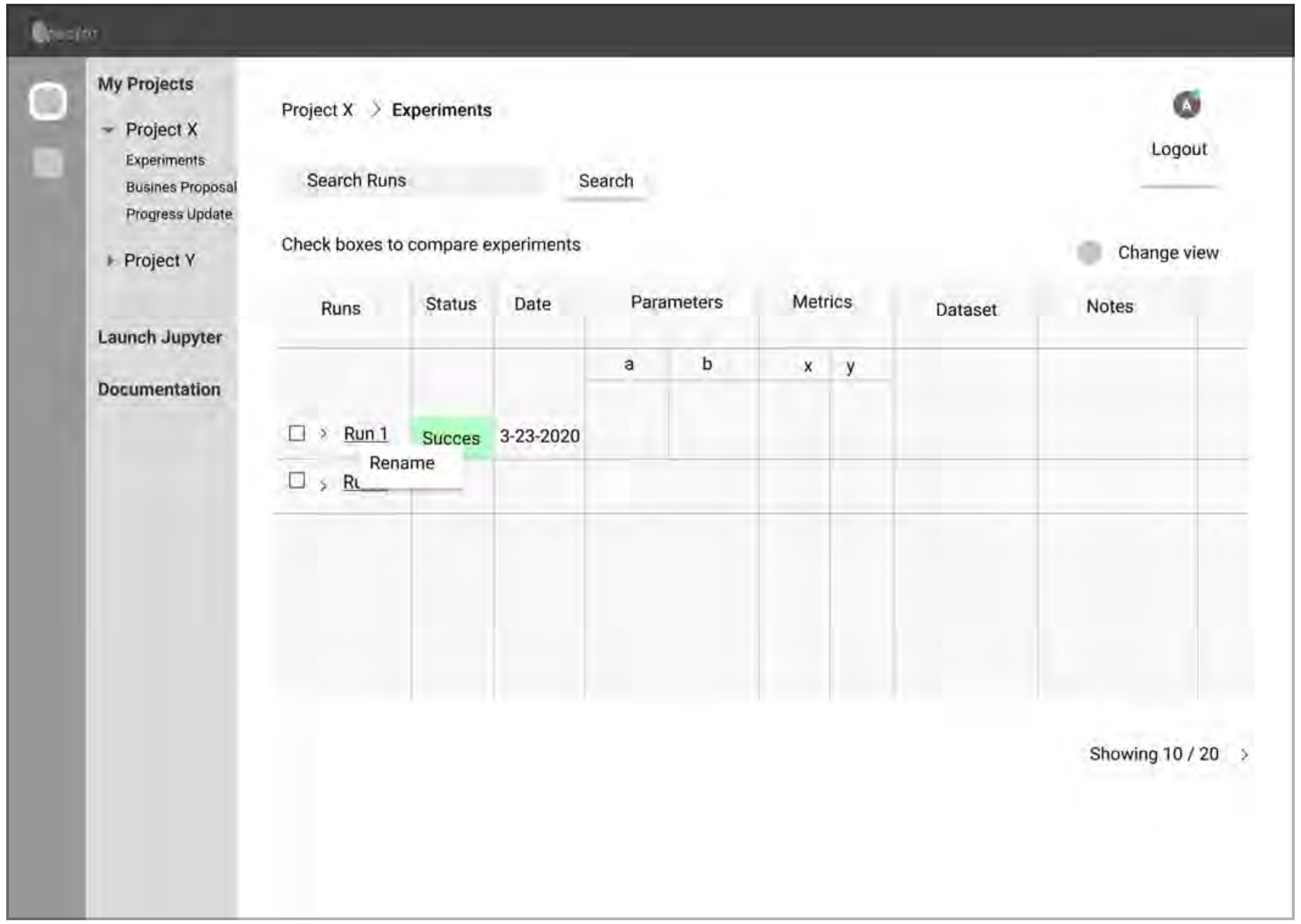
- Gives engineers the ability to track the backbone of an experiment, view its history for easy replication

iii **Visualizing output of an experiment**

- Addresses tracking issue by helping engineers record experiment results as they run one experiment

iv **Progress report into the status of a project**

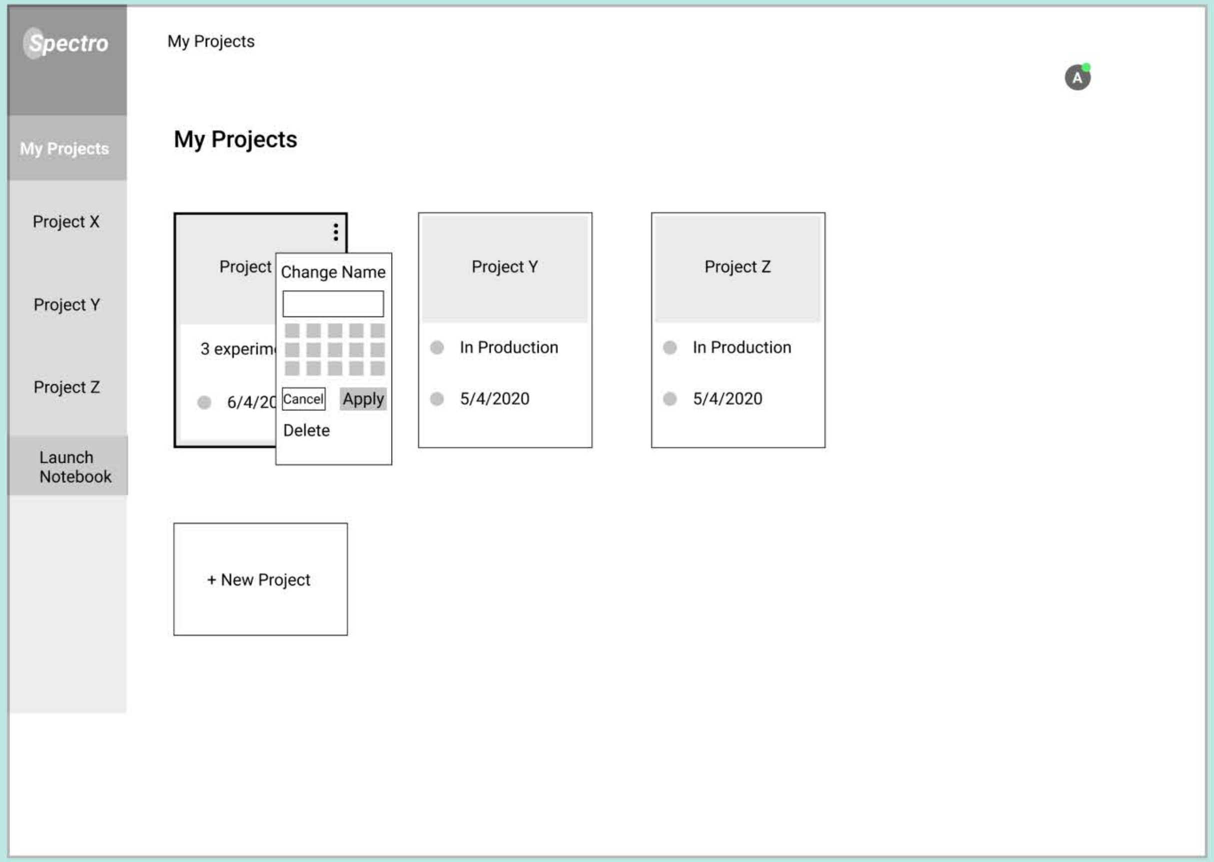
- Addresses transparency and collaboration issues in teams

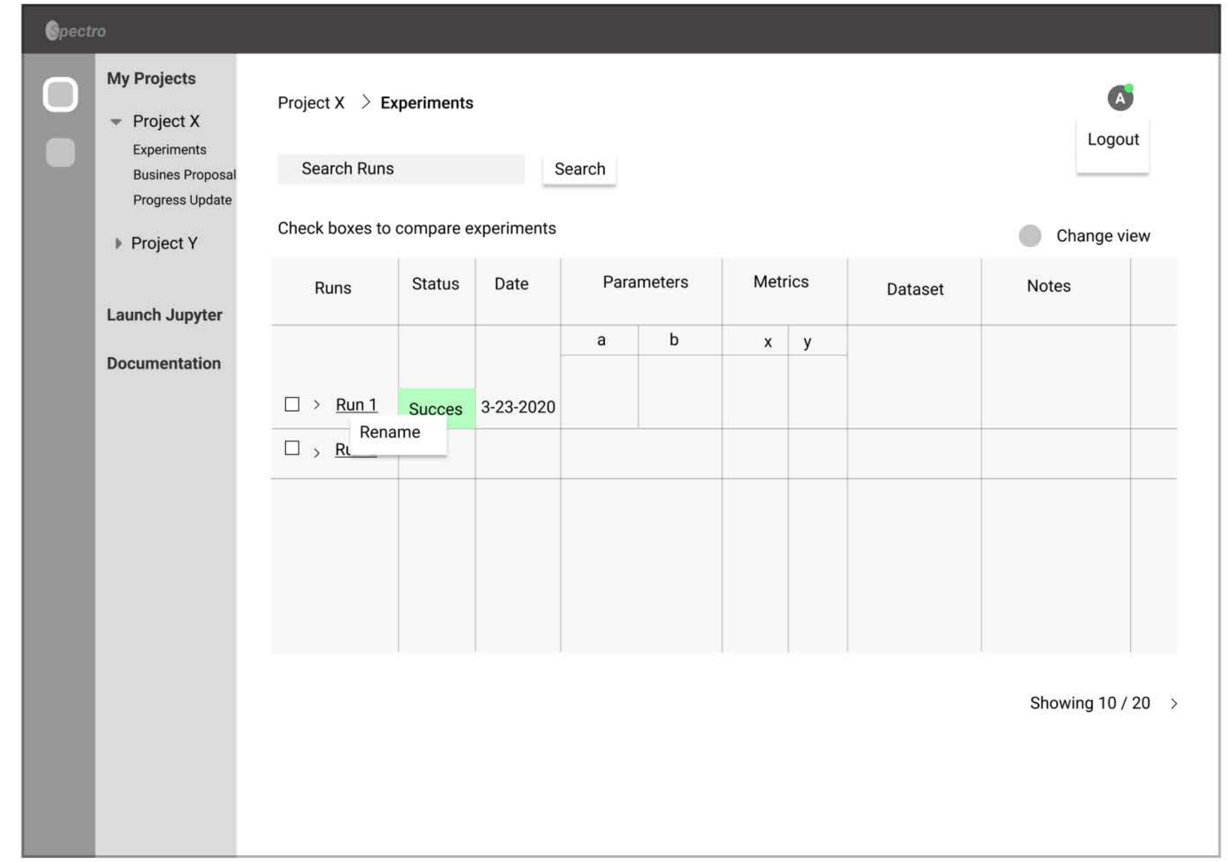


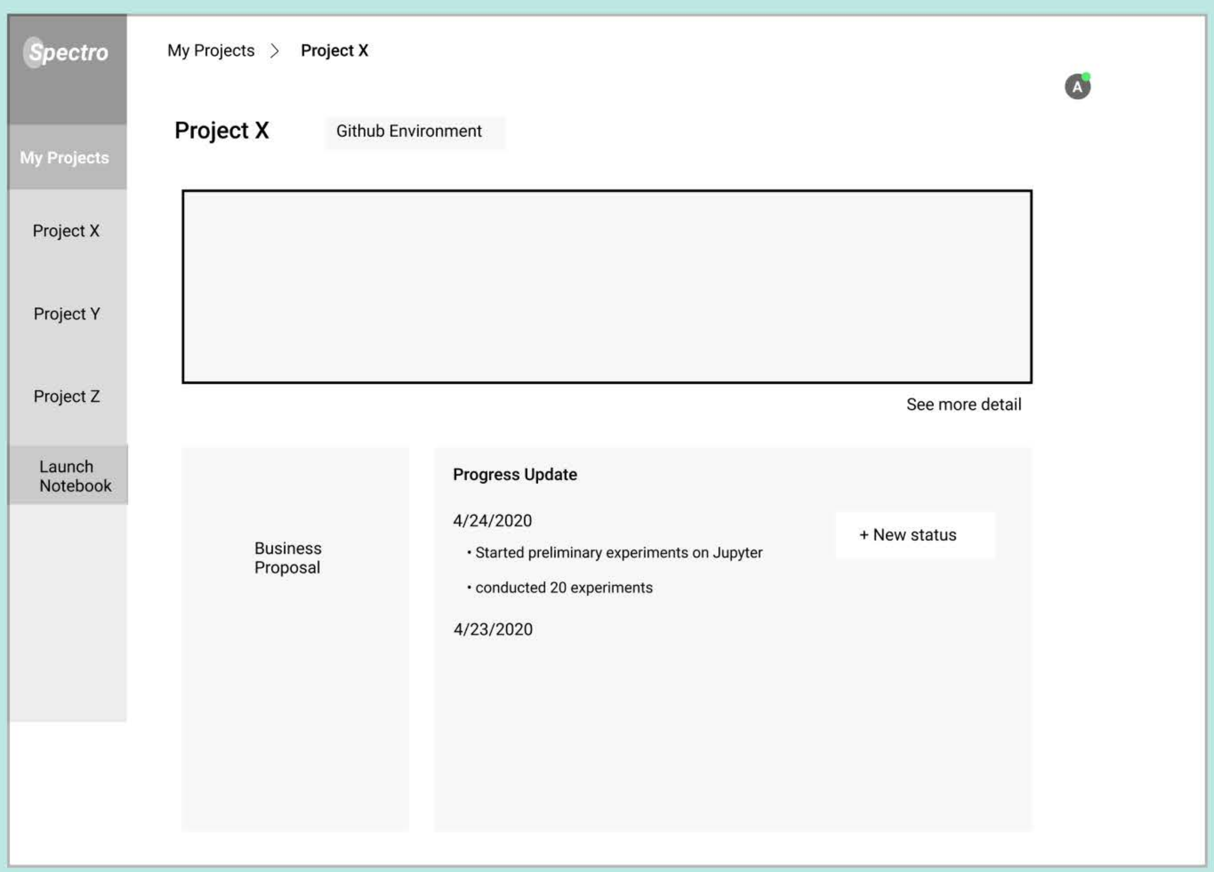
03 Dashboard Prototype

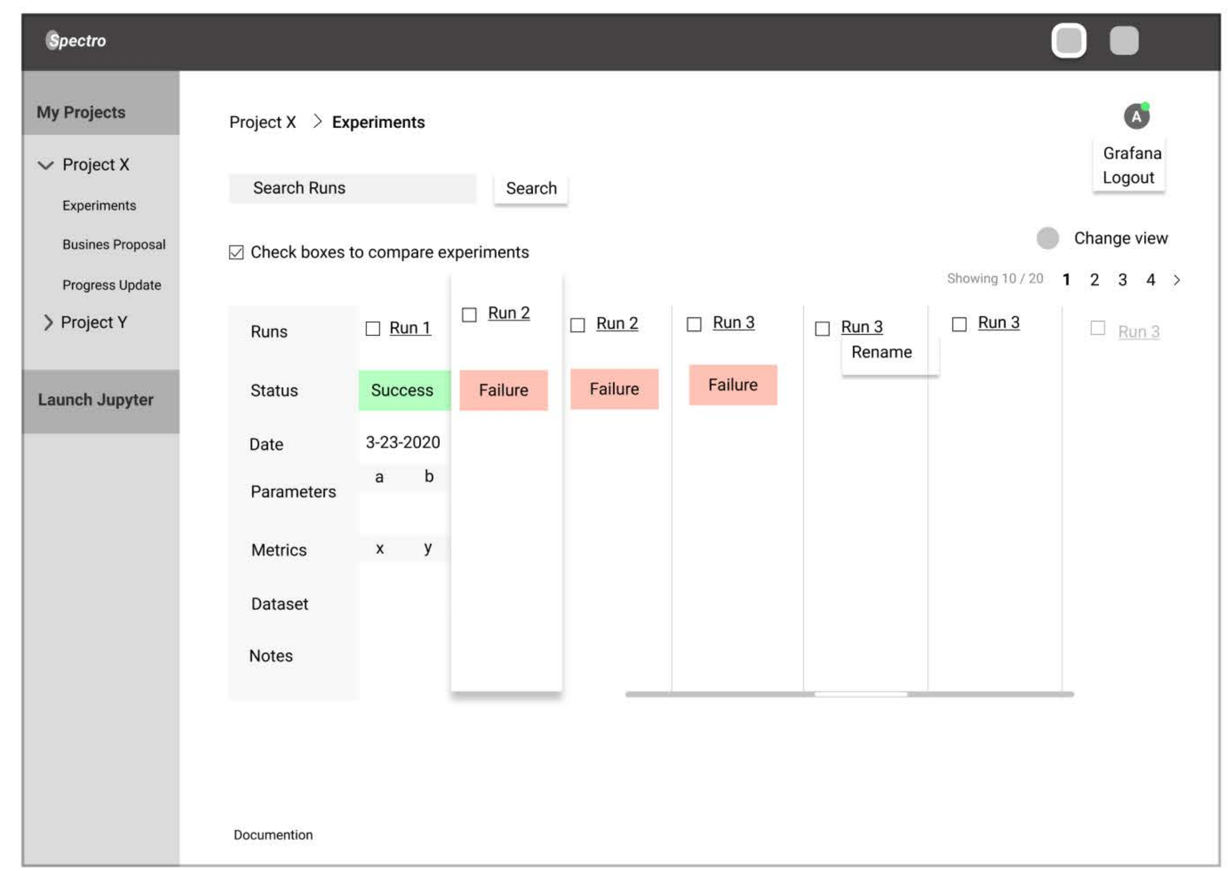
- Based on additional remote contextual inquiries collected, we generated more insights into user needs and pain points. We created mid-fidelity prototypes to further test our assumptions and use as a starting point for next semester.

Design is an iterative process needing constant validation.

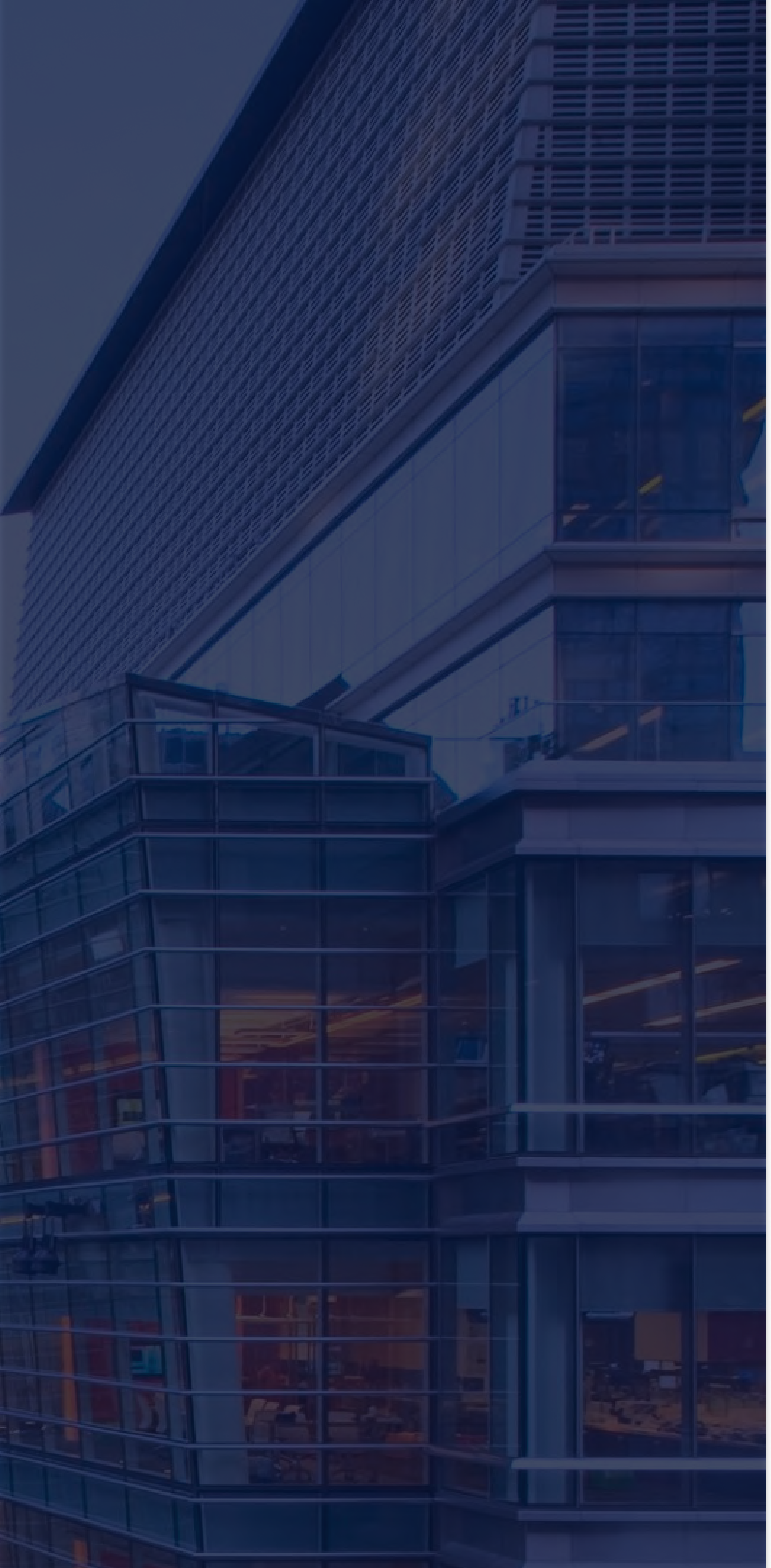

Spectro
My Projects
My Projects
My Projects
Project X
Project Y
Project Z
Launch Notebook
Project
Change Name
Cancel
Apply
Delete
3 experim
6/4/20
Project Y
In Production
5/4/2020
Project Z
In Production
5/4/2020
+ New Project


Spectro
My Projects
Project X
Experiments
Busines Proposal
Progress Update
Project Y
Launch Jupyter
Documentation
Project X
Experiments
Logout
Search Runs
Search
Check boxes to compare experiments
Change view
Runs
Status
Date
Parameters
Metrics
Dataset
Notes
a
b
x
y
Run 1
Succes
3-23-2020
Rename
Showing 10 / 20


Spectro
My Projects
Project X
My Projects
Project X
Project Y
Project Z
Launch Notebook
Project X
Github Environment
See more detail
Business Proposal
Progress Update
4/24/2020
• Started preliminary experiments on Jupyter
• conducted 20 experiments
4/23/2020
+ New status


Spectro
My Projects
Project X
Experiments
Busines Proposal
Progress Update
Project Y
Launch Jupyter
Project X
Experiments
Grafana
Logout
Search Runs
Search
Check boxes to compare experiments
Change view
Showing 10 / 20
1 2 3 4
Runs
Status
Date
Parameters
Metrics
Dataset
Notes
Run 1
Run 2
Run 2
Run 3
Run 3
Rename
Run 3
Run 3
Success
Failure
Failure
Failure
3-23-2020
a
b
x
y
Documention


SUMMER



SUMMER

The conclusion of the Spring research phase of the project has positioned us for success in the design phase. We will be using the findings, insights, and models we created to design, test and finalize prototypes.

Our final goal is to design a product that creates a seamless machine learning experiment management tool that benefits all ML engineers. We aim to use tracking as a starting point for designing opportunity spaces or resources for ML engineers.

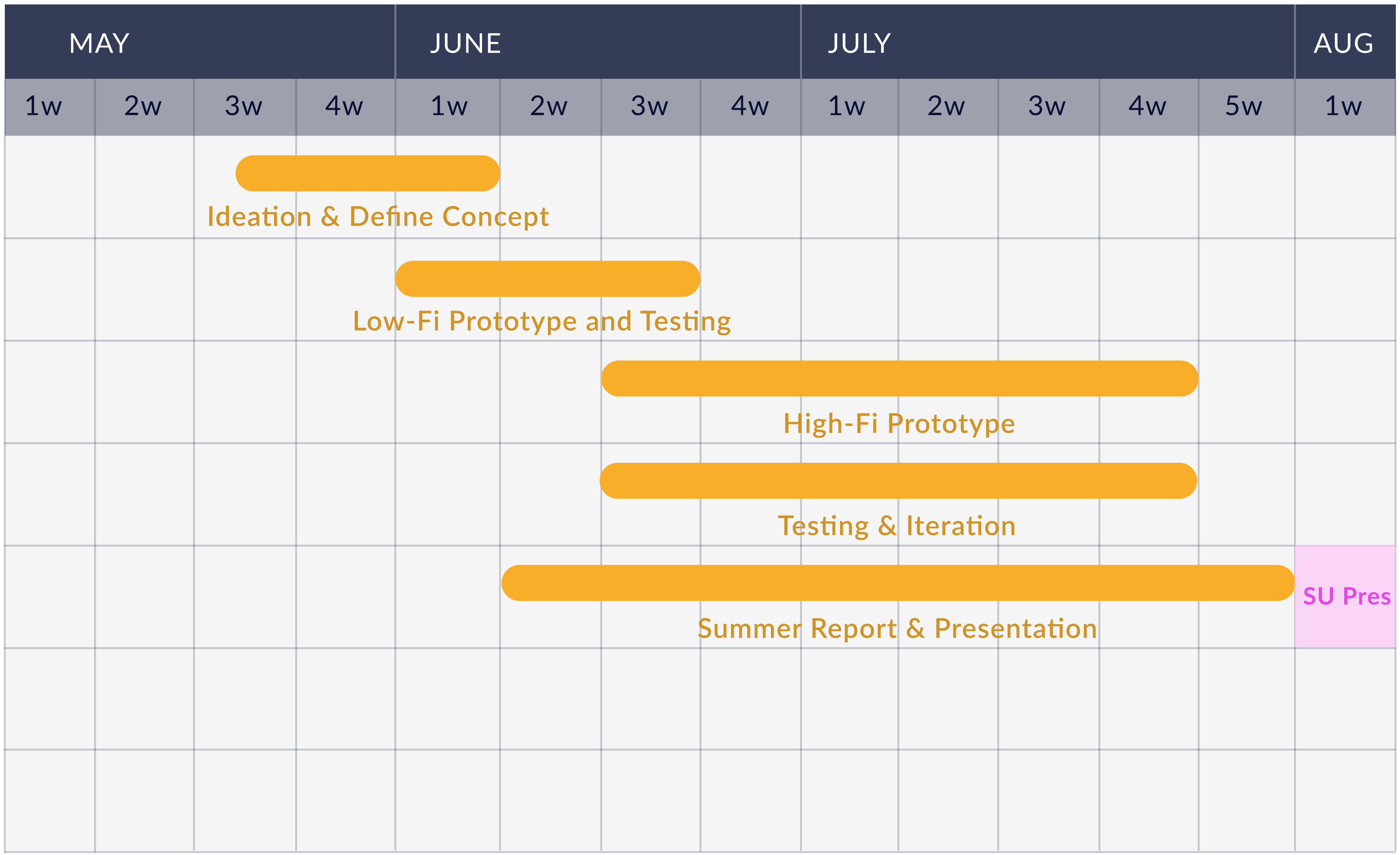
In the summer semester we will:

i. Use findings to drive new prototype designs

ii. Test assumptions made using prototypes as an artifact

iii. Iteratively work on final design

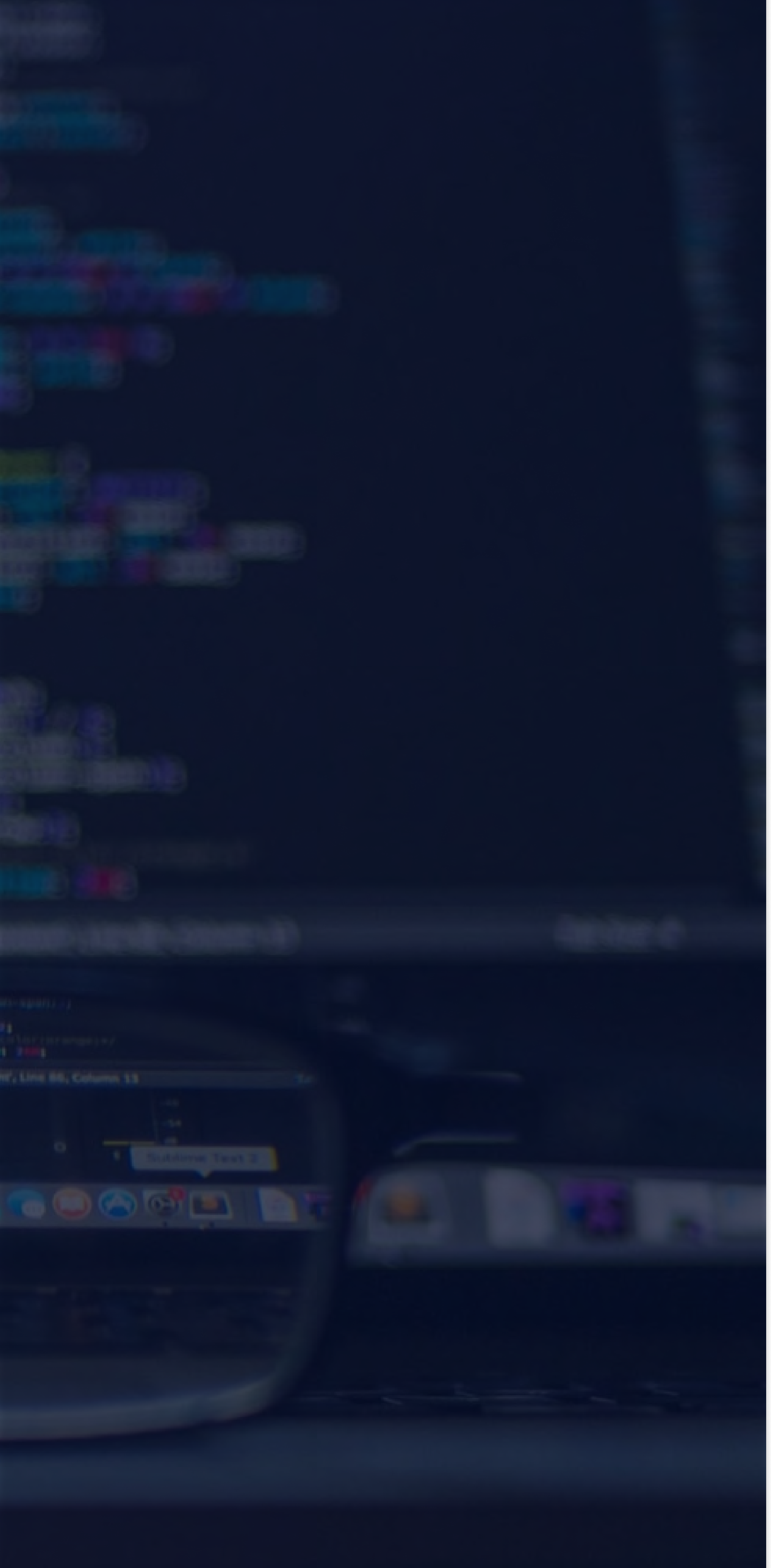
Summer Timeline



A Special Thanks to:

Ian Hummel
Monica Piper
Stephen Cook
and
Bloomberg ML engineers
and data scientists

APPENDIX



APPENDIX

i Additional Findings

- DMZ
- Experiment Comparison

ii Glossary

DMZ

Though the goal of DMZ is to ensure that the sensitive data residing there is completely secured, the way it secures its data significantly disrupts ML engineers' workflow and decreases their efficiency.

A sample from some of the ML engineers' testimonies:

i *"DMZ is terrible...I am constantly having to build my own workarounds."*

ii *"Huge barrier of entry because data is very sensitive. Takes a lot of time to decrypt the data."*

iii *"It's hard to develop, debug, or even run code because there's a hard restriction of what you can do in the environment."*

iv *"DMZ is a pain."*

Multiple ML engineers expressed how disruptive DMZ was to their workflow. Understandably, DMZ has multiple barriers and layers to ensure that sensitive data stays safe and private. However, the way that the DMZ is ensuring security is severely hindering the ML engineers from doing their work effectively.

The problem isn't ensuring security; it's how security is ensured. For instance, ML engineers need to encrypt the data in a way that makes it unsearchable. In general, it's very hard to develop, debug, or even run code in the restricted environment.

There is a need to re-examine how DMZ is structured and redesign it in a way that could guarantee both the security of the data and the satisfaction and seamlessness of ML engineers' work.

Experiment Comparison

47% of ML engineers cited that they most enjoy "comparing experiments", which by far surpasses "collaborating with other teams" (17%) and "building experiment reports" (2%).

Yet current tools available for the ML engineers make it extremely challenging for them to compare experiments. There is no streamlined way for them to aggregate and compare results. From the survey results and semi-structured interviews, it was apparent that comparing experiments and different hypertune jobs require them to take many extra manual steps.

For instance, one ML engineer stated that he needed to manually add boilerplate code to the different experiments, write them to HDFS, download them locally, and then analyze the results.

If there were a streamlined way to compare results, we could not only take the burden off of the ML engineers in setting up the environment for experiment comparison, but could also leverage what they enjoy the most— generating insights and testing out their hypotheses— the fun part!

Ultimately, we found that challenges with the DMZ disrupt the enjoyment of the experiment comparison process for ML engineers. While we believe this finding is important to address, it falls outside of our current project scope.

Glossary

BCS Bloomberg Cloud Storage, where data and models are stored.

DMZ It's an environment where restricted, sensitive data reside in. It contains various barriers of entry to ensure data stays protected.

DSP Data Science Platform, a computing platform that initially started out as a resource management platform, but has since expanded to support development efforts targeting data-driven science, machine learning, and business analytics. It's compatible with Spark, Tensorflow, and Jupyter.

Jira A project management software that's used to distribute workload across team members and track a project's progress.

Jupyter An open-source web application that allows for the use of live code, equations, visualizations, and text.

Katie It's a set of command lines that allow users to submit jobs to the DSP.

Maestro ML engineers often use it to hypertune their models in the DSP once the models are performing relatively well. A hypertune job would generate tens to hundreds of experiment runs, with each run showcasing a different configuration of hyper-parameters. Maestro supports visualizations of the performance of each experiment run.

Phantam DSP hopes to integrates it into the platform to track machine learning jobs.

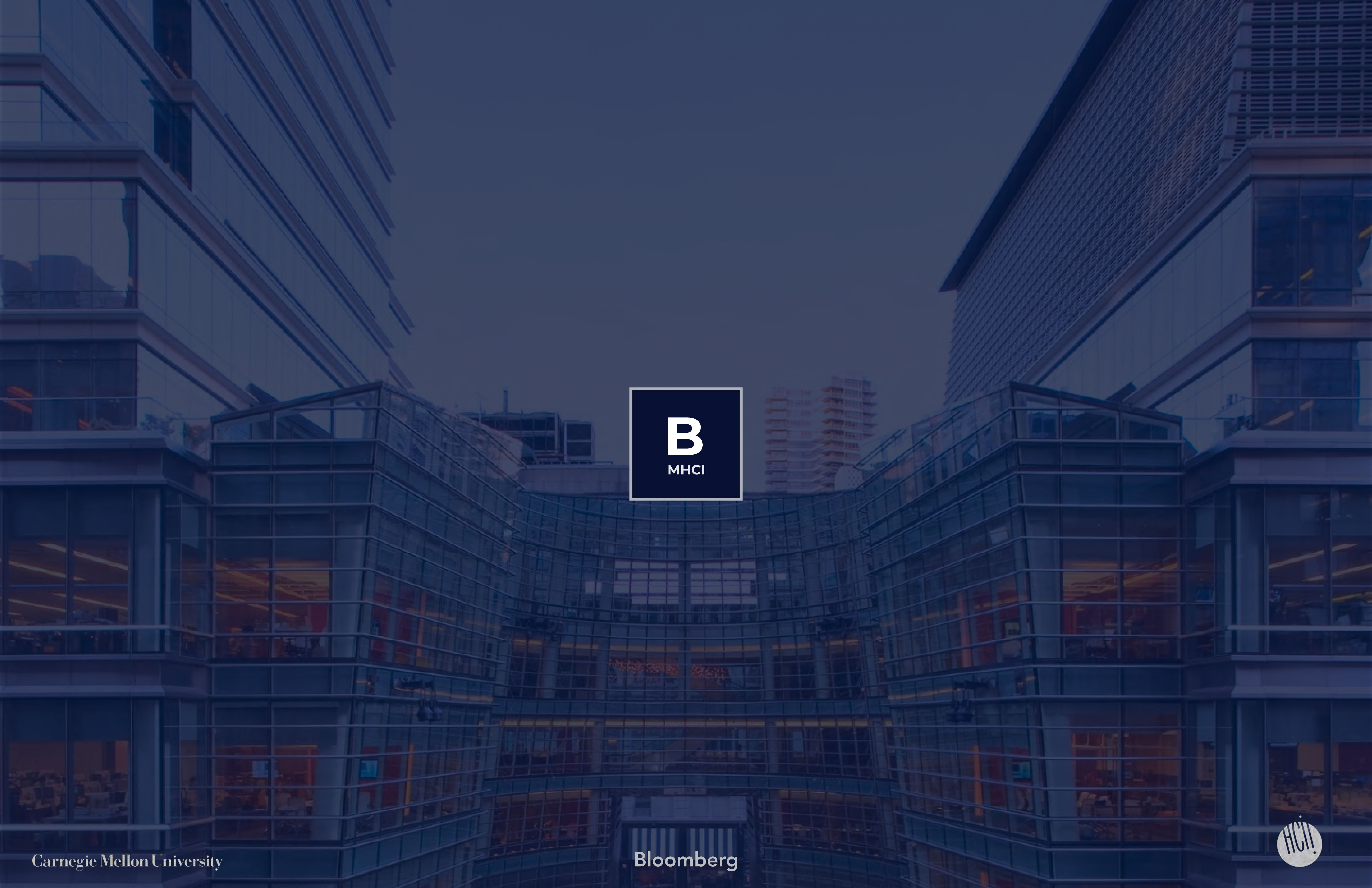
Spark Job It is launched in the DSP in order to gain access to data storage when running an experiment.

Spectro It's the name of the UI of DSP. It displays information of the resources that are still available, such as storage, GPU, and etc. to the users.

TEAM Previously known as Confluence, ML engineers often use this browser-based wiki to document their experiments and the progress of their projects.

Tensorboard It provides visualization for machine learning experiments. It currently supports scalars, images, audio, histograms, and graphs.

Tensorflow A popular open-source library that has machine learning models and algorithms already built in.


B
MHCI
Carnegie Mellon University
Bloomberg