Reminder

Project proposal

- Due 1/30, 10pm

HW1

- Due 2/1, 10pm

AI for Food Rescue

Fei Fang
Slides based on Ryan Shi’s lecture
Carnegie Mellon University

Outline

Case Study

Food Rescue Volunteer Engagement
How to use data/AI to better engage volunteers on crowdsourcing platforms?
(IAAI-20, WWW-21)
Randomized Controlled Trial
Deployed

Discussion

How to build AI for nonprofits?

Learning Objectives

Describe high-level ideas of GBM, LIME

Briefly describe

- Challenges in food rescue
- ML-based and optimization-based solution for improving volunteer notification
- ML-based method for rescue task difficulty level prediction
- Different ways to evaluate the AI-methods

Food waste and food insecurity coexist globally.

1.4 billion tons of food wasted every year

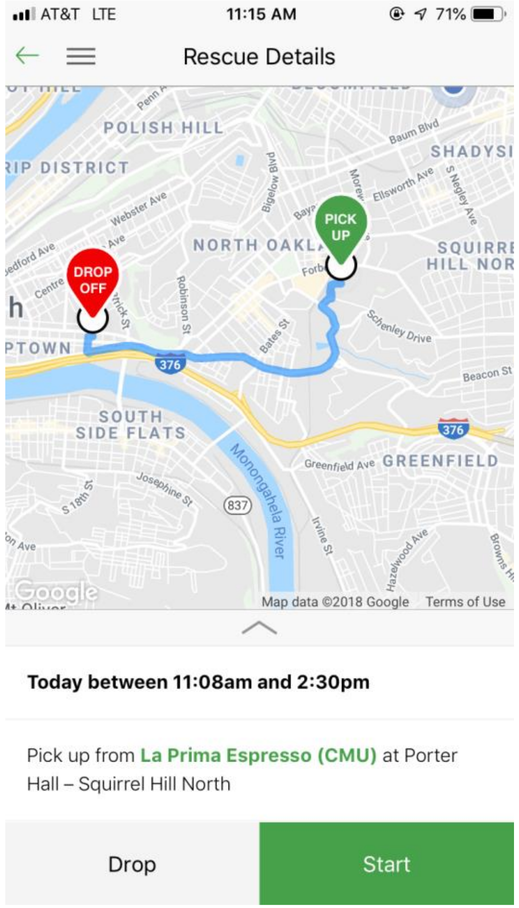
15% of US population suffer from food insecurity

20% of wasted food could feed all in food insecurity

https://www.rts.com/resources/guides/food-waste-america/

Food rescues serve as the intermediary between the food donors and food recipients.

Claim the rescue on the app

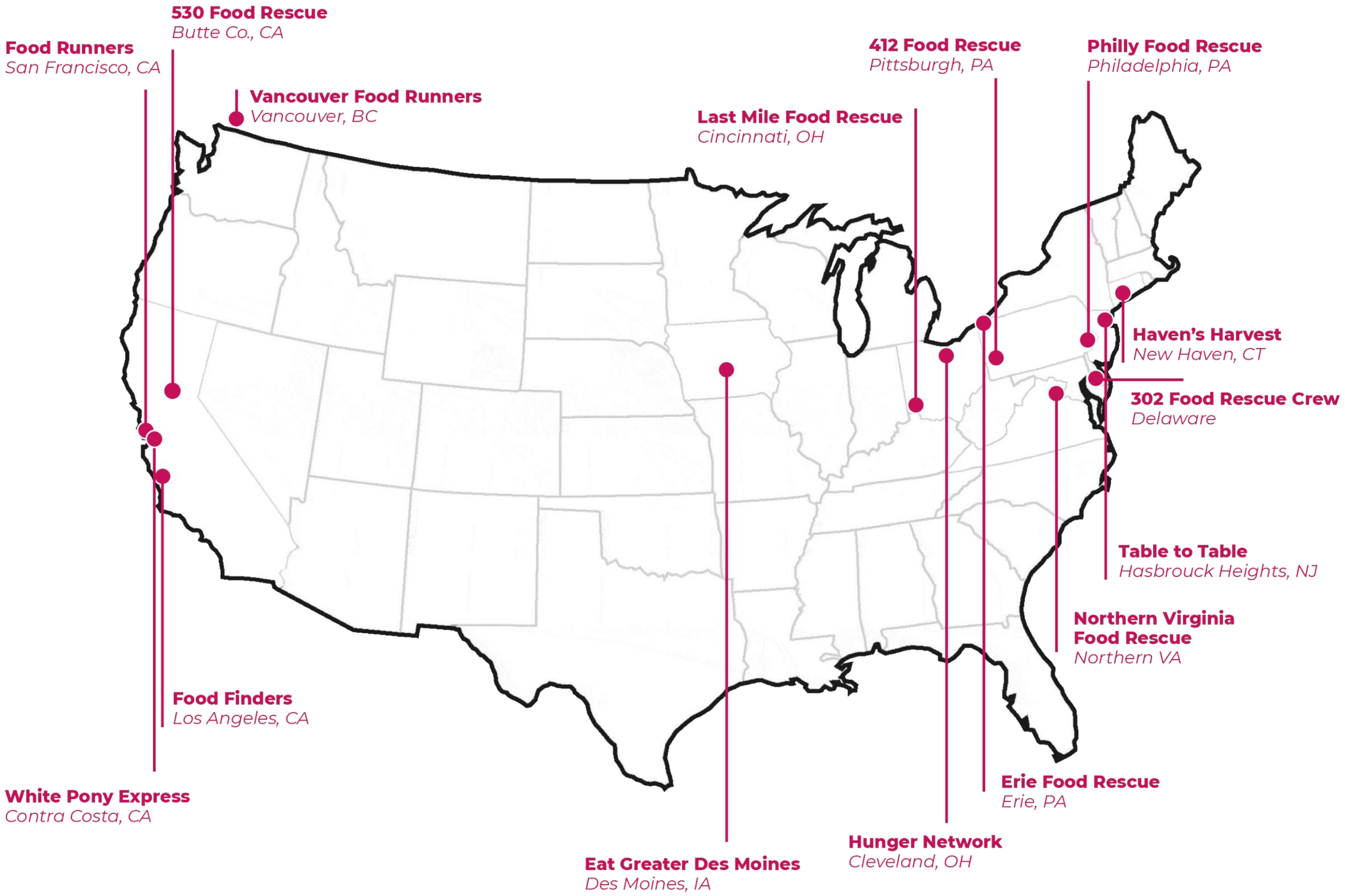


Pick up the food

Deliver the food to the recipient

Complete the rescue

Food rescues serve as the intermediary between the food donors and food recipients.



124 million pounds of food distributed

103 million meals

40,000+ volunteers

6,118 food donors

2,822 recipient community orgs

Research Question:

How to use data/AI to better engage volunteers on crowdsourcing platforms?

Volunteer engagement in food rescue is underexplored in the research literature.

- Social psychology
  [Boezeman and Ellemers, 2008; Vecina et al., 2011; Haivas et al., 2013]

- Management
  [Cuskelly et al., 2006]

- Citizen science
  [Reeves and Simperl, 2019; Sauermann and Franzoni, 2015]

- Food rescue participatory design
  [Lee et al., 2019] ← @CMU

- Volunteer notification
  [Manshadi and Rodilitz, 2020]

> "Technology amplifies human forces, rather than create them."
> --- Kentaro Toyama, *Geek Heresy*, 2015

How to use data/AI to better engage volunteers on crowdsourcing platforms?

- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

Will the rescue be claimed soon?

Rescue $\xrightarrow{\text{ML model}}$ Rescue claimed in time

Small data? ~4000 rescues at that time

Careful feature engineering and ensemble methods!

- Training data
  - May 2018 to Dec 2018
- Test data
  - Jan 2019 to May 2019

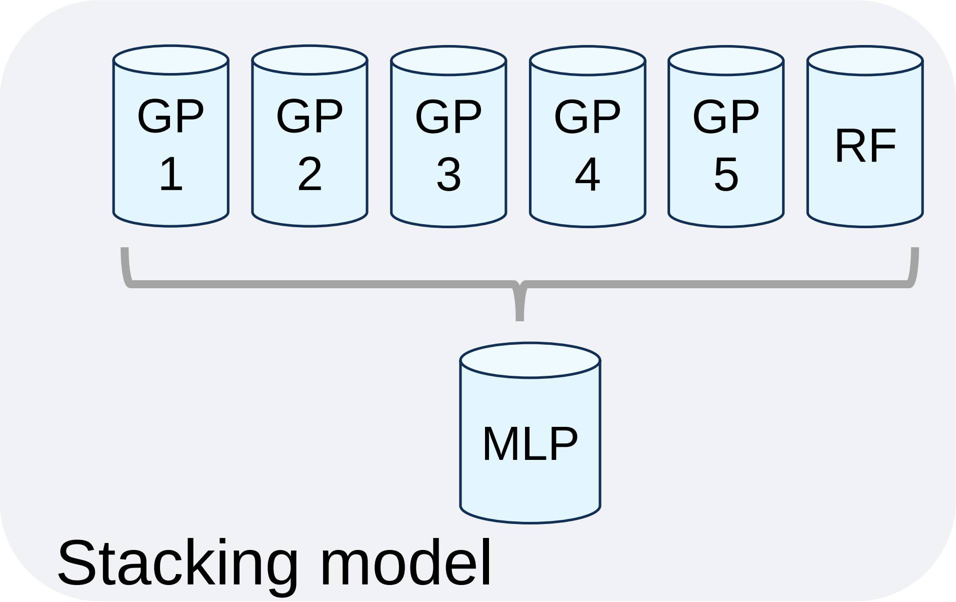
	Features
Timing	Fastest travel time of rescue
	Travel distance of rescue
	Weight of the food
	Time of day
	Time Slot
Weather	Precipitation
	Snowfall
	Average temperature
Location	AVs in donor's cell
	Average AVs in donor's neighboring cells
	AVs in recipient's cell
	AVs in donor and recipient's cells with vehicle

Will the rescue be claimed soon?

Rescue $\xrightarrow{\text{ML model}}$ Rescue claimed in time

Small data? ~4000 rescues at that time

Careful feature engineering and ensemble methods!

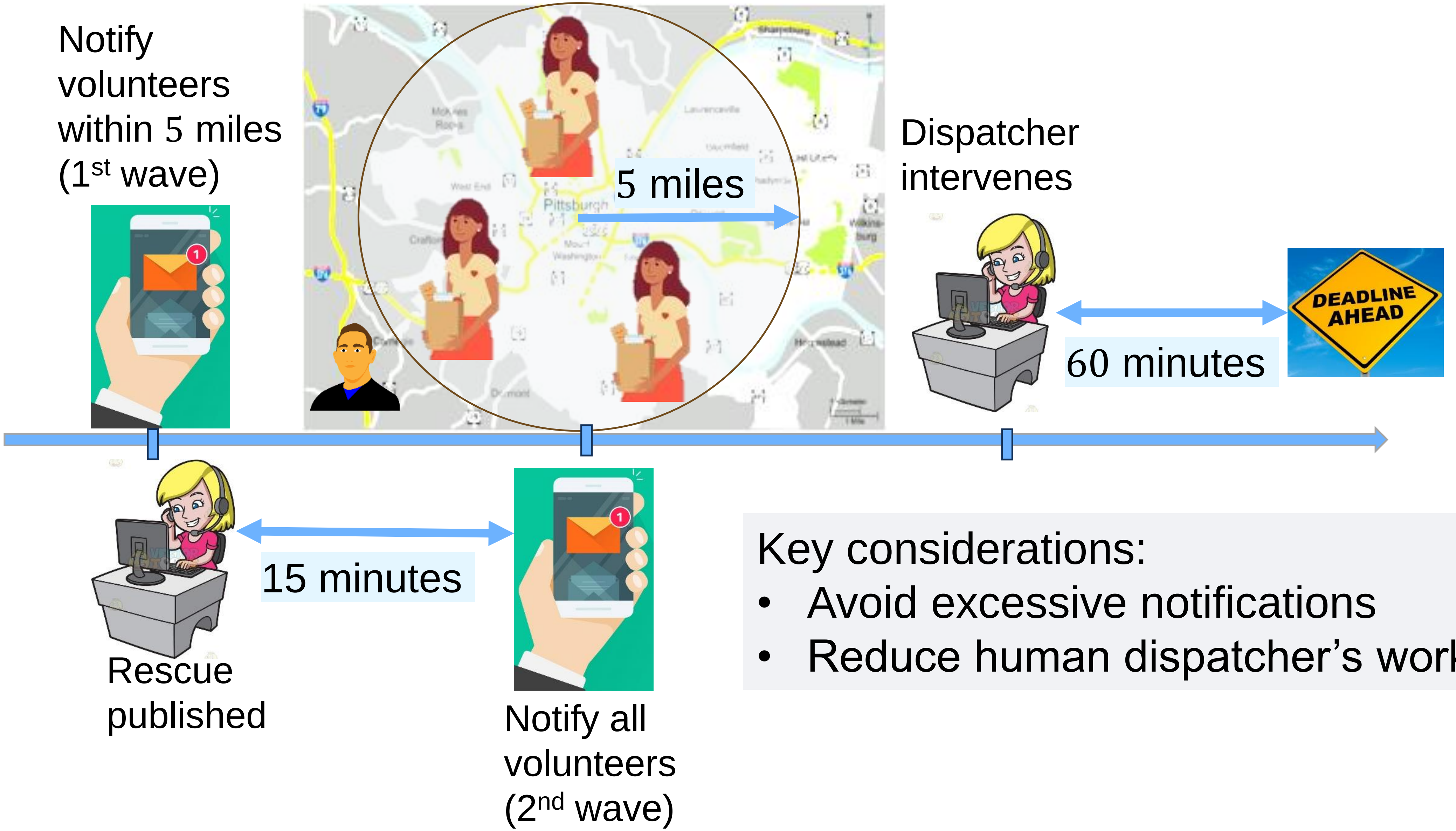


Model	Accuracy	Precision	Recall	F1	AUC
Gradient boosting	0.73	0.86	0.82	0.84	0.51
Random forest	0.71	0.87	0.78	0.82	0.54
Gaussian process	0.56	0.88	0.54	0.67	0.60
Stacking model	0.69	1.00*	0.64	0.78	0.81

How to use data/AI to better engage volunteers on crowdsourcing platforms?

- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

When should the notifications go out?



Key considerations:

- Avoid excessive notifications
- Reduce human dispatcher's workload

When should the notifications go out?

Notify volunteers within y miles (1^{st} wave)

y miles

Dispatcher intervenes

z minutes

Rescue published

x minutes

Notify all volunteers (2^{nd} wave)

$$\min_{x,y,z} \quad \text{Expected \# of volunteer notifications} + \lambda \times \text{Expected \# of human interventions}$$

$$\text{s.t.} \quad \text{Claim rate} \geq \text{threshold}$$

Solve using branch-and-bound

Optimal notifications

When would a rescue be claimed under a different notification scheme?

But all data were from 1 notification scheme!

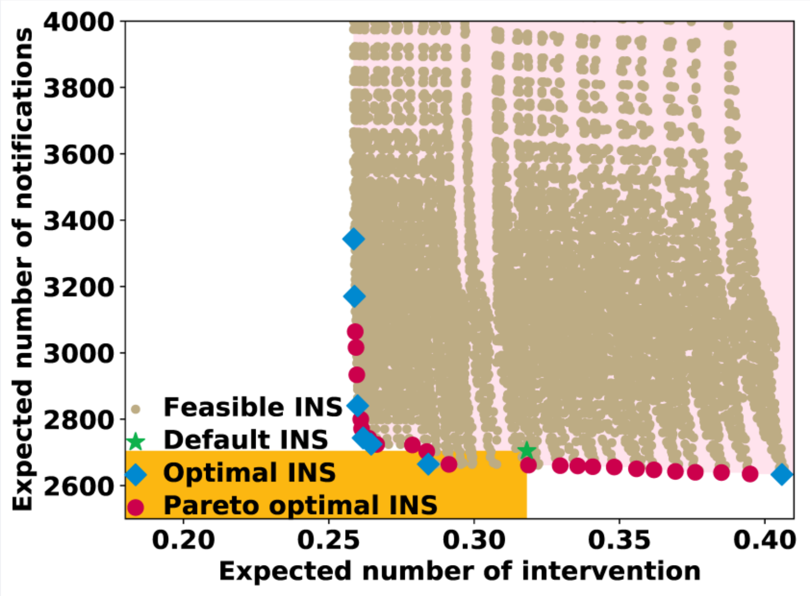
Counterfactual claim time

Assume the same volunteer would claim it

0	10	16	26
Original notification time	Original claim time	Counter-factual notification time	Counter-factual claim time

unit: minutes

- Optimize on data from May 2018 to Dec 2018
- Test on data from Jan 2019 to May 2019
- Checking Pareto frontier



INS	Interventions	Notifications
A: $(16.5, 5.5, 45)$	$-13\%\,(-0.06)$	$0\%\,(-1)$
B: $(15.5, 5.5, 32.5)$	$-24\%\,(-0.10)$	$+2\%\,(+46)$

This has been adopted since February 2020

Preliminary result

- Higher claim rate, lower claim time, less notifications
- Many confounding factors might exist

Condition	Claim Rate	Average time from publish to claim (min)	Average # of push notifications sent
Before 2/10/2020 (Previous scheme)	0.84	78.43	11499.45
2/10/2020-3/1/2020 (New scheme)	0.88	43.05	9167.52
After 3/1/2020 (After COVID)	0.92	39.73	9735.54

How to use data/AI to better engage volunteers on crowdsourcing platforms?

- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

Send push notifications to the “right” people.

First wave notification volunteers within 5 miles

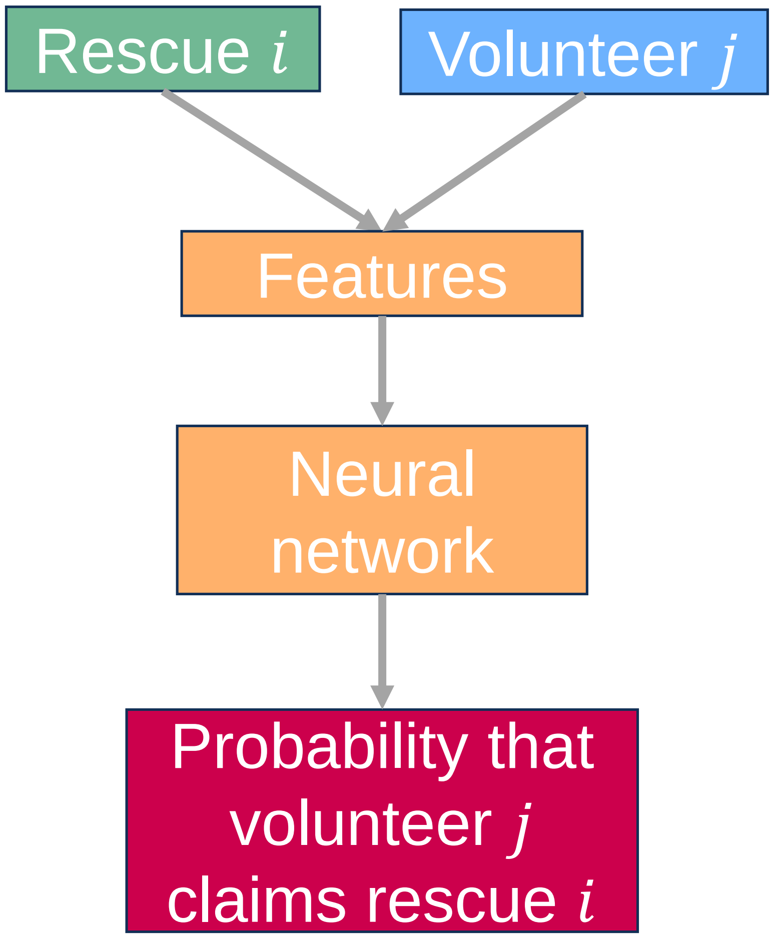
On average, a rescue sends first wave push notification to 964 volunteers.

Hit rate: the percentage of rescues whose first wave notifications include the volunteer who claimed the rescue in the end.

Currently, the hit rate is 43.9%.

Given a food rescue trip (donor, recipient, time, etc.), can we find the best 964 volunteers to send notifications?

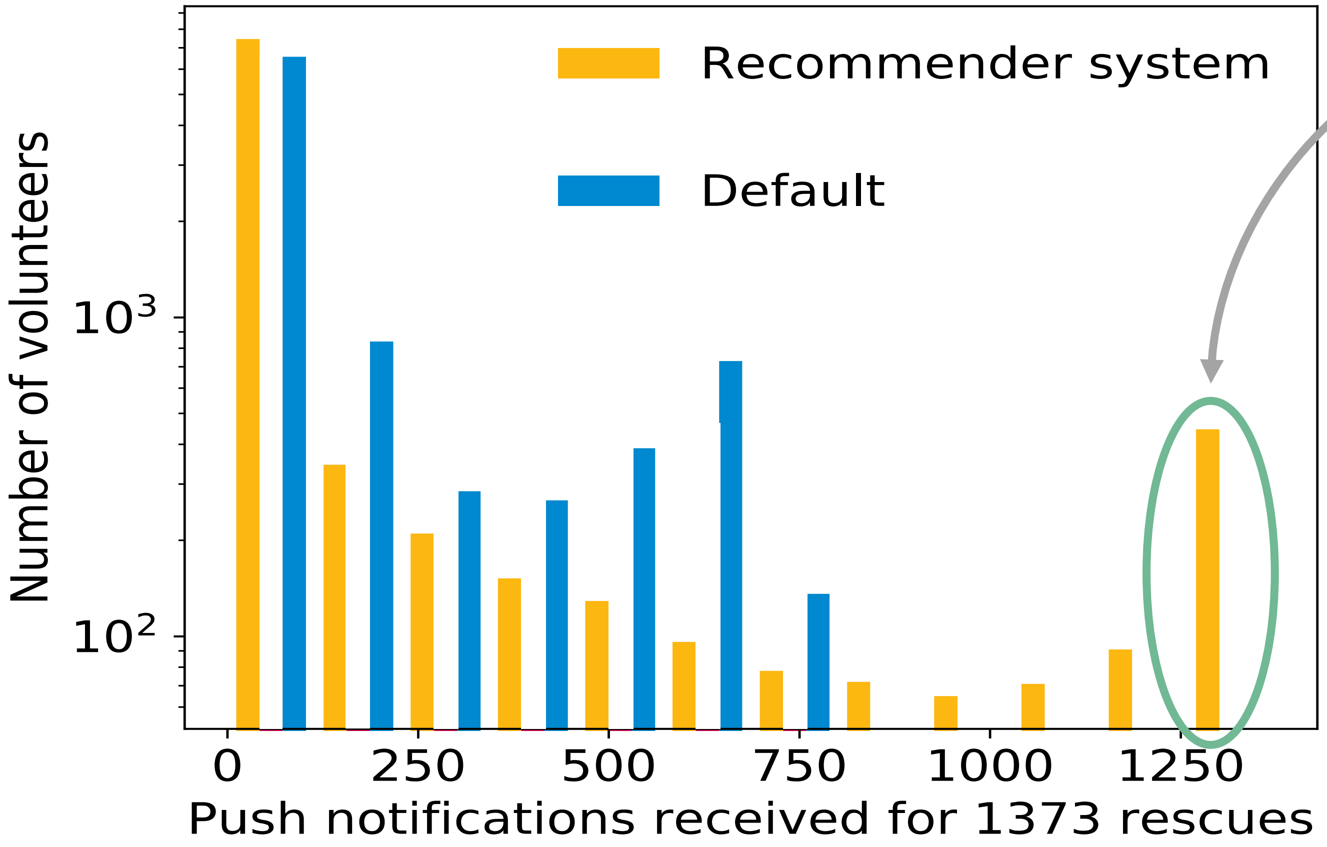
The machine learning model achieves a 72.7% hit rate (compare to the current 43.9%).



Model	HR@k (SD)
NN	**0.7269 (0.0310)**
RF(1:1)	0.5989 (0.0395)
RF(1:20)	0.6035 (0.0511)
GBDT(1:1)	0.6235 (0.0549)
GBDT(1:20)	0.5394 (0.0152)
SM(1:1)	0.4996 (0.0005)
SM(1:20)	0.5219 (0.0125)
Default	0.4392 (N/A)

But this is not the end of the story.

Histogram of #notifications received by each volunteer



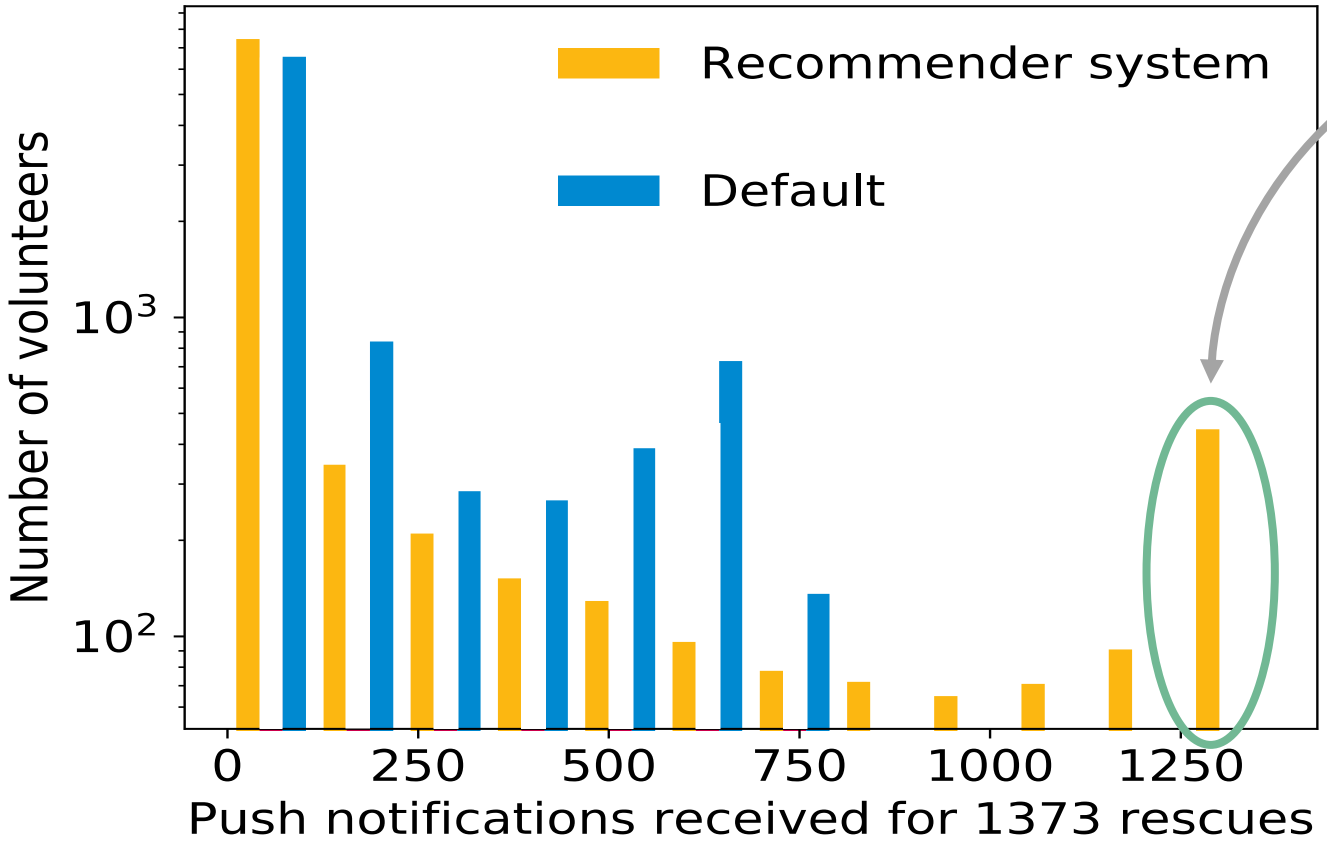
What is our ML model doing here?

The ML model discovers a small subset of volunteers and sends them notifications almost all the time!

Why is this a problem?

But this is not the end of the story.

Histogram of #notifications received by each volunteer



These 446 volunteers contain 39 of the top 50 most frequent volunteers.

They have been the most productive volunteers in the past.
They would get annoyed.
They would uninstall the app.

Then many rescues would go unclaimed.

How would volunteers react to the intervention?
Deploying an ML model might backfire on its very purpose.

How many notifications do you want to receive every day?

Each volunteer receives at most L notifications a day [Adomavicius & Kwon, 2014]

For current rescue i, determine who to send notifications to by planning with the set of future rescues R

$x_{ij} \in \{0,1\}$: Whether to send notification of rescue i to volunteer j

$p_{ij} \in [0,1]$: Output of ML model indicating the prob. that volunteer j will claim rescue i

$b_j \in \{0, \ldots, L\}$: Number of notifications volunteer j can receive for the rest of the day

$$
\begin{aligned}
\max_{x} \quad & \sum_{j \in V} \left(p_{ij} x_{ij} + \sum_{i' \in R} p_{i'j} x_{i'j} \right) && \text{[Maximize claim probability]} \\
s.t. \quad & \sum_{j \in V} x_{i'j} \le k, \quad \forall i' \in R \\
& \sum_{j \in V} x_{ij} \le k && \text{[Notify at most } k \text{ volunteers per rescue]} \\
& \boxed{x_{ij} + \sum_{i' \in R} x_{i'j} \le b_j, \quad \forall j \in V} && \text{[Each volunteer receives at most } L \text{ notifications per day]} \\
& x_{ij} \in \{0, 1\}, \quad \forall i \in R, \forall j \in V
\end{aligned}
$$

ML + online planning

February 28 [CURRENT RESCUE]	
10:30	Carnegie Library to WCHA

February 21	
10:00	Giant Eagle Greenfield to Veteran’s Leadership Center
11:15	CMU Gates 5th to Ace Daycare
13:00	…
16:00	…

February 14	
9:00	Target Waterfront to Care 4 You
10:15	…
14:00	…
16:00	…

We cannot look into the future, but we can use the past to predict the future.

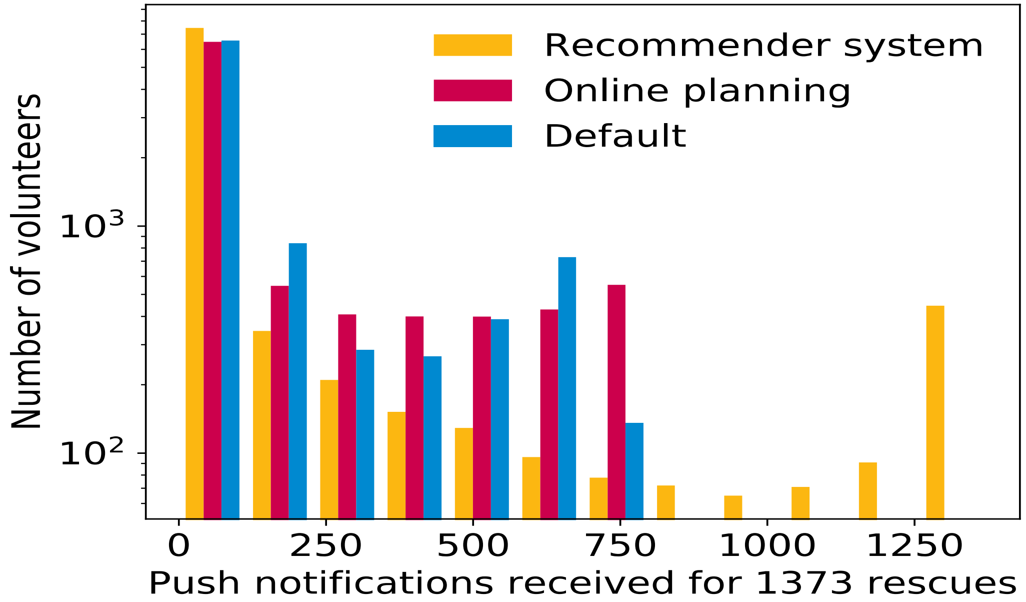
We sample rescue trajectories 7 days ago, 14 days ago, 21 days ago.

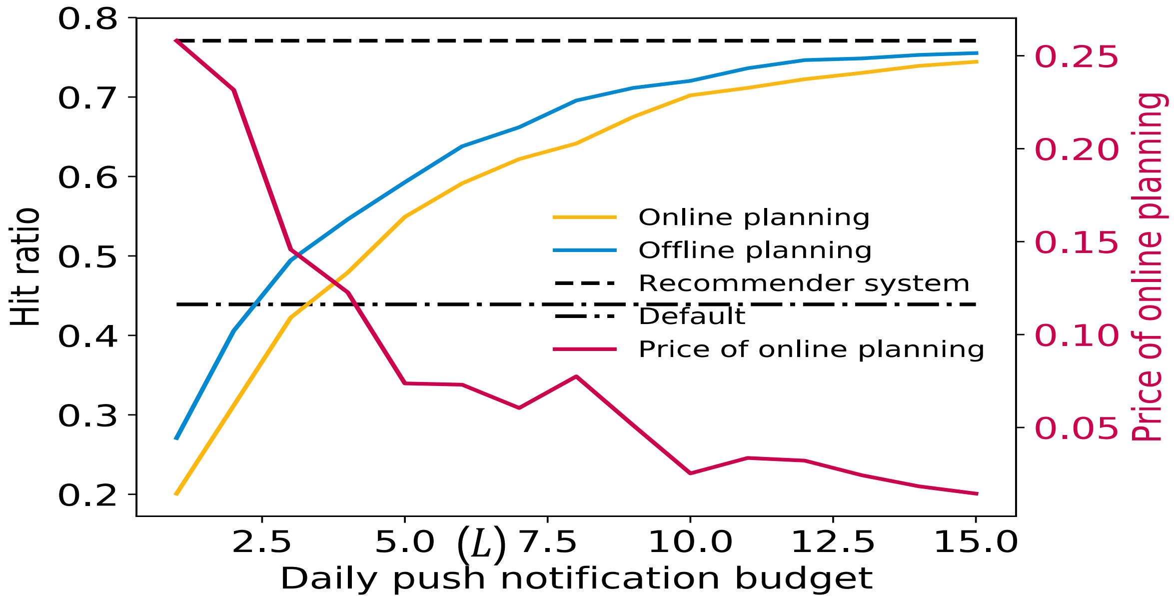
$$
\begin{aligned}
(\Pi_i) \quad \max_{x} \quad & \sum_{j \in V} \left(p_{ij} x_{ij} + \sum_{i' \in R} p_{i'j} x_{i'j} \right) \\
s.t. \quad & \sum_{j \in V} x_{i'j} \leq k, \quad \forall i' \in R \\
& \sum_{j \in V} x_{ij} \leq k \\
& x_{ij} + \sum_{i' \in R} x_{i'j} \leq b_j, \quad \forall j \in V \\
& x_{ij} \in \{0, 1\}, \quad \forall i \in R, \forall j \in V
\end{aligned}
$$

Solve for X, one column at a time.

Diversity constraint is strictly enforced.

Online planning-based rescue-specific notification





Avoid the over-concentration with $L = 6$

Hit Rate @ k = 0.645, much better than current practice

Price of online planning

$$1 - \frac{\mathrm{HR(online)}}{\mathrm{HR(offline)}} < 10\%$$

But do we only/really care about the hit rate?

We ran an RCT with 412FR.

How to use data/AI to better engage volunteers on crowdsourcing platforms?

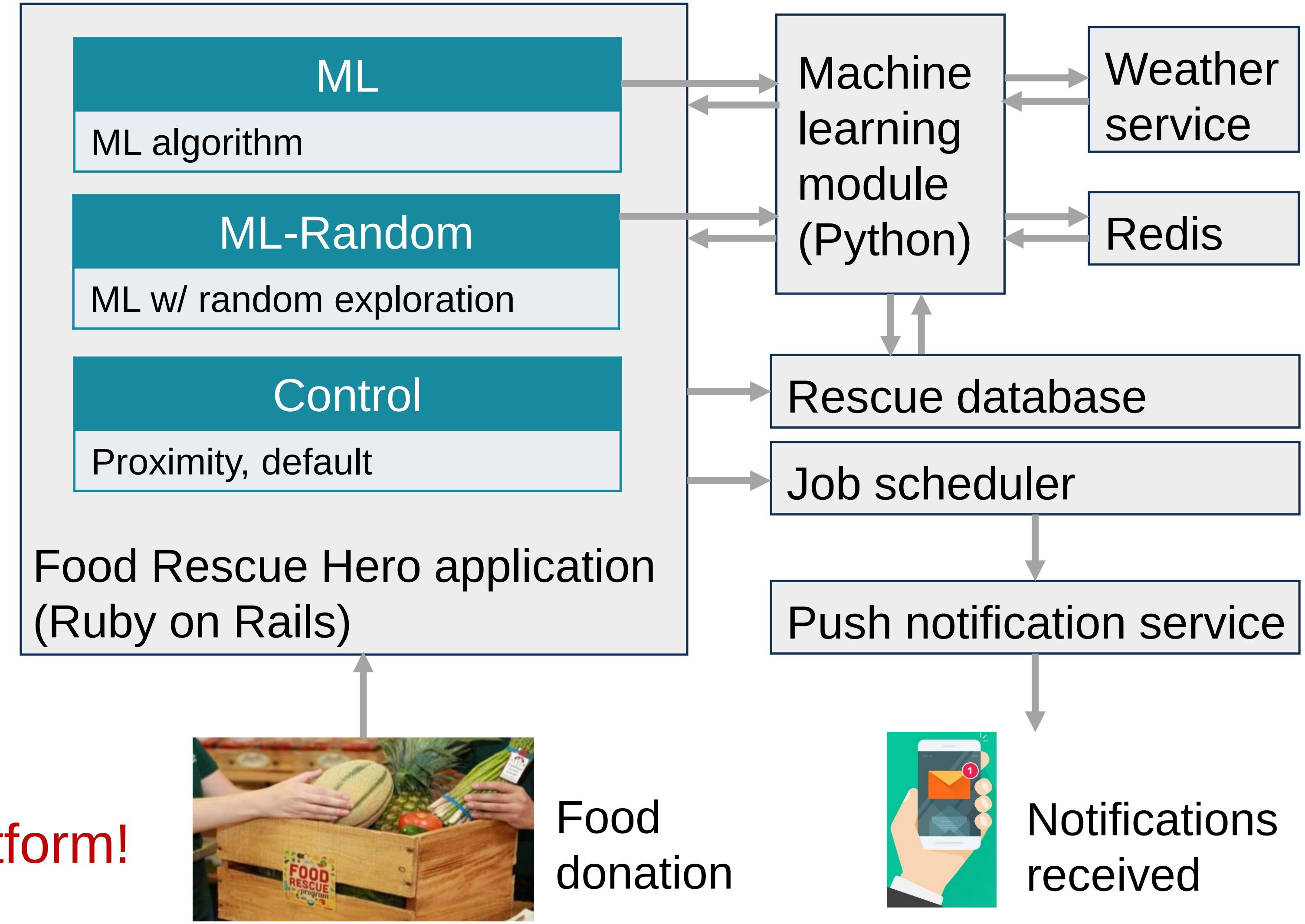
- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

Deployment in a real-world system is highly nontrivial.

Time: May 2022 – June 2022

Scope: All food donations in the Greater Pittsburgh area

Environment: Real-time notification module in a Ruby on Rails application



First ever A/B testing on the platform!

The numbers speak for themselves.

Control
Proximity, default

ML
ML algorithm

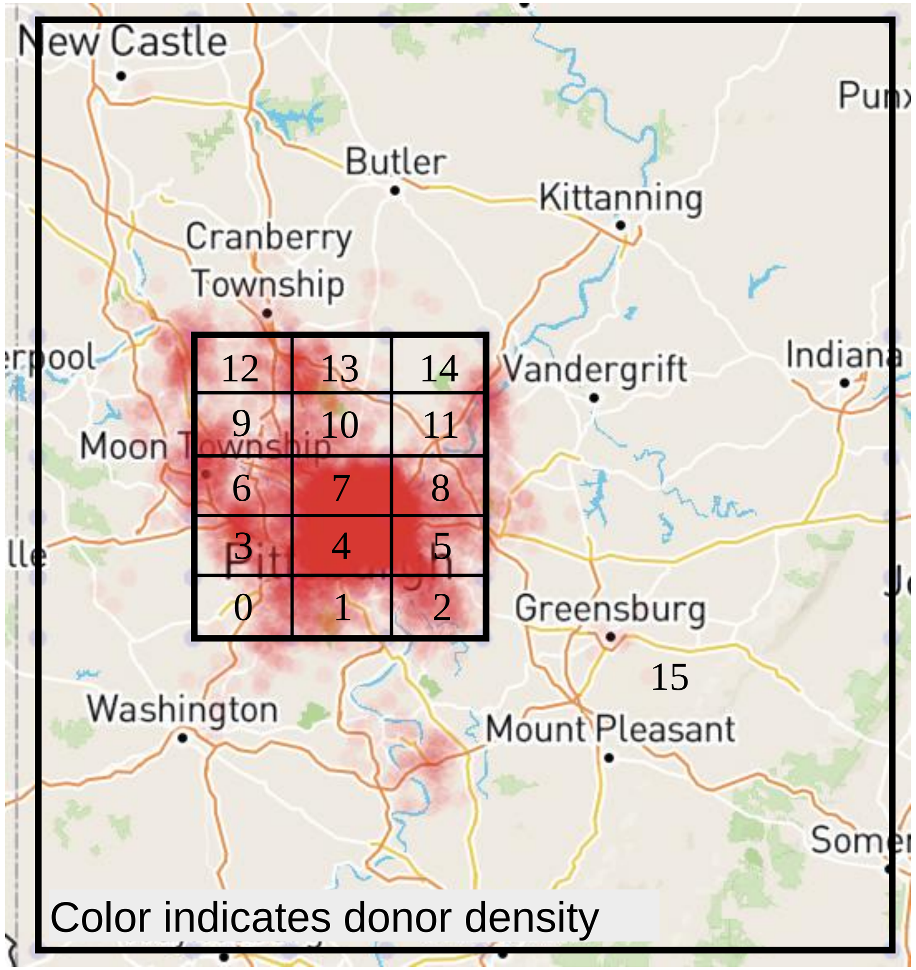
ML-Random
ML w/ random exploration

Randomized Controlled Trial (May-June 2022)

Condition	Hit Rate (p-value w.r.t. control)	Claim Rate (p-value w.r.t. control)
Control	0.468	0.807
ML	0.651 (0.001)	0.882 (0.047)
ML-Random	0.489 (0.696)	0.844 (0.317)

- The ML model significantly improved the hit rate (as expected).
- The ML model also significantly improved the claim rate.
- ML with random exploration also worked, but not as significant. This is an important future direction.

We also learned important lessons.



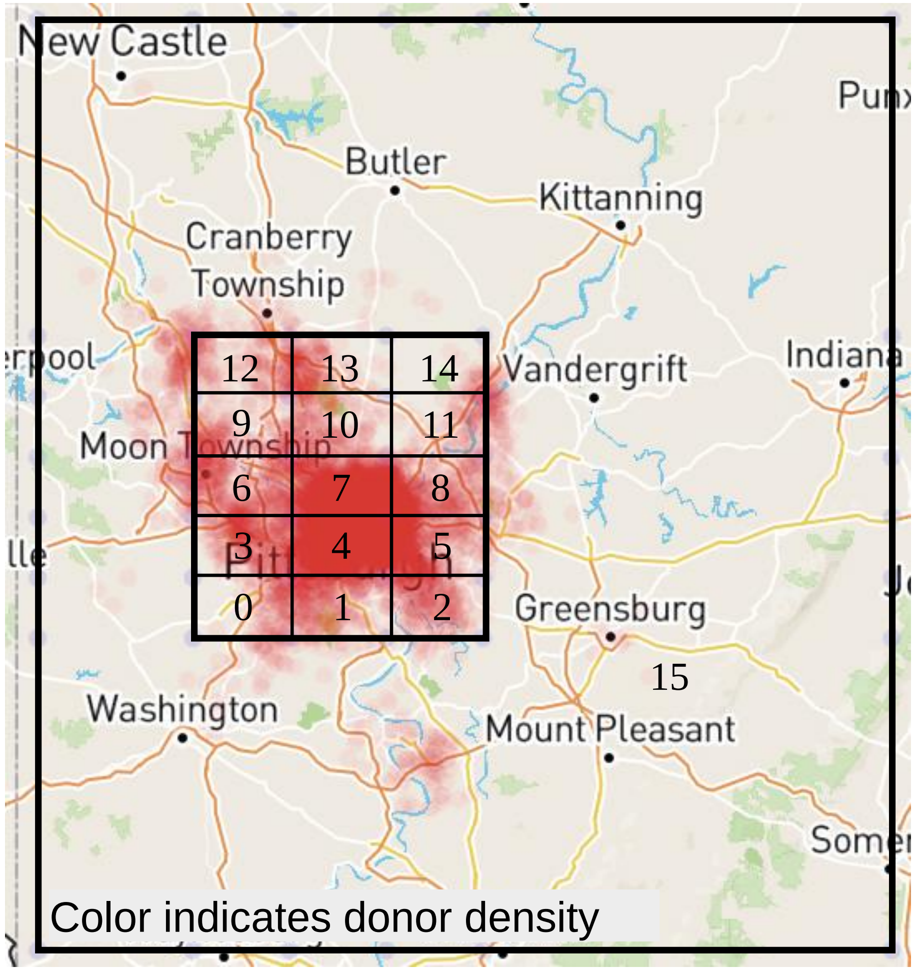
Area 4: Downtown Pittsburgh		
Condition	Hit Rate (p-value)	Claim Rate (p-value)
Control	0.541	0.897
ML	0.688 (0.172)	0.865 (0.620)
ML-Random	0.593 (0.577)	0.857 (0.486)

The ML model's impact is limited in downtown, where

- volunteers are abundant, and
- transportation is easy.

But this is okay, the status quo is good enough.

We also learned important lessons.

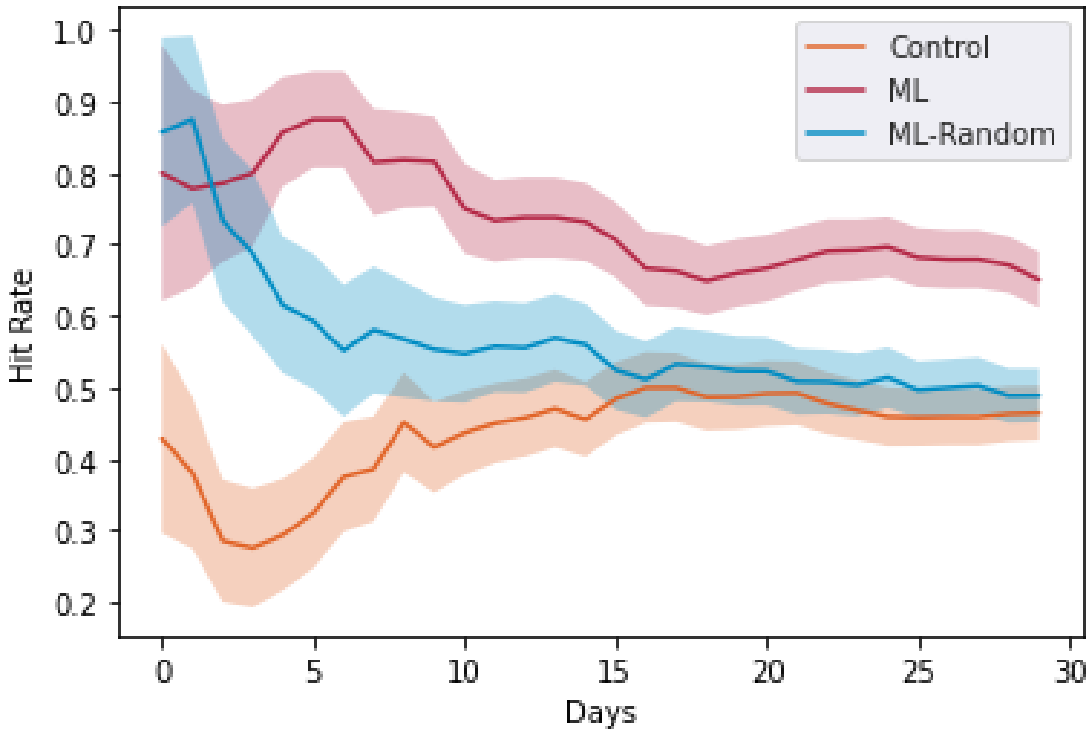


Area 15: Outer suburbs of Pittsburgh		
Condition	Hit Rate (p-value)	Claim Rate (p-value)
Control	0.125	0.457
ML	0.409 (0.057)	0.688 (0.057)
ML-Random	0.174 (0.677)	0.575 (0.308)

The ML model's impact is much bigger in outer suburbs, where

- volunteers are less abundant,
- transportation is difficult, and
- the current approach fails catastrophically.

We also learned important lessons.



Performance of the ML model degrades over time.

Need regular model update

Deploying on AWS Sagemaker to fully automate the ML pipeline!

How to use data/AI to better engage volunteers on crowdsourcing platforms?

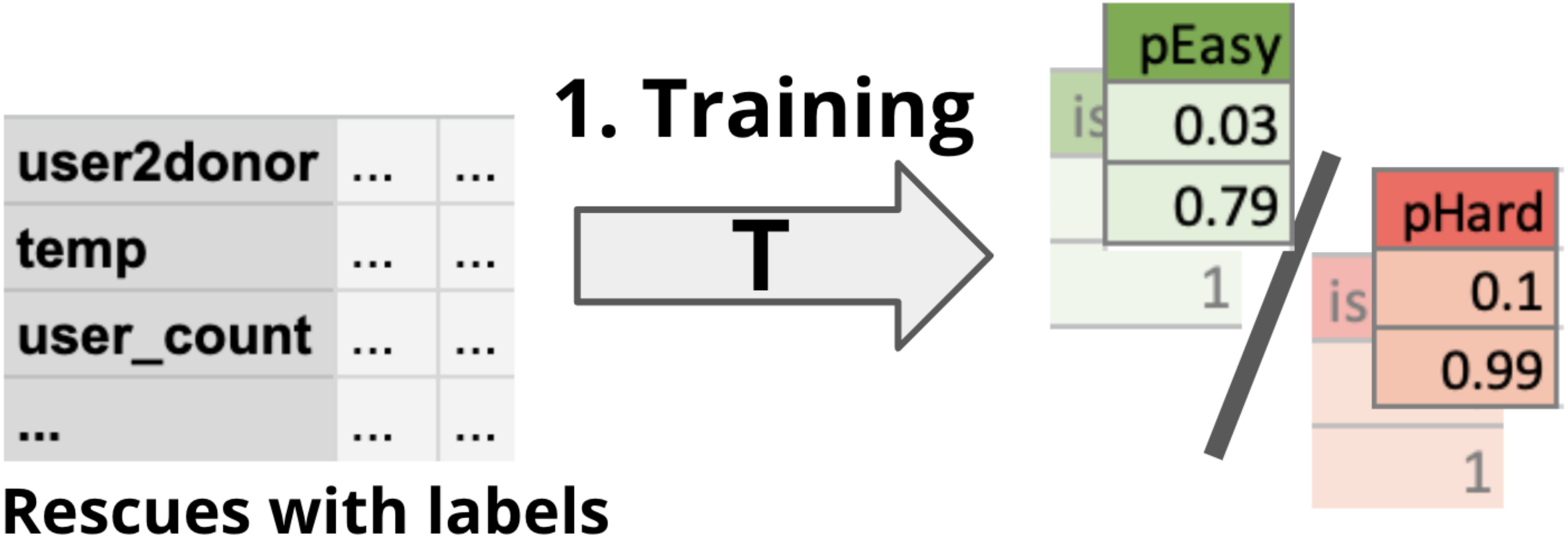
- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

Predict the difficulty level of task with Machine Learning

Can we predict whether a rescue task will be difficult for a volunteer?

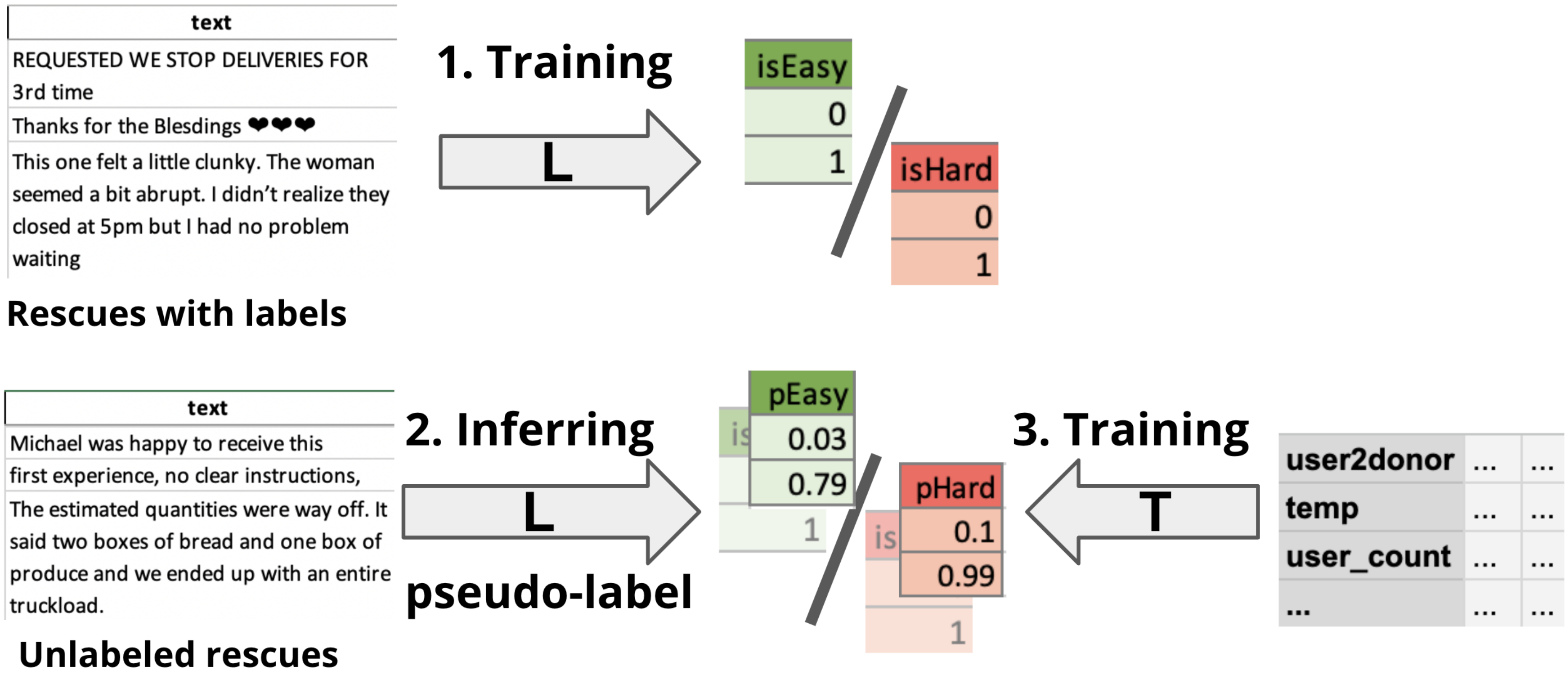
- Experts from 412 Food Rescue labeled a small set of rescues based on volunteers’ feedback, dial pad record etc. Labels: hard, easy, or neither

- Build simple ML models for classification (one for hard, one for easy):



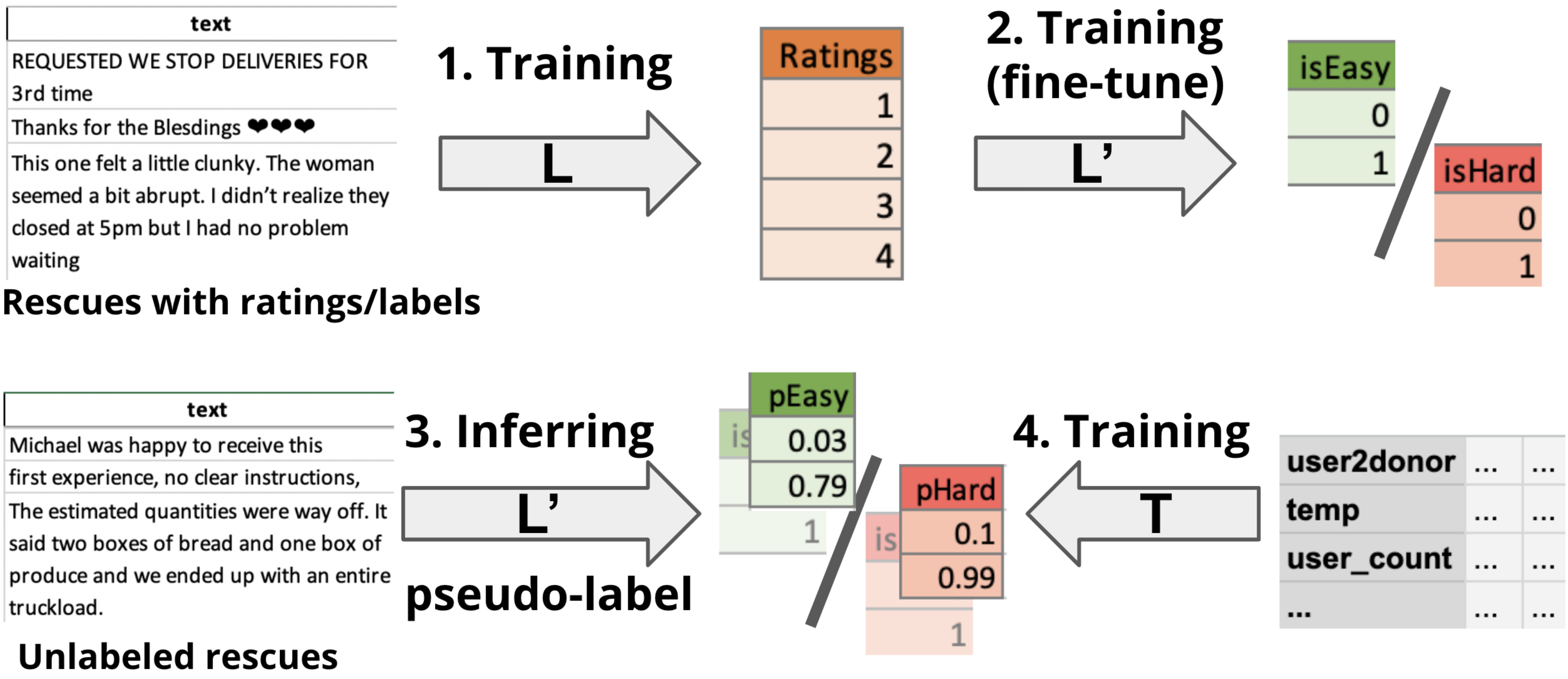
Predict the difficulty level of task with Machine Learning

Make better use of the unlabeled volunteers' text feedback data

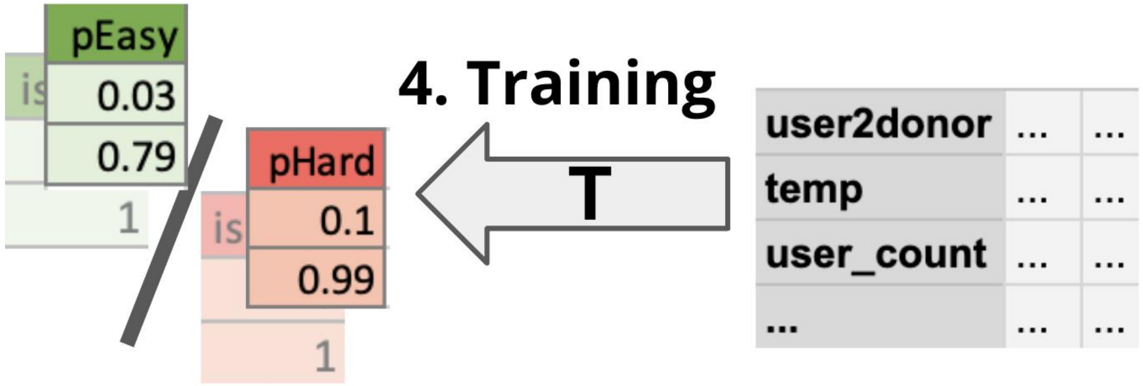


Predict the difficulty level of task with Machine Learning

Make better use of the unlabeled volunteers' feedback (text + rating!)



Gradient Boosting



Gradient boosting machine (GBM), a decision-tree ensemble method that sequentially train decision trees.

In each iteration, train a new decision tree to improve the ensemble's performance

Let $F_m(x)$ be the ensemble model after m iterations
After $m-1$ iterations, for all training data, compute pseudo-residuals

$$r_{im} = -[\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}]$$

Train a decision tree h_m use r_{im} as labels
Let $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$ with γ_m being a tunable parameter to minimize loss $L(y_i, F_m(x_i))$

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp.1189-1232
https://en.wikipedia.org/wiki/Gradient_boosting

Gradient Boosting in Practice

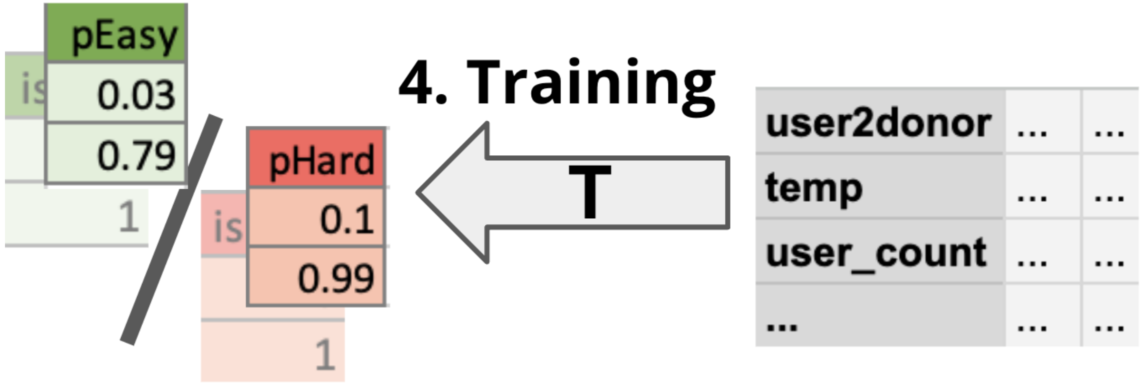
In practice, often use XGBoost, LightGBM for higher efficiency

```
import xgboost as xgb
params = {"objective":"binary:logistic",'colsample_bytree': 0.3,'learning_rate': 0.1,
                'max_depth': 5, 'alpha': 10}
classification = xgb.XGBClassifier(**params)
classification.fit(X_train, y_train)
```

```
import lightgbm as lgb
lgb_train = lgb.Dataset(X_train, y_train)
lgb_eval = lgb.Dataset(X_test, y_test, reference=lgb_train)
params = {'boosting_type': 'gbdt',
              'objective': 'binary',
              'num_leaves': 40,
              'learning_rate': 0.1,
              'feature_fraction': 0.9
              }
gbm = lgb.train(params,
    lgb_train,
    num_boost_round=200,
    valid_sets=[lgb_train, lgb_eval],
    valid_names=['train','valid'],
   )
```

https://neptune.ai/blog/gradient-boosted-decision-trees-guide#:~:text=In%20gradient%20boosting%2C%20an%20ensemble,average%20of%20all%20weak%20learners.

Predict the difficulty level of task with Machine Learning



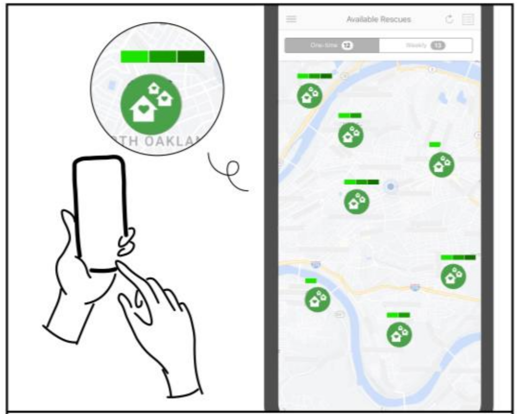
Predictor	Validation Set AUC	Validation Set Std. Dev.	Test Set AUC	Test Set Std. Dev.
GBM	**0.686**	**0.118**	0.710	0.023
RF	0.663	0.057	0.703	0.027
LR	0.562	0.055	0.535	0.025
SVM	0.485	0.050	0.470	0.022
MLP	0.495	0.027	0.495	0.031
KNN	0.654	0.022	0.643	0.021

Make better use of the feedback data leads to better predictions

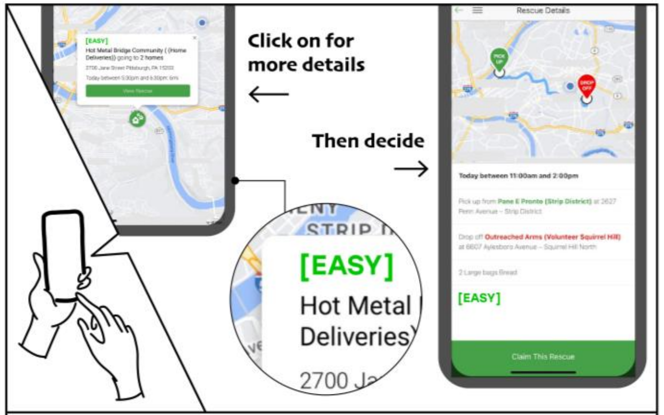
Algorithm	Easy AUC	Easy Std. Dev.	Hard AUC	Hard Std. Dev.
Ours	**0.710**	**0.023**	**0.685**	**0.041**
Baseline 1	0.543	0.024	0.495	0.025
Baseline 2	0.709	0.037	0.563	0.000

How to present the predicted difficulty to volunteers?

Charlie, a new volunteer with limited or no experience in food rescue initiatives, is interested in exploring employment options through the Food Rescue app. Charlie doesn't want to be frustrated as a beginner.



On the map, Charlie can view **all the tasks, and the difficulty levels associated with each task** are indicated.



Charlie can click on a task to check more information like time and location. Then, Charlie can click on the button "View Rescue" for more details about the task and decide to take on the task. **Difficulty levels are always displayed**. Charlie can decide to take on **tasks that are available on the map**.

When not using the app, Charlie will get **notifications of tasks across all difficulty levels with difficulty levels shown**.

Front-end	Back-end Scaffold-ing	Design Concept of Each Storyboard
No display	Low-level	A.1: No display, showing all tasks on the map, and sending notifications of tasks of all difficulty levels
Display	Low-level	A.2: With display, showing all tasks on the map, and sending notifications of tasks of all difficulty levels
No display	Medium-level	B.1: No display, showing all tasks on the map, and customizing notifications by only sending easy tasks
Display	Medium-level	B.2: With display, showing all tasks on the map, and customizing notifications by only sending easy tasks
No display	High-level	C.1: No display, only showing easy tasks on the map, and customizing notifications by only sending easy tasks
Display	High-level	C.2: With display, only showing easy tasks on the map, and customizing notifications by only sending easy tasks

Findings from the User Study

- Volunteers value the difficulty prediction AI as a decision-supporting tool to help them navigate the complicated workflow

- In terms of integration method, they prefer the least back-end scaffolding and more front-end display to integrate the AI

- They strongly request more explanations to better understand the difficulty prediction AI with a goal to better support their decision-making process

How to use data/AI to better engage volunteers on crowdsourcing platforms?

- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment
- Rescue difficulty level prediction
- Explanations for prediction

Prompt Large Language Models to provide explanations

Type	Explanation
	Instance 1
Natural Language	This task is HARD for you because you have less experience, a mixed satisfaction from previous rescues, and the recipient location is far.
Tag-based	Hard for less experienced • Prior mixed satisfaction • Far recipient location
Augmented Tag-based	Hard for less experienced (your past rescue counts lower than 26%) • Prior mixed satisfaction (your average rating higher than 28%) • Far recipient location (higher than 94%)

Prompt Large Language Models to provide explanations

Step 1: Extract top 10 features that influence the most model's prediction using LIME

Step 2: Ask GPT-4 to generate explanations using these top features

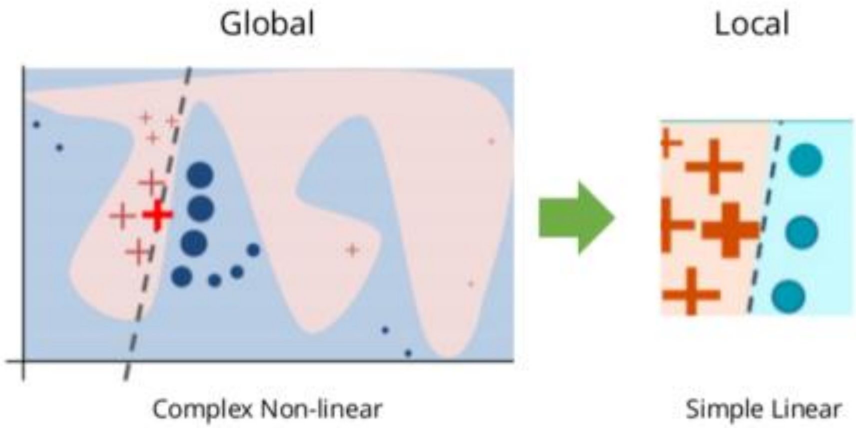
```
% [Instruction for the explanation]
You are tasked with explaining how different
↪ features influence the difficulty level of
↪ food rescue tasks to an audience with no
↪ expertise in AI...
In the context of LIME, or Local
↪ Interpretable Model-agnostic Explanations,
↪ interpreting the outputs...

% [Feature Meanings]
PRCP means precipitation
...
user_counts means how many rescues has the
↪ user completed previously, higher means
↪ more experience

% [Top 10 Features from LIME]
Feature user_counts <= 5.00: 0.69
Feature total_quantity > 10.00: 0.15
...

Complete this: this task is {HARD/EASY}
↪ because
```

LIME (Local Interpretable Model-Agnostic Explanations)



$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \ \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

f: black box ML model to be explained
G: a class of interpretable models (e.g., linear regression, decision tree)
g: an interpretable model to explain f
$\pi_x(z)$: distance metric (measure the distance from data point z to x)
$L(f, g, \pi_x)$: a measure of how unfaithful g is in approximating f in the locality defined by π
Ω: Complexity of model (e.g., depth of decision tree, non-zero weights in linear regression)

Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).
https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/?irclickid=QJJ2MdUa7xyIUXTzH0VWsUnYUkHzaPzW0Xdm0c0&irgwc=1

LIME (Local Interpretable Model-Agnostic Explanations)

Training procedures

Step 1: Given a data point x, sample data points around x

Step 2: Label the sampled data points using the black box model f

Step 3: Solve the optimization problem to get $\xi(x)$

Using LIME in Practice

```
from lime.lime_tabular import LimeTabularExplainer

def predict(X):
    return models['lgb'].predict(X)

explainer = LimeTabularExplainer(train, feature_names=features, class_names=['not hard', 'hard'], mode
='regression', discretize_continuous=True, random_state=515)
testcase = i = 146
exp = explainer.explain_instance(test[i], predict, num_features=10)
exp.show_in_notebook(show_table=True)
exp.save_to_file('report.html')
```

Outline

Case Study

Food Rescue Volunteer Engagement
How to use data/AI to better engage volunteers on crowdsourcing platforms?
(IAAI-20, WWW-21)
Randomized Controlled Trial
Deployed

Discussion

How to build AI for nonprofits?

Discussion: How to build AI for nonprofits?

How to connect with these public serving nonprofits?

How to identify a pain point that AI can solve or mitigate?

How to engage practitioners from nonprofits?

Backup Slides

Applied

Food Rescue
(IAAI-20, WWW-21)
(Manag. Sci. Submitted)
Randomized Controlled Trial
Deployed

Technical

Bandit Data-Driven
Optimization
(AAAI-22)

AI4NP

AI for Nonprofits
Research

How to use data/AI to better engage volunteers on crowdsourcing platforms?

- Rescue status prediction
- Generic notifications
- Rescue-specific notifications
- Field deployment

How to carry out principled iterative prediction-prescription in low-resource settings?

- Bandit data-driven optimization

What is AI for nonprofits research?

- Research overview
- Future work

Volunteer engagement as iterative prediction-prescription

Features
Volunteer & rescue attributes

ML prediction

Predicted labels
Claim probability of each volunteer

Optimization

Interventions
Send notifications to whom

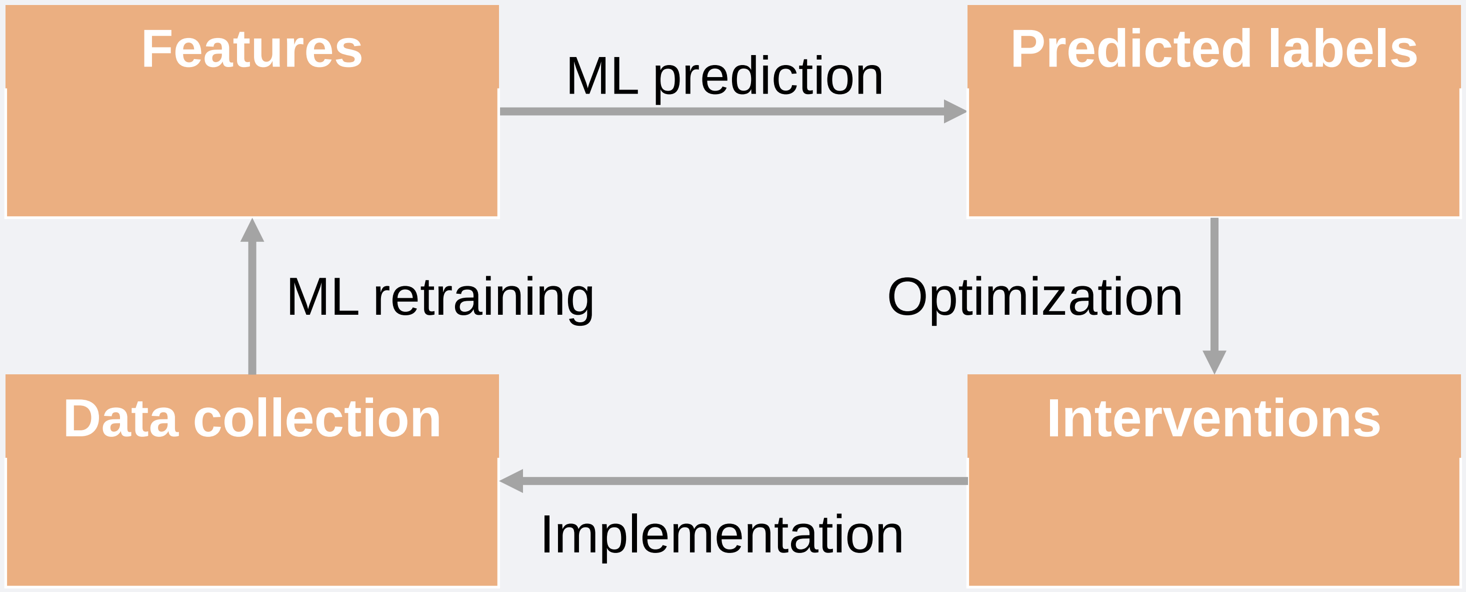
Implementation

Data collection
Observe which volunteer claims it

ML retraining

Application-independent iterative prediction-prescription

- Data-driven optimization
  [Bertsimas and Kallus, 2020; Elmachtoub and Grigas, 2017]
- Decision-focused learning
  [Donti et al., 2017]
- Contextual/linear bandit
  [Dani et al., 2008; Lai and Robbins, 1985]



Applied

Food Rescue
(IAAI-20, WWW-21)
(Manag. Sci. Submitted)
Randomized Controlled Trial
Deployed

Technical

Bandit Data-Driven
Optimization
(AAAI-22)

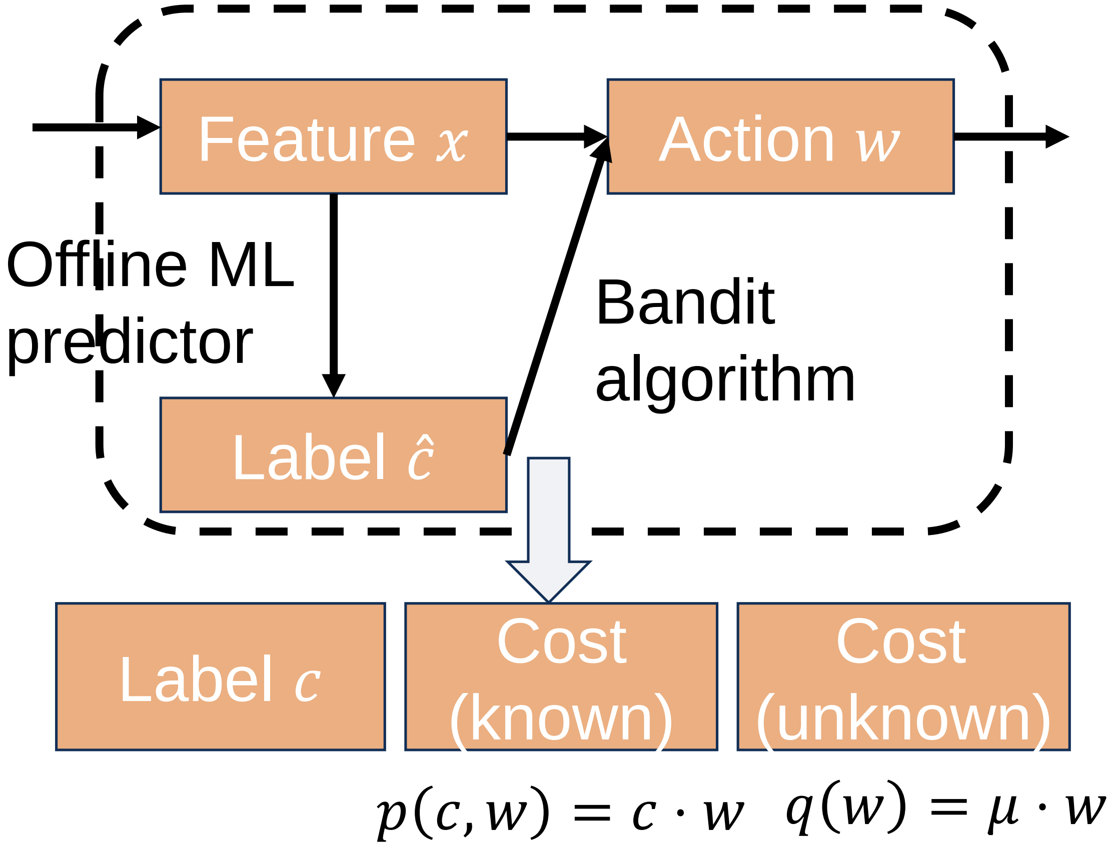
AI4NP

AI for Nonprofits
Research

Research Question:

How to carry out principled iterative prediction-prescription in low-resource settings?

We propose bandit data-driven optimization.



Optimal policy: $\pi(\mathbf{x}) = \arg\min_{\mathbf{w}} \mathbb{E}_{\mathbf{c},\eta|\mathbf{x}}[u(\mathbf{c}, \mathbf{w})]$

Regret: $R_T = \mathbb{E}_{x,c,\eta}\left[\sum_{t=1}^{T}\left(u(\mathbf{c}^t, \mathbf{w}^t) - u(\mathbf{c}^t, \pi(\mathbf{x}^t))\right)\right]$

PROOF: PRedict-then-Optimize with Optimism in Face of uncertainty

Algorithm 2: PROOF: Predict-then-optimize with optimism in face of uncertainty

Initialize:
  Find a barycentric spanner $b_1, \dots, b_d$ for W
  Set $A_i^1 = \sum_{j=1}^{d} b_j b_j^{\dagger}$ and $\hat{\mu}_i^1 = 0$ for all $i = 1, 2, \dots, n$

Receive initial dataset $\mathcal{D} = \{(x_i^0, c_i^0; w_i^0)_{i=1,\dots,n}\}$ from distribution D on (X, C).

for $t = 1, 2, \dots, T$ **do**

  Train the ML model & use it to make a prediction

  Set the confidence radius for UCB

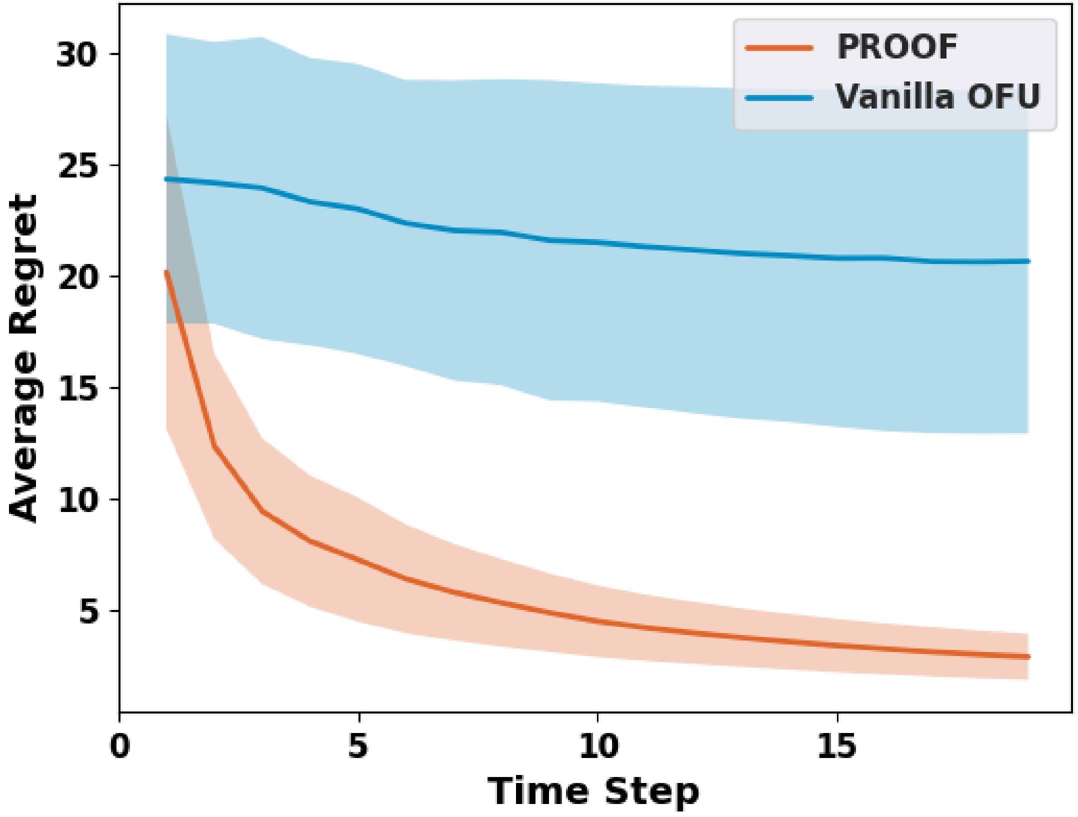
  Select action by integrating UCB with offline ML model

  Receive the true labels and cost

  Update the bandit cost estimate

Theorem. Assuming ordinary least squares regression, the PROOF algorithm has regret $\tilde{O}\left(n\sqrt{dmT}\right)$ with probability $1 - \delta$.

Numerical simulations



PROOF converges

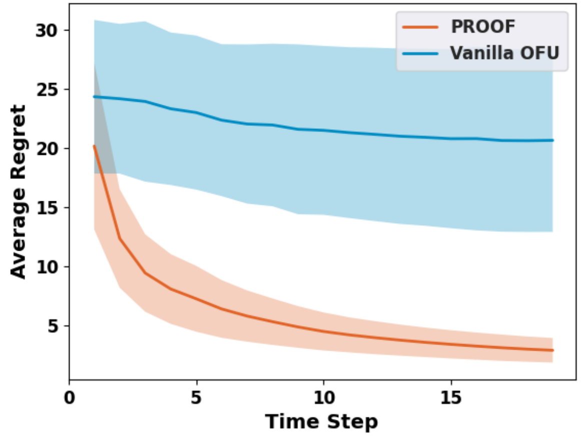
- much faster, and
- with smaller variance

than vanilla bandit.

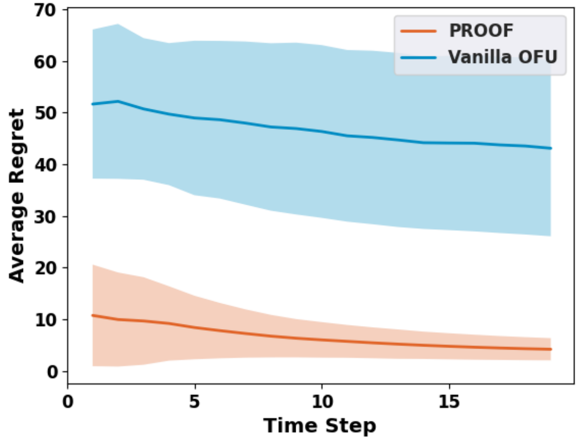
Numerical simulations

PROOF outperforms vanilla bandit in both convergence speed and variance.

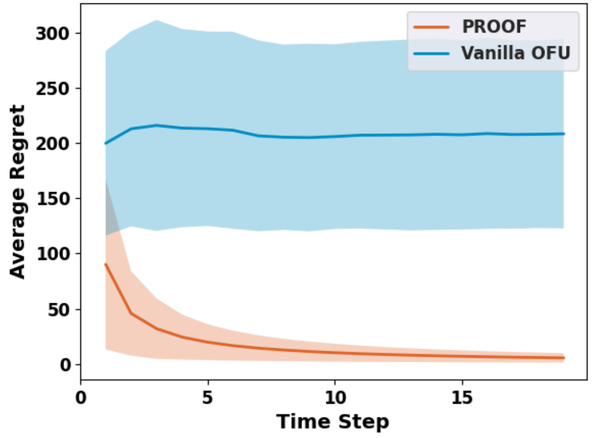
Small scale base case



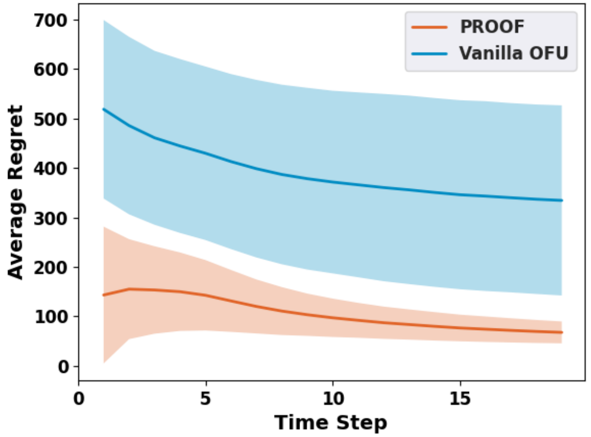
Data/step increased from 20 to 40



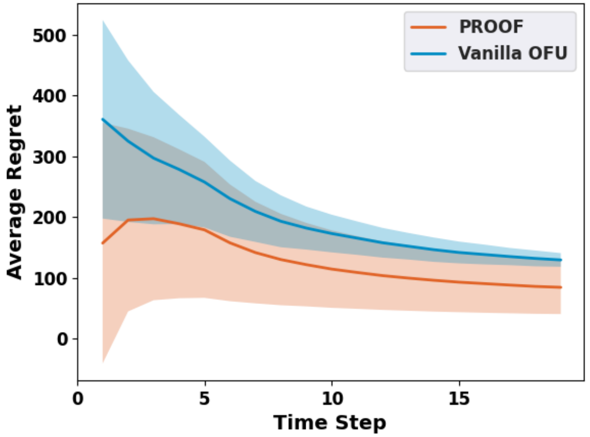
Linear mapping norm multiplied by 10



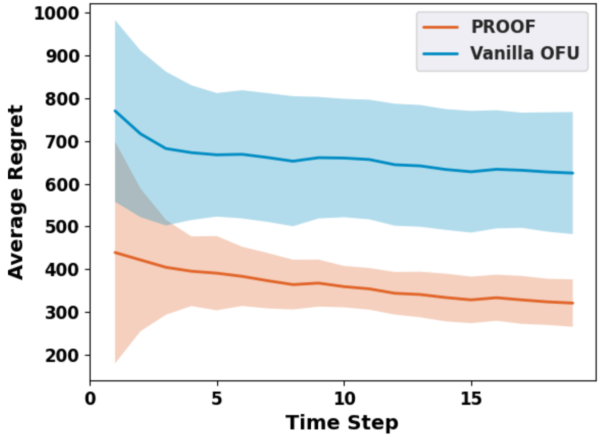
Large scale base case



Linear mapping norm divided by 10



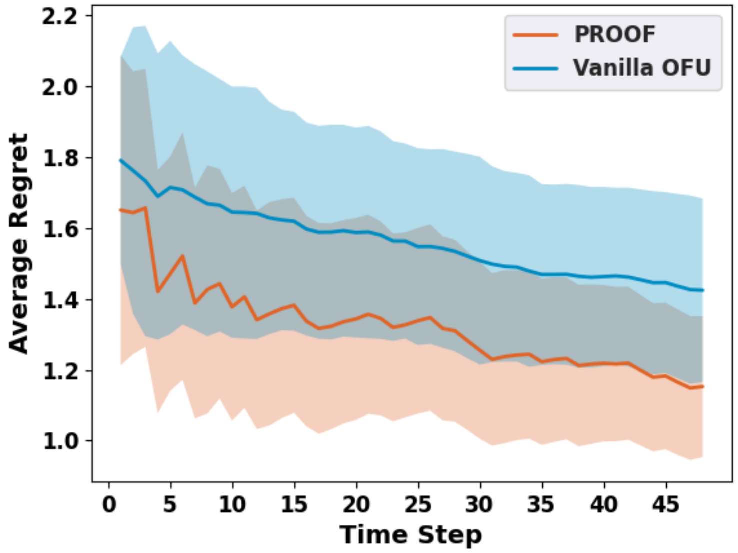
Data noise multiplied by 5



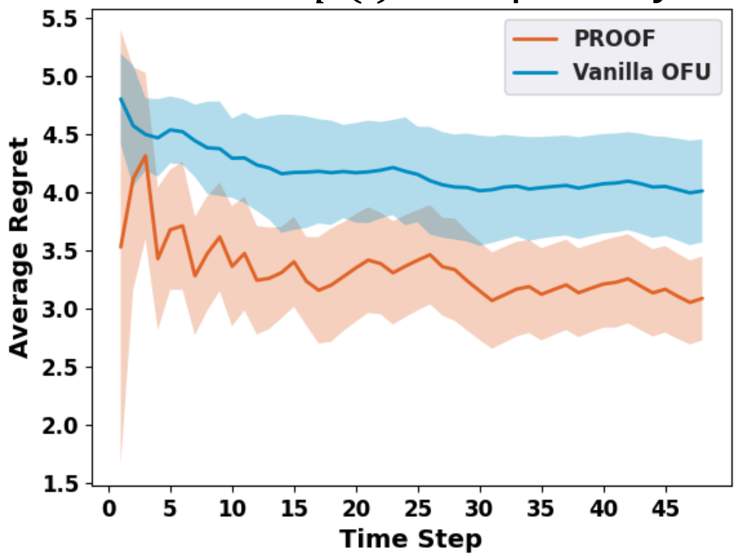
PROOF for food rescue volunteer recommendation

Feature x	Volunteer-rescue pair features
Label $c \in \{0,1\}^d$	whether volunteer claimed the rescue
Action $w \in \{0,1\}^d$	whether to send push notifications to each volunteer
Known cost $p(c, w)$	whether we send push notifications to the “right” volunteer
Unknown cost $q(w)$	how volunteers might react to notifications

Base case



Known cost $p(\cdot)$ multiplied by 4



PROOF can be seen as a middle ground between one-shot ML and online bandit.

One-shot recommendation

YouTube

[Covington et al., RecSys-16]

PROOF recommendation

AD SPACE FOR LEASE

[Shi et al., AAAI-22]

Bandit recommendation

yahoo!

[Li et al., WWW-10]

Compared to one-shot recommendation, PROOF allows for

- Iterative exploration of user's reactions to recs
- Principled model improvement by collecting new data
- All four motivation questions raised earlier

Cost (known)

Cost (unknown)

Cost (known)

Cost (unknown)

PROOF can be seen as a middle ground between one-shot ML and online bandit.

One-shot recommendation

YouTube

[Covington et al., RecSys-16]

PROOF recommendation

AD SPACE FOR LEASE

[Shi et al., AAAI-22]

Bandit recommendation

yahoo!

[Li et al., WWW-10]

Compared to bandit recommendation, PROOF

- Uses supervised learning to reduce variance in cost estimation
- Leverages historical data to avoid over-exploration, retaining stakeholder's trust, esp. in the early stage

Cost (known)

Cost (unknown)

Cost (known)

Cost (unknown)

Cost (known)

Cost (unknown)