Efficiency and Fairness of Food Rescue Platforms: An Initial Study

Yiwen Yuan[1] , **Kimberly Lo**[1] , **Zheyuan Ryan Shi**[1] , **Leah Lizarondo**[2] and **Fei Fang**[1]
[1]Carnegie Mellon University
[2]412FoodRescue

Abstract

Food waste and food insecurity are two challenges that coexist in many communities. To mitigate the problem, food rescue platforms match excess food with the communities in need, and leverage external volunteers to transport the food. Based on a real world dataset from 412FoodRescue, we analyze how to improve the efficiency and fairness of a food rescue platform. We make the following contributions. (1) We identify that the completion of a food rescue is associated with the seasonality and location of the rescue by analyzing the operational dataset. (2) We train a machine learning model which predicts the completion of a rescue with high accuracy and recall. (3) We develop an online matching algorithm which greatly improves the fairness of the current practice.

1 Introduction

In the United States, over 25% of the food is wasted, with an average American wasting about one pound of food per day [Conrad *et al.*, 2018]. Meanwhile, 11.8% of the American households struggle to secure enough food at some point during the year [Coleman-Jensen *et al.*, 2018]. Among the several responses to this evidently inefficient food distribution, many cities worldwide are seeing a growing number of food rescue organizations (also known as food banks). Food rescue organizations receive edible food from restaurants and groceries ("donors") and distribute it to organizations that serve low-resource communities ("recipients"). These food rescue organizations are an important force to fight against food waste and food insecurity, both of which are included in the United Nations' Sustainable Development Goals [UN, 2015].

A food rescue organization functions as a platform between the donors and the recipients. Upon receiving the notice from a donor, the food rescue organization matches the food to a recipient. Typically, it transports the food from the donor to the recipient, and if a recipient is not immediately identified, it stores the food at its own facility. Obviously, this incurs cost and there are existing works on optimizing the matching process to minimize this cost [Nair *et al.*, 2018; Phillips *et al.*, 2011], and some even attempt to create a market [Prendergast, 2016]. However, many of these organizations operate under tight budget and human resource constraints. As a result, some outsource the transportation of food to local volunteers, which brings in a new dimension to the problem.

We collaborate with 412FoodRescue[1], a food rescue organization serving over 500 donors and 500 recipient organizations in Pittsburgh, US. Upon receiving a notice of donation, the dispatcher at 412FoodRescue first matches the donation to a recipient. Then, the dispatcher posts the rescue on its app available on iOS and Android. If a volunteer claims the rescue on the app, she will be provided with detailed instructions to transport the food from the donor to the recipient. To date there have been over 1500 registered volunteers. While involving volunteers saves some cost, it leads to more uncertainty about whether a rescue trip will be successfully completed. Thus, we tackle the following two problems in this paper. (1) We study whether we can predict when a rescue will be missed. (2) Knowing the likelihood of the completion of a rescue, we ask whether we can design more informed matching mechanisms that are more fair than the current manual procedure.

We make the following contributions. (1) We analyze the operational data from 412FoodRescue and found that there is some seasonality and location effect on the success of a rescue. We also found that most volunteers who stick with the program tend to decide so after their first five or less rescues. (2) We train several machine learning models to predict whether a given rescue will be missed. We overcome the imbalanced labels in our dataset using random oversampling and our bagging ensemble with Ivotes model achieves a recall of 0.897. (3) Using the machine learning model, we develop an online matching algorithm which improves the fairness of the existing matching mechanism by significantly increasing the allocation to the recipient organization which receives the least donation.

2 Data Description and Analysis

2.1 Food Rescue Data

The dataset contains the 20536 food rescues in the year 2018 from January 1st to August 25th at 412FoodRescue, involv-

[1]https://412foodrescue.org/

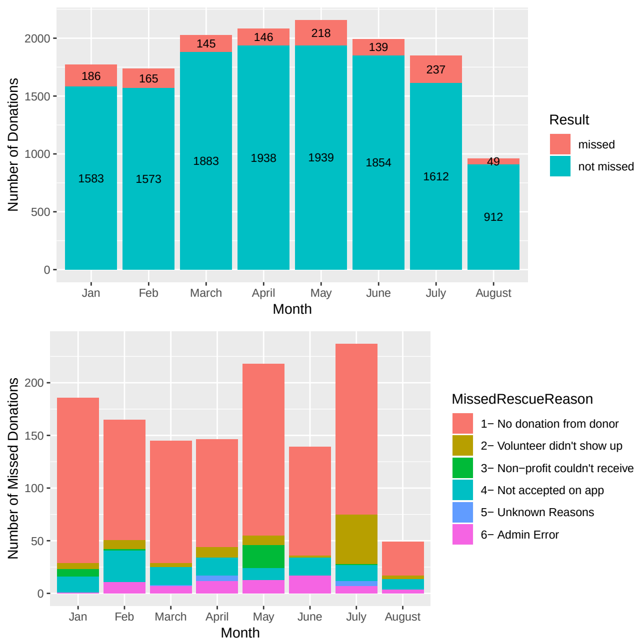

Figure 1: The rate of missed regular rescues and their reasons.

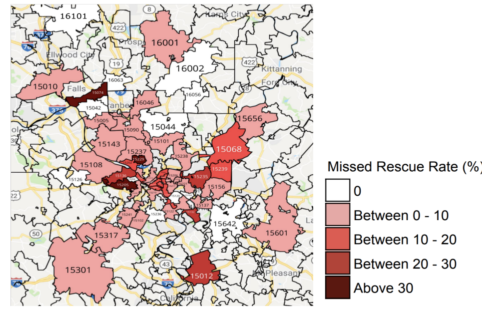

Figure 2: Spatial variation of missed rescue rate. Regions with darker color have higher missed rescue rate.

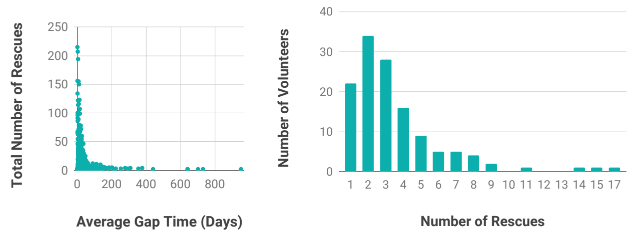

Figure 3: (left) Distribution of volunteers' rescues and average gap time. (right) Number of rescues until conversion to regular rescue

ing 537 donor organizations, 541 recipients organizations, and 1515 registered volunteers. Among these 20536 rescues, 5569 are transported by the employees of 412FoodRescue instead of volunteers. We drop these data points in all of our study. The remaining 14967 rescues are handled by the volunteers, of which 387 are ad-hoc rescues and the rest are regular rescues. Regular rescues are prescheduled recurring donations, while ad-hoc rescues are created in real-time. We analyze them separately.

For each rescue, the dataset contains the identity of the donor, recipient, and volunteers, and the type and weight of the donation. If a rescue is missed, the reason is also included.

Among the 387 ad-hoc rescues, only 4 are missed: two because of no donation from the donor and the other two due to no volunteer claimed them on the app. We proceed to analyze the data from the 14580 regular rescues. A slight difference in the missed rescue rate across the year can be observed from Figure 1. Generally, hotter and colder months have a higher missed rescue rate. This implies that weather can be correlated to the outcome of the rescues. The most common reason for a rescue to be missed is no donation from donor [2] .

Furthermore, we analyze the missed rescues in different regions. This is because 412FoodRescue serves the large area of Allegheny County, and the location of a donor might be a major reason for whether a volunteer claims a rescue or not. We cluster the donors using zip code, and found that the missed rescue rate is particularly high in the highlighted areas in Figure 2.

[2] Technically, "no donation from donor" does not mean a rescue *failed*: it means a donor who had scheduled regular rescue happened to have no donation this time. We include this as "*missed*" because it is still a dismay to the volunteer who signed up for it.

2.2 Volunteer Data

The volunteer dataset contains the rescue history of each volunteer from 2016 to 2018. 41.87% of the volunteers were one time volunteers. On average, volunteers contributed 9.37 times in 2016, 4.72 times in 2017, and 9.94 times in 2018. The maximum number of rescues completed by a volunteer was 92 rescues in 2016, 73 in 2017, and 215 in 2018.

We are interested in the gap time, which is the number of days between two consecutive rescues for a volunteer. As shown in Figure 3(left), volunteers tend to perform in the range of 1 to 50 rescues with average gap time in the range of 0 to 200 days. The standard deviation of gap times was 63.08 days, indicating a large variation in the average number of days between volunteers' rescues. Out of the volunteers who participated in more than 3 rescues, 32.76% of them became regular volunteers who made 4 consecutive rescues with gap times less than 7 days at some point in their record.

Within this group of regular volunteers, 44.17% of them were consistent on a weekly basis for all rescues: participated in at least one rescue a week starting from their first recorded rescue to their last one. For the rest in this group who were not participating in rescues every week to begin with, most of them converted to weekly rescue after one to five rescues, as shown in Figure 3(right). After 5 consecutive rescues which were not within one week spans of each other, it became much less likely for the volunteer to convert from an ad hoc volunteer to a regular one.

2.3 Mobile App Data

We collect ratings and reviews from Google Reviews, Google Play Store, and Apple Store. On Google Reviews, 412FoodRescue had an average rating of 4.6 over 9 ratings. On Google Play Store and Apple Store, their Food Rescue

Figure 4: Distribution of ratings across platforms

Hero app had an average rating of 3.7 over 27 ratings and an average rating of 4.1 over 12 ratings respectively. Google Play Store has the largest number of written reviews across the three platforms, with more than 1000 downloads. Figure 4 shows that there were exceedingly more reviews with ratings of 5. Users indicated that they were generally very pleased with the app's intent to reduce food waste and many users were happy with their volunteer experience.

3 Prediction

Based on the initial data analysis, we proceed to answer the first of our two questions: predict whether a given rescue will be missed.

The previous section suggests that weather and locations of the donation might be strong indicators of the success of a rescue. Thus, for the feature space of the prediction task, we first consider using the weather information. There are four types of weather information that we use as feature in the paper: precipitation, snowfall, snow depth, and average temperature. We do not use the identity of the donor or recipient as an input feature because a new model would be needed every time a new donor or recipient is added into the system. Instead, we only use their zip codes.

A key challenge in the prediction task is the label imbalance in the data set. There are 1107 missed rescues, and the rest 12077 rescues are not missed.[3] Thus, blindly applying machine learning models such as decision trees could easily cause a bias towards the majority class. To address this issue, we use random oversampling to pre-process our data by duplicating the examples with missed rescues 10 times.

The imbalanced dataset also motivates our choice of classifier, as we may train multiple weak classifiers and take the majority vote among them. We use the idea of Bagging Ensemble with Ivotes [Galar *et al.*, 2012], where the basic idea is to use importance sampling to pick the training data for different classifiers. In the original algorithm, the data is sampled without replacement. However, in our algorithm, we sample with replacement,as it can train some different classifiers for the voting process.

To test our proposed classifier against other baselines, we randomly split the data into training and testing set with a ratio $6 : 4$. We use the decision tree (DT), support vector machine (SVM), XGBoost, logistic regression, and the neural network as baselines. However, both logistic regression and neural network with various choices of architecture do not converge. This is possibly because the one-hot encoding of the location causes the feature space to be sparse, and deep structures typically do not perform well on sparse data.

Table 1 shows the performance of our bagging ensemble and the other baselines. All methods exhibit high accuracy, but since our dataset is highly imbalanced, the accuracy does not tell a complete story. Since our goal is to reduce the total number of missed rescues, we aim at reducing the false negative mistakes, i.e. an actually missed rescue being predicted as not missed. On the other hand, predicting a completed rescue as missed is less of a problem in practice. Thus, the primary interest goes to optimizing the recall of the method. As shown in Table 1, the three baseline methods achieve unsatisfying recall. Using oversampling does improve the recall and F1 score on SVM, as indicated by the 4th row in Table 1. Our proposed bagging ensemble approach shows superior performance than all the baselines. In fact, the superior performance is also consistent when using different weak classifiers in the bagging ensemble.

Method	Accuracy	Precision	Recall	F1 Score
DT	0.907	0.408	0.212	0.279
XGBoost	**0.923**	**0.6**	0.065	0.117
SVM	0.911	0.415	0.109	0.173
SVM (os)	0.884	0.315	0.305	0.310
BE(2SVM/DT)	0.895	0.447	0.889	0.595
BE(3SVM)	0.911	0.482	**0.897**	**0.627**
BE(5SVM)	0.908	0.472	0.806	0.595
BE(3DT/2SVM)	0.806	0.241	0.603	0.345

Table 1: Accuracy, precision, recall, and F1 score of the proposed bagging ensemble method and the baseline models. The models in the parenthesis indicate the weak learners used in the bagging ensemble (BE).

4 Online Matching

In this section, we take on our second question: design a volunteer-aware matching algorithm that is more fair than the current practice. A good prediction of whether a rescue will be missed (the previous section) enables us to design a more efficient algorithm to match the donors with recipients. However, if a small subset of recipients always get the majority of the food, other recipients could be discouraged from participating in the program. Therefore, in this section, we present an online matching algorithm which is more "fair" while only matches donors and recipients if it is likely that some volunteer will claim the rescue. We define our notion of fairness in the following paragraphs.

We treat the donations and recipients as two sets of vertices D and R on a graph. We note that here the "donation" is not to be confused with "donor": a donor may make multiple donations, each of possibly different quantities, within a certain period of time. Since we typically do not care whether a donor is always matched to the same recipient, the object in our matching problem is the donation instead of the donor.

The donations arrive online while the recipients are static. An edge $e_{ij} = (d_i, r_j)$ between a donation d_i and a recip-

[3]There are 1783 rescues where the zip code location of the donor and/or the recipient cannot be obtained. We excluded these data points from the total 14967 data points.

Algorithm 1: Weights Decay and Refresh

```
Initialize w_j = 0 for all recipients r_j ∈ R
while a new donation d_i arrives do
    Calculate the edges E_i between the donor of this
      donation and all recipients using our machine
      learning model.
    if |E_i| = 0 then
        Notify the human dispatcher of this situation.
    else
        Match d_i to recipient r_{j*}, where
          j* ∈ arg max_j (1 − w_j)
        Update w_{j*} = w_{j*} + d_i/c, where d_i is the amount
          of food in the current donation.
    if w_j ≥ 1 for all recipients r_j ∈ R then
        Set w_j = 0 for all r_j ∈ R.
```

ient r_j indicates whether a rescue between them is likely to be completed by a volunteer. To this end, we may use the machine learning model developed in the previous section to predict the existence of the edge. Our problem is related to but different from the well-studied online bipartite matching problem [Karp *et al.*, 1990]. In an online bipartite matching problem, a recipient is removed once it is matched to a donation. Yet in our setting this assumption does not make sense. We suppose a recipient r_j accumulates the allocation A_j, which equals the sum of donations d_i (with an abuse of notation) that are matched to it. We consider a fairness objective to be maximizing the minimum allocation A_j across all recipients $r_j \in R$ [4]. In addition to the amount of allocation, we also consider the frequency of receiving a donation as another criteria, because in practice, the number of times a recipient receives donation has an impact on its perception of the program, and this impact is not necessarily the same as the impact of the allocation amount.

Since our goal is design a mechanism so that, food can be fairly distributed to each recipient in terms of the frequencies they arrive and their quantity, a natural method is to assign each recipient with a donation until its allocation reaches a cap. Furthermore, we want the assignment to be smooth; that is, we want the allocation to each recipient grow uniformly. This intuition leads to our Algorithm 1. The capacity parameter c determines a soft cap of amount of food received by an individual recipient. Once a recipient's allocation exceeds this amount, it has to wait until every other recipient's allocation also exceeds this amount before it can get donations again. This algorithm is generalized from Alg. 2.1 in [Mehta *et al.*, 2015], which is a special case of our Algorithm 1 if we take $c \leq d_{\min}$, the smallest possible quantity of donation.

In Figure 5, we show how Algorithm 1 with different choices of capacity parameter c compares with the current practice. We test c with values $d_{\min}$, $d_{\max}$, the maximum quantity of a donation, and d_{avg}, the average quantity of the donations. The actual assignment exhibits the 80/20 rule.

[4] In fact, the problem is NP-hard even if we had known all the donations ahead of time.

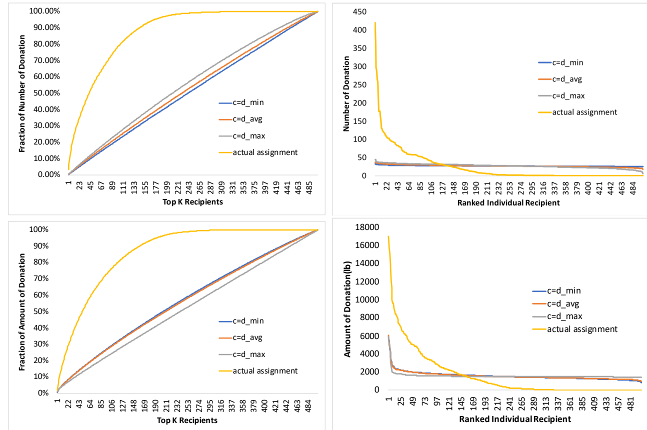

Figure 5: Number of donations and amount of allocations, as the result of Algorithm 1 with different capacity parameters, compared with the actual assignment.

Metrics	$c = d_{\min}$	$c = d_{\text{avg}}$	$c = d_{\max}$	Actual
Min	25	18	6	0
Max	33	37	45	420
Stdev	1.38	3.07	5.79	47.22

Table 2: The minimum, maximum, and standard deviation of the number of donations for different choices of capacity parameter c, taken across all recipients.

That is, it obviously concentrates both the number of donations and the quantity of food on a small subset of recipient organizations. On the other hand, our algorithms achieve a much more even spread, especially in terms of the number of donations. In Table 2, we see that setting $c = d_{\min}$ achieves the best fairness objective in terms of the number of donations. This should not be surprising because with $c = d_{\min}$, Algorithm 1 is effectively assigning donations to recipient one by one, assuming that all donor-recipient pairs are predicted to be successful by the machine learning algorithm. Meanwhile, Table 3 suggests that setting $c = d_{\max}$ is most fair in terms of the quantity of allocation, which is also reasonable, as the smaller value c takes, the more we are ignoring the quantity in a donation. Yet again, all variants of our algorithm are more fair than the actual assignment.

Metrics	$c = d_{\min}$	$c = d_{\text{avg}}$	$c = d_{\max}$	Actual
Min	841.0	981.0	1469.0	0.0
Max	5663.0	6094.0	5936.0	16955.9
Stdev	466.66	448.97	335.45	2578.54

Table 3: The minimum, maximum, and standard deviation of the quantity of allocation for different choices of capacity parameter c, taken across all recipients.

We acknowledge that fairness, or at least the definition of fairness in this paper, is not the only criteria of an allocation mechanism. In practice, a number of objective constraints and human factors could override any fixed algorithm. As our next step, we will work closely with the human dispatchers to design algorithms which take into account as many of these factors as possible.

References

[Coleman-Jensen *et al.*, 2018] Alisha Coleman-Jensen, Matthew P Rabbitt, Christian A Gregory, and Anita Singh. Household food security in the united states in 2017. 2018.

[Conrad *et al.*, 2018] Zach Conrad, Meredith T Niles, Deborah A Neher, Eric D Roy, Nicole E Tichenor, and Lisa Jahns. Relationship between food waste, diet quality, and environmental sustainability. *PloS one*, 13(4):e0195405, 2018.

[Galar *et al.*, 2012] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

[Karp *et al.*, 1990] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[Mehta *et al.*, 2015] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1388–1404. Society for Industrial and Applied Mathematics, 2015.

[Nair *et al.*, 2018] DJ Nair, H Grzybowska, Y Fu, and VV Dixit. Scheduling and routing models for food rescue and delivery operations. *Socio-Economic Planning Sciences*, 63:18–32, 2018.

[Phillips *et al.*, 2011] Caleb Phillips, Rhonda Hoenigman, and Becky Higbee. Food redistribution as optimization. *arXiv preprint arXiv:1108.5768*, 2011.

[Prendergast, 2016] Canice Prendergast. The allocation of food to food banks. *EAI Endorsed Trans. Serious Games*, 3(10):e4, 2016.

[UN, 2015] UN. *17 Goals to Transform Our World*, 2015. https://www.un.org/sustainabledevelopment.