ChariTable

Choose to give with confidence

PREPARED BY

Bonnie McDougal

McKay Matheson

Jeffrey Mohler

Josh Kartchner

Table of Contents

Executive Summary **3**

Project Overview **4**

Success Measures 4

Application Features 5

Technical Breakdown: Website **9**

Front End 9

Back End 9

Technical Breakdown: Data and Data Pipeline **10**

Data Cleaning 10

Data Pipeline 15

Technical Breakdown: Deployment **17**

Technical Breakdown: Security **17**

Recommendations **19**

Using the Predictor Service For New Campaigns 19

Targeting Weak Campaigns 19

Assumptions **21**

EXECUTIVE SUMMARY

This report serves as a project proposal to be presented to the end users of GoFundMe (including both donors and GoFundMe analysts), as well as the faculty of the Marriott School of Business Information Systems program.

GoFundMe is a crowd-sourced charitable campaign platform. Over the past few months, there has been a steep increase in the creation of campaigns relating to the COVID-19 pandemic. The abundance of campaigns relating to COVID-19 has resulted in an increase in attempted fraudulent campaigns, as well as difficulty for well-intending individuals to identify which campaigns will successfully utilize their donated funds. In an effort to combat these issues faced by donors, GoFundMe has provided our team with over 6000 campaign details that include either “COVID19” or “Corona” anywhere in the campaign information.

As requested by GoFundMe, the solution to the pain points currently faced by donors and the campaign organizers will consist of a website with the following characteristics:

- A search page that allows campaigns to be discovered based on a variety of criterion
- A search details page that allows a single campaign to be viewed in greater detail
- A variety of analytics and metrics that differentiate high- and low-quality campaigns
- A prediction calculator that helps donors estimate how successful a hypothetical campaign will be

The website, called ChariTable, will be utilized by two different types of users: GoFundMe campaign organizers and GoFundMe analysts. The intent of each element on the website will be to give users a more informed approach as they select and create campaigns. Also, the website will enable GoFundMe analysts to reach out to users with suggestions that will increase their likelihood of success. It is our hope that ChariTable will help donors to contribute funds to campaigns that have the highest likelihood of helping those impacted by COVID-19.

As part of this project development, the team made several assumptions to help deliver the end product. These assumptions are referenced throughout the report, and a complete list of the assumptions can be found on page 21.

Project Overview
Within this project, we determined how to measure success, and then used that success measure to develop relevant features for campaign creators and GoFundMe administrators and analysts.

Success Measures
As part of our project, we identified measures to use for determining success in campaigns. We applied this measure to both our features for campaign creators as well as GoFundMe administrators and analysts.

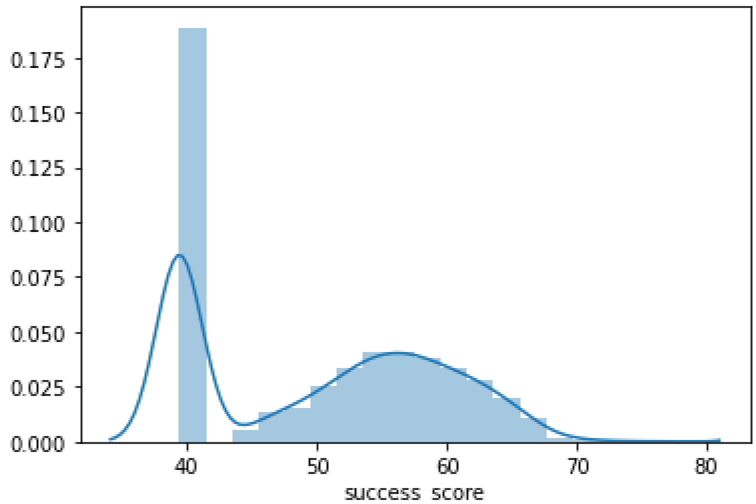
To measure success in our prediction model, we created a calculated field titled success score. This score is the average T-score of two metrics:

- Donation amount per donor
- Percent of goal reached in collections

This score results in a number between 0 and 100, with 50 being an average success score. The following ranges can be used to interpret success score:

Success Score	Meaning
< 35	Poor
35-45	Fair
45-55	Average
55-65	Good
> 65	Excellent

Success scores in our data set are normally distributed except for a massive amount of campaigns that have not raised any money yet resulting in zero percent of goal reached and zero dollars per donation. The histogram below shows the count of success scores in our data set.



The complete process of this calculation can be found in the attached "T-Score.ipynb" file. However, in summary, these are the steps we took to find the success score:

1. Filter down data to only that from 2020 (**See Assumption 1**)
2. Convert current_amount into all USD
3. Calculate and create an amount-per-donor column (current_amount_USD/donators)
4. Calculate and create a percent_of_goal column (current_amount/goal)
5. Adjust both new columns for outliers and skewness
6. Calculate the T-score of both columns
7. Take the average of the two T-scores to arrive at a final success score

This methodology was applied in both the database and the data pipeline.

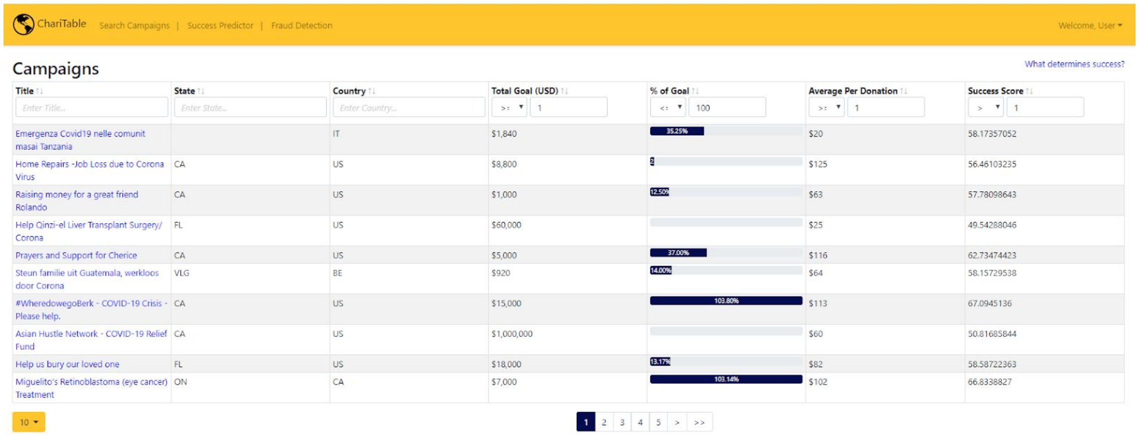
Application Features

For campaign organizers, we have created a search feature to view other COVID-19 campaigns as well as a predictor page to experiment with features within a campaign and predict the success of a campaign. Additionally, we have provided some static pages where campaign creators can learn about the success score and ways to boost their campaign. Currently, all pages and features are accessible for any user, but with further implementation of this project, we would add authentication to the application and have select features only available for specific users. (**See Assumption 2**)

Search Feature

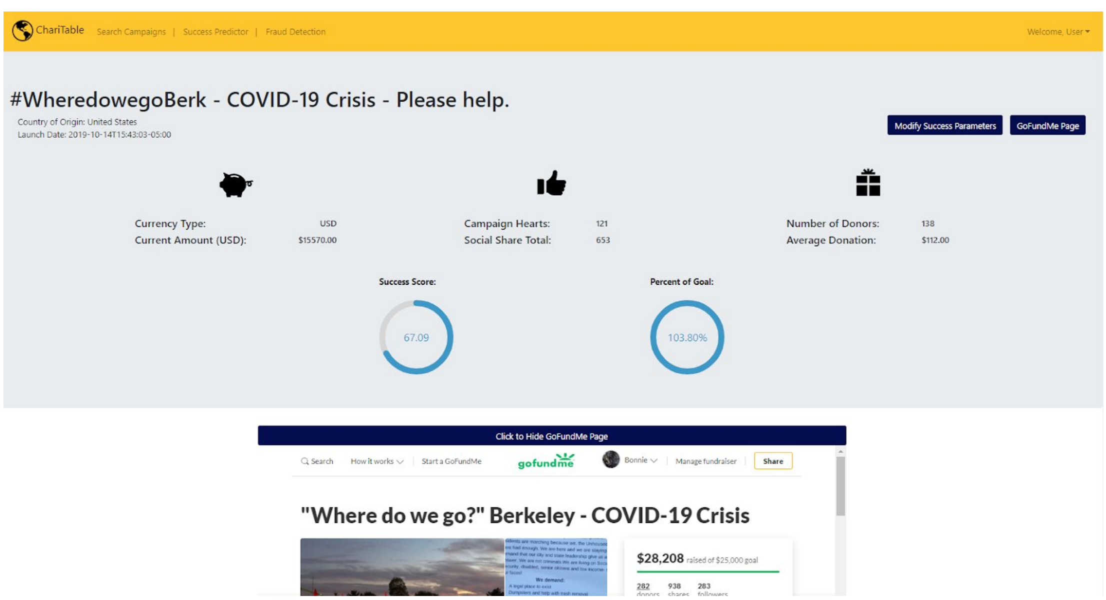
A search feature is available for campaign creators to search other COVID-19 - related campaigns. Campaign creators can use this search feature to filter down to certain campaigns and see the success score given for these campaigns.

Analysts can also use this page to filter down to campaigns that currently have a low success score and then find ways to help campaign creators boost their success score. This will also help increase campaign creators’ satisfaction, boost business for GoFundMe, and aid more individuals who have been negatively impacted by COVID-19. The screenshot below shows the details page.



From the search page, users can select a link to see details, statistics, and metrics about the specific campaign. This includes the embedded GoFundMe page to see what the current GoFundMe page looks like as well as a link to access the actual GoFundMe page for the campaign.

The campaign details page also has a link for analysts to use that takes them to a campaign edit page. This feature is explained more further on in the report. A screenshot of the campaign details page is provided below.



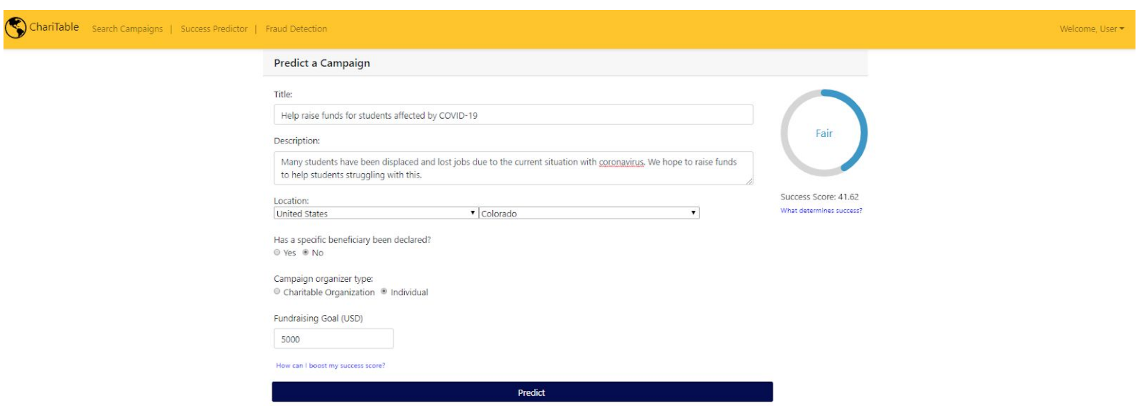
Campaign Success Prediction Page

In the campaign success prediction page, campaign creators can enter in the following features to predict campaign success:

- Title
- Description
- Country
- State
- Presence of an involved beneficiary
- Charity organization involvement
- Fundraising goal

After inputting these fields and submitting the features, this page generates a success score. The page also includes links to the success score page and boost camp page, so campaign creators can learn more about what the success score means and how to increase the score.

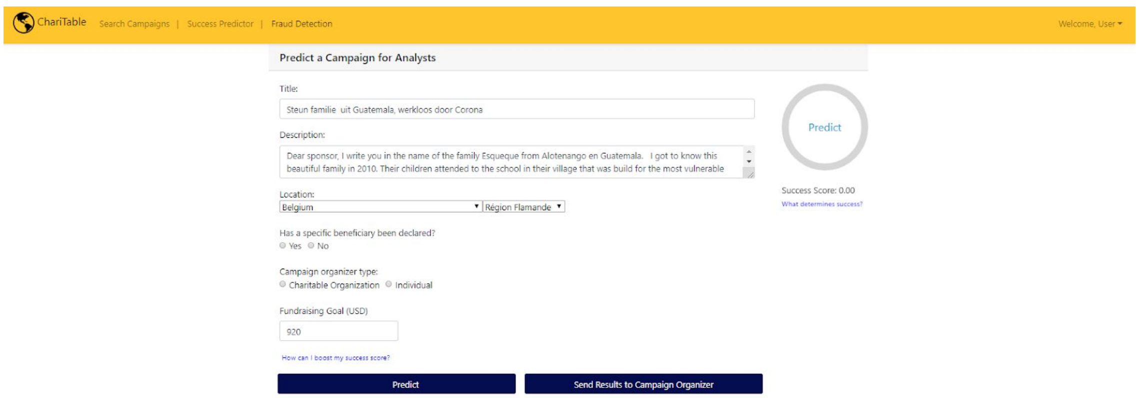
A screenshot of the predictor page is provided below.



Campaign Editor Page

The campaign editor page is intended only for administrators and analysts at GoFundMe. This page is like the campaign success prediction page, but with a few additional features.

The campaign editor page is accessed through the campaign details page and all fields of the page are populated with the data from the campaign details page. Analysts and administrators can edit these fields and re-calculate the success score. Then, they can opt to send an auto-generated email to the campaign creator that invites campaign creators to try the success predictor page for campaign boosting. See the screenshot below.



Success Score Page

This page was designed to help campaign creators to understand the makeup and meaning of the success score. It includes components of the success score, appropriate ranges of the success score, as well as a brief explanation of the methodology behind success score.

Boost Camp Page

The boost camp page gives general guidelines for campaign creators to boost the success score of their campaign. It explains the value of changing values one at a time and also gives hints for certain “buzz words” that influence successful COVID-19 campaigns.

We found the buzz words by exporting the topic feature matrix csv from the Azure Machine Learning Model (see Technical Breakdown: Data and Data Pipeline). We uploaded the csv in python and found the top 25 words and phrases for each topic in order of feature importance. This code is available in the buzzWords.ipynb file.

Fraud Detection Page

This page is designed to provide each individual who visits the website with a few different tips and recommendations to avoid fraudulent or ill-intentioned campaigns. All of the content on this page was provided directly from the GoFundMe website.

Technical Breakdown: Website

The website for this application was built with a front-end React application and a back-end Django Rest Framework API.

Front End

With React.js at the forefront of interactive web technologies, we built out the aforementioned application features on a React app that serves up content from a Django backend (see following section). We designed this website to be interactive, intuitive, and insightful for all end users. Because campaign users come from a wide variety of backgrounds, they may not interpret all metrics at a level of understanding like seasoned analysts, so we opted to use visual representations for metrics throughout the application.

Extensive use of React Bootstrap libraries enabled us to expand the manipulation of data through progress bars and circles, collapsible embedded pages, paginated tables, and vibrant colors. Functionality to view campaigns and predictions for both creators and analysts is condensed into a minimal amount of pages to reduce potential navigation errors.

Back End

One of our primary goals in configuring the back end of the website was to make the maximum amount of data from the provided datasets available for use on the front end of the website. We determined that the best manner to accomplish this goal would be to pull all data from a Django Rest Framework API.

After cleaning the data, each CSV file (campaigns, donations, updates) was converted from a CSV file to a JSON file. Each JSON file was then added to the project. After much discussion and experimentation with databases, we determined that the easiest method of storing the records for this particular case would be to use the default SQLite database. Our initial fear in utilizing SQLite was that it would not support datasets as large as the ones provided in the case and that the client would be overwhelmed when attempting to display the data. As a solution to this issue of an overloaded client, the front end is configured in a way that the client never displays more than 100 campaigns at a time.

Next all of the data from the three cleaned datasets was uploaded to the SQLite database file stored in the project. Due to the fact that some of the data in the provided CSV files were not formatted correctly for use with SQLite, all data was stored as text data types. Anytime the client calls for info on the API, the data is converted to its appropriate format (date, integer, etc) and displayed to the end user in it’s appropriate format.

See Assumption 3

Technical Breakdown: Data and Data Pipeline

We used the .csv files provided by GoFundMe for both the prediction feature and the search feature in our website. Within this project, we cleaned the data using python and analyzed the data using Azure MachineLearning Studio.

We cleaned the data differently for the database used on the website and the data used in the predictor model. The universal changes described in the next section were applied to both data sets, but all other changes explained were applied only to the data in the predictor model. We kept the data for the website database in its original form so that search results represented the original data.

Data Cleaning

Each .csv data set was cleaned individually according to its individual attributes. However, two changes were made universally across all three data sets.

For all three data sets, we only pulled data from the year 2020 in order to have a recent data set to work with. We also deleted any rows from the data set that were not related to COVID19 in order to maintain consistency in the coronavirus sample. In other words, any columns that’s organizer’s name contained “corona,” but was not related to COVID19 was removed.

Data changes for each individual data set are addressed in the next three sections.

Campaigns

The table on the next page shows the deleted columns from the campaigns set with the justification for deletion:

Column	Justification
user_facebook_id	High majority of null values
user_profile_id	High majority of null values
social_share_last_update	High majority of null values
url	Not relevant for prediction
default_url	Not relevant for prediction
campaign_image_url	Not relevant for prediction
user_profile_url	Not relevant for prediction
user_first_name	Not relevant for prediction
user_last_name	Not relevant for prediction
collected_date	Not relevant for prediction
velocity	Only held one value
turn_off_donations	Majority only had one value
deactivated	Majority only had one value
status	Majority only had one value
media_type	Majority only had one value
project_type	Majority only had one value
category	Majority only had one value
state	Majority only had one value
is_launched	Majority only had one value
user_first_name	Not relevant for prediction
user_last_name	Not relevant for prediction

We also removed any line breaks in the description column to help with our text analytics in Azure MachineLearning Studio.

We adjusted the following columns for skewness:

- Donators
- Campaign_hearts
- Social_share_total
- Goal
- Current_amount
- Days_created
- Days_active

After correcting for skewness, we ran all numeric columns through a loop to replace outliers with the theoretical max and min.

For more details of the data cleaning of the campaigns csv, see the attached “Intex Campaigns.ipynb” file.

Donations

The table below shows the deleted columns from the donations set with the justification for deletion:

Column	Justification
Unnamed: 0	This column is an index counter, not relevant to prediction
collected_date	The date of the collection of the data is not relevant to prediction
is_offline	We don’t understand what this column contains. For now, we will delete from our data set for analysis
profile_url	A profile_url is an identifier, but not necessarily correlated with our label
verified	We don’t understand what this column contains. For now, we will delete from our data set for analysis
name	Name is an identifier, but not necessarily correlated with our label

The donations data set contained one column with significant outliers: donation amount.

Based on our evaluation in the donations.ipynb file, we adjusted 2506 outlier values to be the theoretical max. This also reduced the skewness level to 0.088, so that we can work with this column for prediction.

No other adjustments were made to columns within the donations data set.

Updates

The table below shows the deleted columns from the updates set with the justification for deletion:

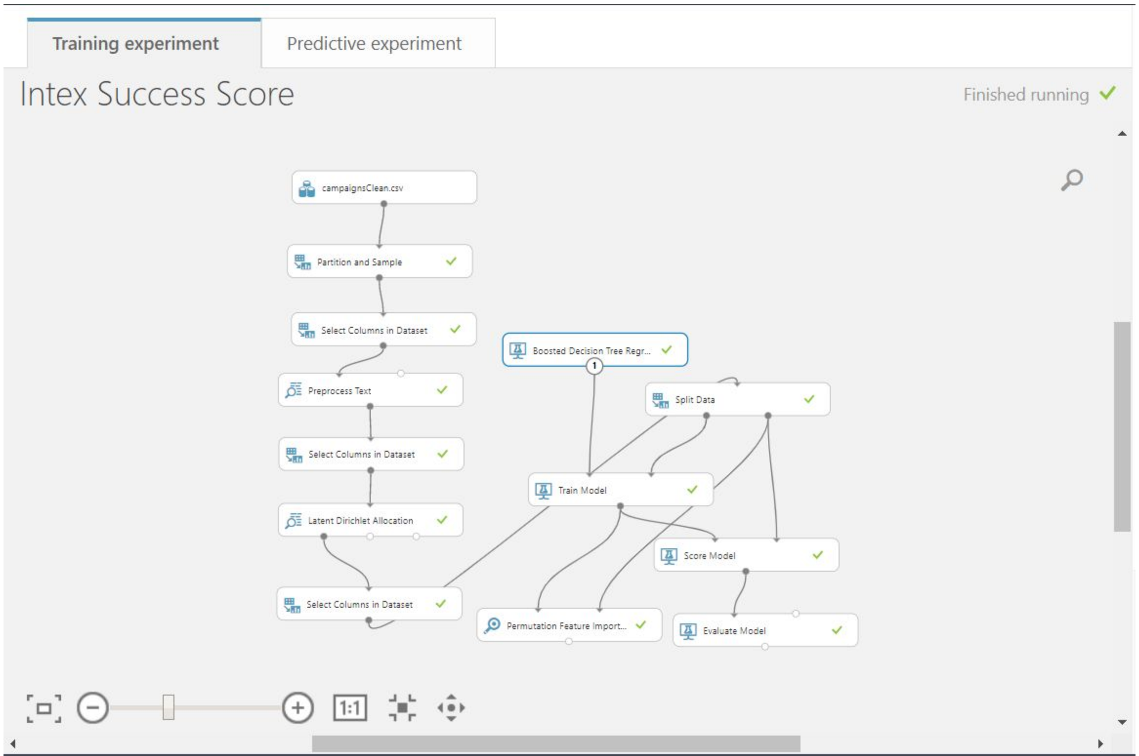
Column	Justification
Column1	This column keeps track of row indexes. Not relevant for prediction
comments	This column only contained one value: “[]”. This is irrelevant for prediction
photo_url	This is the url for the photo used for an update. Not relevant for prediction
collected_date	The date of the collection of the data is not relevant to prediction
created_at	This is the date that the update was created. While we understand that people may be more inclined to donate on certain dates, many event related to COVID19 are unexpected, and because of this we consider this data to be irrelevant
updates_author	This is a name, which functions as an identifier, and is not relevant to prediction

As part of this data cleaning, we took line breaks out of the updates_text field to help with text analytics.

The updates_text column had 2066 blank values. Due to the small number of null values (there were 39,434 total rows in the dataset), we chose to delete the rows that contained blank values.

Data Pipeline

We built a data pipeline in AzureML Studio with the objective of predicting success score. A screenshot of our experiment model is shown below.



We chose n-grams, topics, the regression model, and features on a trial-and-error basis with the goal of a high r-squared value and a low Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

In the Latent Dirichlet Allocation (LDA) pill, we selected 5 n-grams and 5 topics.

In the boosted decision tree, we selected the following parameters:

Parameter	**Amount**
Maximum number of leaves	20
Minimum number of leaves	1
Learning Rate	0.1
Total number of trees constructed	125

We selected the following features for our model:

- Topic4
- Goal_lnplus (the goal column adjusted for skewness)
- Has_beneficiary
- Topic1
- Location_country
- Location_state
- Is_charity
- Topic2
- Topic5
- Title
- Topic3

After completing this model, we had the following metrics:

Metric	Value
R-Squared	0.247
Root Mean Squared Error	8.143
Mean Absolute Error	6.505
Relative Absolute Error	0.764
Relative Squared Error	0.753

Predictive Experiment

For running our predictive experiment, we inserted the web service input pill before the preprocess text. By placing the web service input in this location, campaign descriptions can be text-analyzed to help predict the success score.

Technical Breakdown: Deployment

In addition to creating the predictive experiment, we deployed the Azure experiment as a web service and connected it to our web application. In this process, we tested the service in an Excel workbook and in Postman to ensure the Azure API would return a valid value. With security in mind, our team chose to only put API requests to our own API on the frontend of our website. This enables our server to handle any external API requests. Accordingly, sample python code, with a few changes, was then taken from Postman and placed into a new post method in the views.py file of our Django server. This post method was called in the “on submit” method of the prediction button on the front end. With this, an end user can run our deployed Azure experiment by simply filling in a couple inputs and clicking a button on our webpage.

Next, our team deployed the whole web application which enables any end user to access the ChariTable web page by typing in a url. Due to the simplicity and low cost, we chose to deploy using Heroku cloud platform services. Heroku is a free service that enables users to deploy various types of web applications. As stated above, the ChariTable website consists of a ReactJS frontend and a Django Python backend. The deployment process consisted of various steps, including creating a new heroku app, installing a few new packages, and adding the built React project as a new app in the Django project. Several changes were also made to the configuration of the files. The whole project was then pushed up to the secure Heroku cloud using the Heroku Git CLI feature.

Technical Breakdown: Security

ChariTable handles a large amount of personal user information. As such, security is a critical aspect in the implementation of this solution developed for GoFundMe. In planning this project, we have implemented and/or planned for various cybersecurity measures that will ensure the protection of all user data as well as GoFundMe data. These preventative measures include user logins to view certain data, avoidance of cross site scripting (xss), avoidance of SQL injection (SQLi), data confidentiality, secure HTML, and database security.

- **User Logins:** In an effort to ensure that private data is accessed only by those who should be accessing it, the production version of the ChariTable website will have two different end user experiences. **See Assumption 2**
    - Experience 1: The analyst/administrator access gives the user access to all pages on the website, including the ability to determine which features would enable campaign organizers to have a more successful campaign.
    - Experience 2: The campaign organizers would have a more limited experience while accessing ChariTable. After logging in, the user would have access only to their own campaign information.
- **Cross Site Scripting:** In an effort to avoid cross site scripting (XSS), our team decided to build the ChariTable website using a React web framework. The React

web framework prevents many types of XSS attacks. However, in future production, the team would need to look further into XSS through HTML <a> tags.

- **SQL Injection:** In an effort to avoid SQL injections on the ChariTable website, we have decided to utilize a Django that references all database fields, as opposed to including SQL queries directly in the project code. Django then generates the queries using parameterized SQL, which prevents SQL injection.
- **Confidentiality:** All code that project team members collaborated on was handled in a confidential manner. All Github repositories, Azure Machine Learning pipelines and Python data cleaning code was privately published. Only members of the project team had access to these different repositories during the creation of ChariTable.
- **HTTPS:** ChariTable is hosted using HTTPS rather than HTTP. The main benefit of using HTTPS is that the connection is secure. This means that HTTPS encrypts all HTTP requests and responses using Transport Layer Security protocol.
- **Database Security: See Assumption 3**

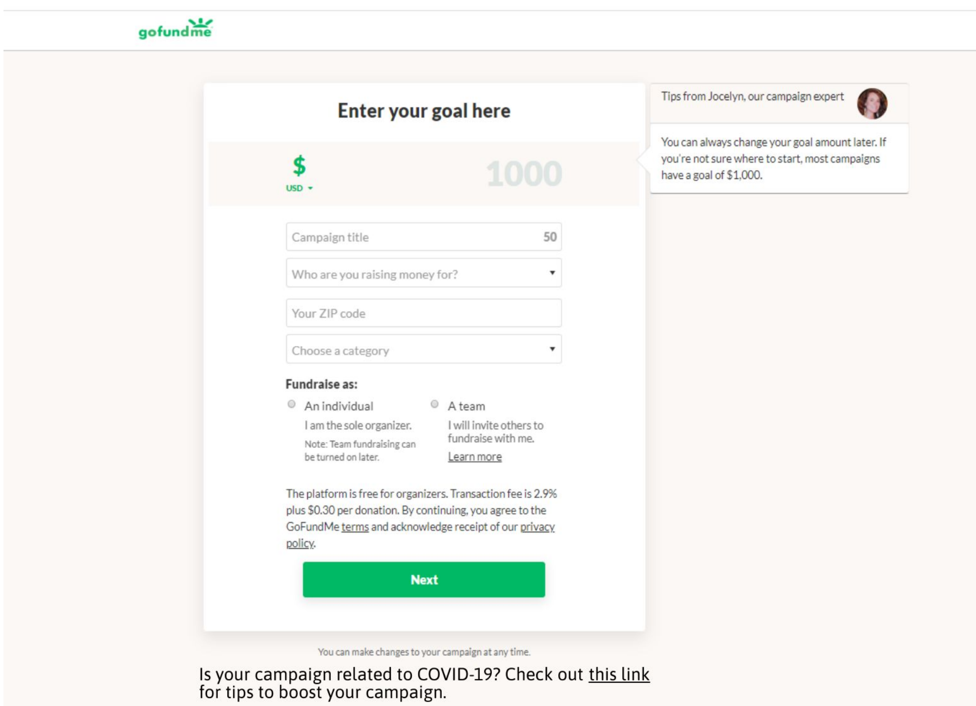
Recommendations

We recommend that GoFundMe use this application to help boost campaigns related to COVID-19. GoFundMe can do that by

- Recommending the predictor service to campaign organizers when they start creating a campaign
- Targeting current campaigns with a low success score to make specific change recommendations for campaign organizers

Using the Predictor Service For New Campaigns

We recommend temporarily embedding a link on GoFundMe's current campaign creation page that links to this service so that campaign organizers can create more compelling campaigns from the beginning. As an example, GoFundMe could edit their page at https://www.gofundme.com/create/details to look like this:



The link at the bottom of the page would link to the campaign predictor page provided in the application.

Targeting Weak Campaigns

Analysts should use this site to search for campaigns with an unusually low success score. They can then reach out to these campaign organizers and recommend changes as well as use of the predictor feature. This includes using the automated email feature, which notifies campaign organizers of specific change recommendations to improve campaign success.

Assumptions

1. All cleaned data referenced in this project is data from 2020. While there were cases recorded in 2019, the New York Times reports that the virus did not grow to a global pandemic with widespread impacts until the start of 2020.

   Refer to https://www.nytimes.com/article/coronavirus-timeline.html for a more detailed explanation of the timeline of events in relation to the coronavirus pandemic.

2. Our current working prototype does not have any form of user authentication. Should GoFundMe choose to implement our proposed solution, it would be extremely important to include proper authentication techniques seeing as this website will be accessed by the general public, as well as GoFundMe analysts and administrators. As such, the website would include login functionality for each of these unique rolls.

3. A more substantial, secure database would be used when ChariTable is used by GoFundMe in a production environment. All data in this case was gathered using a one-time web scrape by GoFundMe, suggesting that data would need to be constantly collected and stored. Large quantities of data would be referenced, requiring a larger database than simply SQLite.

4. The data provided by GoFundMe does not include any form of email contact for donors or campaign organizers. The data includes a facebook ID number, as well as a profile URL, but a majority of instances in the data do not provide both. When ChariTable is used by GoFundMe in a production environment, it would need some form of email contact. This case assumes that all users would provide an email. With that being said, the current emailing feature provided in ChariTable is somewhat limited in its capabilities. A production version of ChariTable would need a more robust version of the feature that would allow for emails containing more content and details for the campaign organizers.

5. This prototype version of ChariTable does not incorporate any form of user authentication. However, for security reasons, it would be crucial to implement authentication for all users who access data on the website. As explained in the Security Section of this report, the campaign organizers and the GoFundMe employees would need to be authenticated at login and would have different user experiences while visiting the website.
6. The ChariTable website references the GoFundMe website in a few different ways, including embedded HTML pages and links. If the GoFundMe website is ever under maintenance, the Charitable website will continue working but certain features will not be available.