**GO*F*UNDME : Report of work**

Alban d’Hauthuille

Thomas Lance

Victor Bakache

Manon Krook

Hannes Marks

Ulrich Laitenberger

**INTRODUCTION:**

We decided to look into a crowdfunding company called GoFundme. This platform aims to allow anyone who wants to publish a project and raise funds. We therefore want to collect different data that would allow us to understand which factors are responsible for the success or failure of a project.

This platform is in the form of a web site whose url is : <https://www.gofundme.com/fr-fr>. It allows individuals or private companies to publish a project and call for donations.  To collect the features of a collect : number of donors, creation date, description, category, amount collected, amount targeted, town, title, etc.., we used Selenium to navigate on the website and then a parser, BeautifulSoup, to get theses features from the html code of each page. To analyse the data, we use matplotlib to have a first look, matrixes of correlation and mutliples OLS regression (statmodels) to analyze the influence of the differents features on the amount collected.

We found several conclusions, some obvious like the greater the mean donation of a collect is, the more successful is the collect; but others are surprising, for example the duration of a collect is not that much correlated to its success. But most and foremost, the correlation between the presence of ‘keywords’ in the description is very important !

**Data collection strategy:**

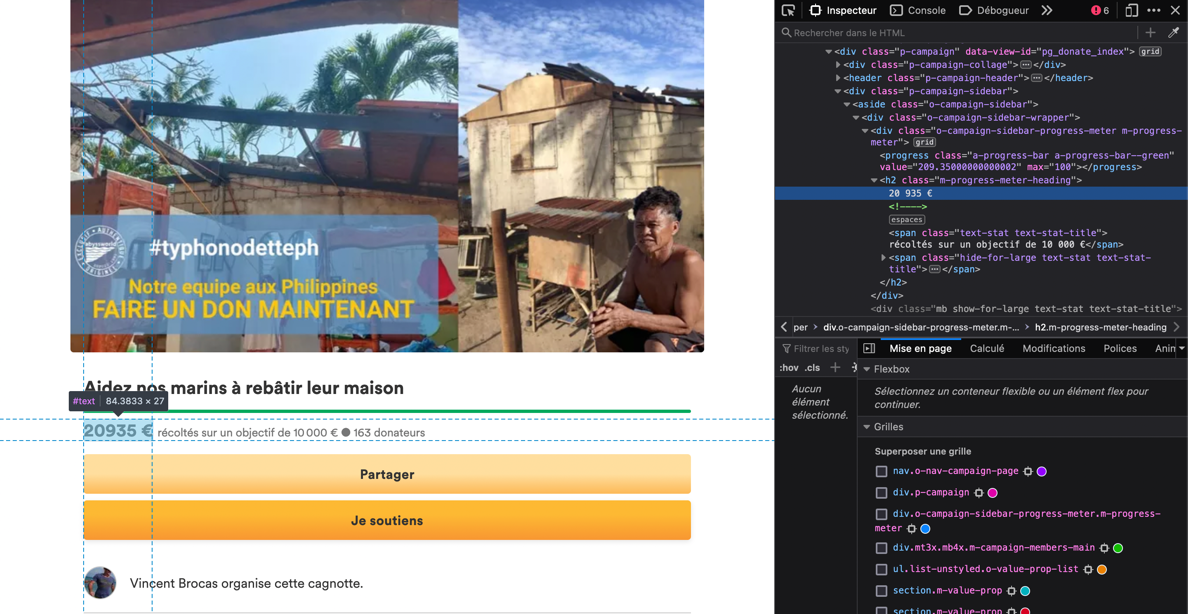
As planned in step 1, we had to go to the page of each collection and for that, we had to use Selenium to navigate and BeautifulSoup to retrieve the different characteristics. As blockages, we didn't have any capcha to deal with but if we didn't put a time.wait(), the loops failed. Also, it was sometimes complicated to locate exactly the tag of such or such figure displayed on the screen (*Figure3)*, so we had to try inspecting the code source of the webpages. For the number of donors for example, we had to try with Selenium and then BeautifulSoup, using first the class, then the XPATH and then the selector or the CSS path. Sometimes projects have no donors for example and so the tag linked to the number of donors does not exist, we had to use ‘Try’ loops. 

Figure 1 Finding the exact tag

Moreover, the webscraping program was very long to run because of the navigation on the different collection categories and then on the page of each collection, about 10-15  minutes with fiber connection and powerful computers. That is why we decided to collect 96 collections per category, i.e. 18\*96=1728 collections in total. We had 1200 at the beginning but we decided to let the program run longer to get more. To discuss 'How representative is our sample?' The answer is that we took the collections that would correspond to a person browsing the homepage of each category and looking at the most popular collections displayed by the platform. This means that they are the ones that a normal person would look at, without for example using a solidarity link that points to a specific collection. This being the case, such a collection would surely be displayed in the popular collections. *Figure1* belowshows our first DataFrame before cleaning.

Une image contenant table

Description générée automatiquement

Figure 2: The Data freshly scraped

After this scraping work, we had a lot of cleaning to do in preparation for the analysis, so we decided to create a notebook 'II' just for cleaning. First we checked that each column corresponded to the right row, for example the right amount associated to the right collect, we were sure of this thanks to our 'for' loop which scrapes the data as it returns 'None' if the tag does not exist in the html parser. It consisted of several steps: retrieving the data that was of type 'objects' in the DataFrame to transform it into strings or floats. We often had to tokenize the strings, remove the ., €, %, :, transform the string 1k into 1000 and 1M into 1 000 000 in the columns of 'amount collected', 'description', 'city' etc. We had to create a program that transformed the sentence 'Created 2 days ago' into a correct DateTime in the format 'MM-DD-YYYY' and that calculates the difference with today to get the duration of the collection (for OLS and plots). We also transformed the string column 'Description' into a tokenized word list column of length greater than 4 to avoid 'the', 'and 'an', etc which are not interesting for the word study. Last but not least, we removed the 10% highest values of each category on the 'Amount targeted' feature because there were false collections which distorted all the analyses, for example: "Give me 1B€ to buy watches"...

Finally, we had the Data Frame of the *Figure3*, ready for the analyze in the notebook III.

Une image contenant table

Description générée automatiquement

Figure 3 Data Cleaned

**Data description**

Une image contenant table

Description générée automatiquement

Figure 4 Statistics

The global statistic are presented in the Figure4.

We will work on these features: ‘MeanDonation’, ‘NumberDonors’, ‘Amount targeted’ ‘Duration in days’, ‘Amount targeted’, ‘Pourcentage Raised’ and of course, ‘Amount collected’.

It’s interesting to see that the even if we drop the 10% highest quantile, most of collects are “little” collects with 28 donors in mean. On the other hand, the success rate is very low, with half of the collections reaching only 5% of their target. Our first question was: how to explain the Amount Collected. Up to now, we have worked in a global way, but from now on we will look category by category, since each category has its own behavior, and a global analysis makes little sense (we will confirm this later with OLS Regressions)

**Analysis of the durations of collects:**

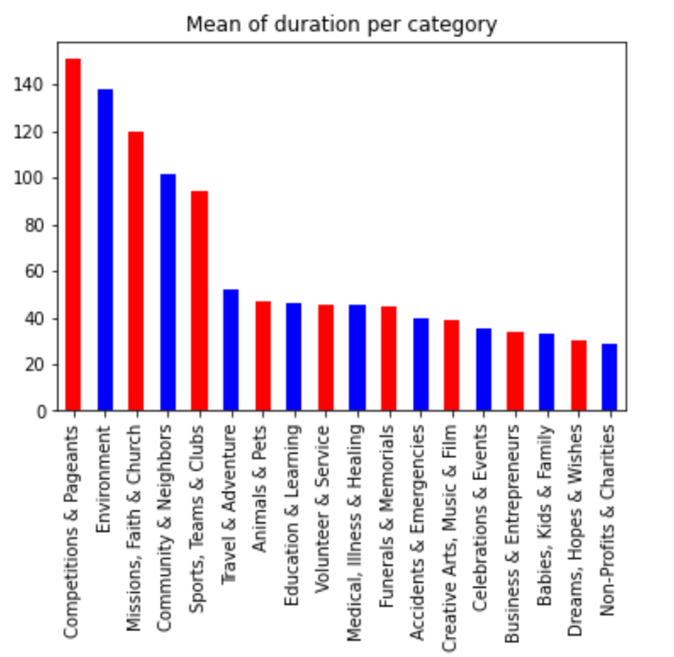
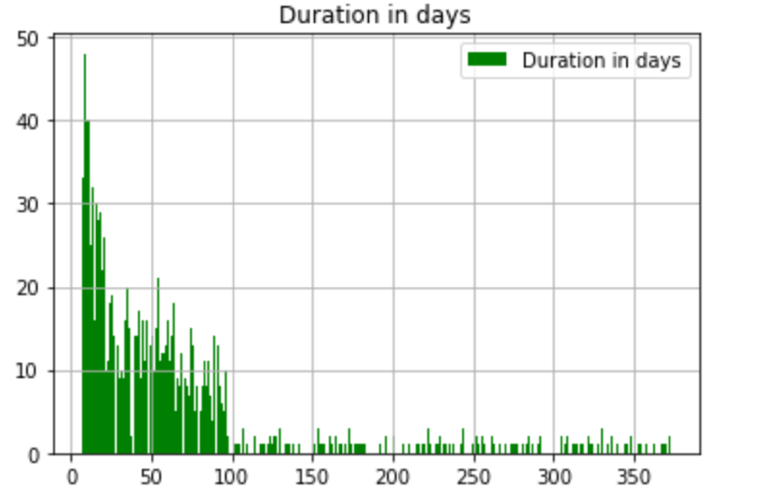


Figure 5 Means of duration per category

Figure 6Distribution of the duration

We can see that the collects displayed first by GoFundMe, the popular ones, are mostly ‘recent’ collects, under 100 days. 75% of them are even more recent than 74 days if we look at the *Figure4.*  Furthermore, the turnover depends on the category, for example, charities collects have an average duration of a month (*Figure 5)* when “Faith &Church” or “Environment” collects have less turnover, which confirms their long-term nature and why they take more time to be funded.

**Analysis of the amount targeted**

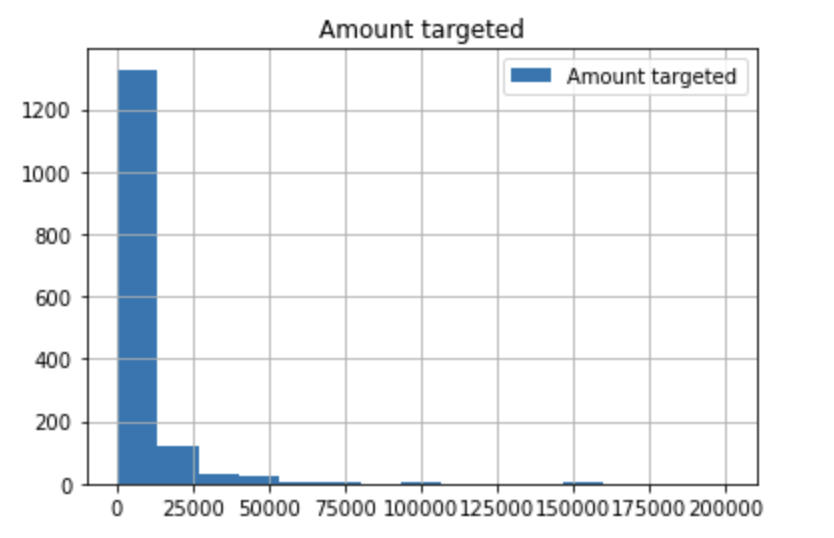
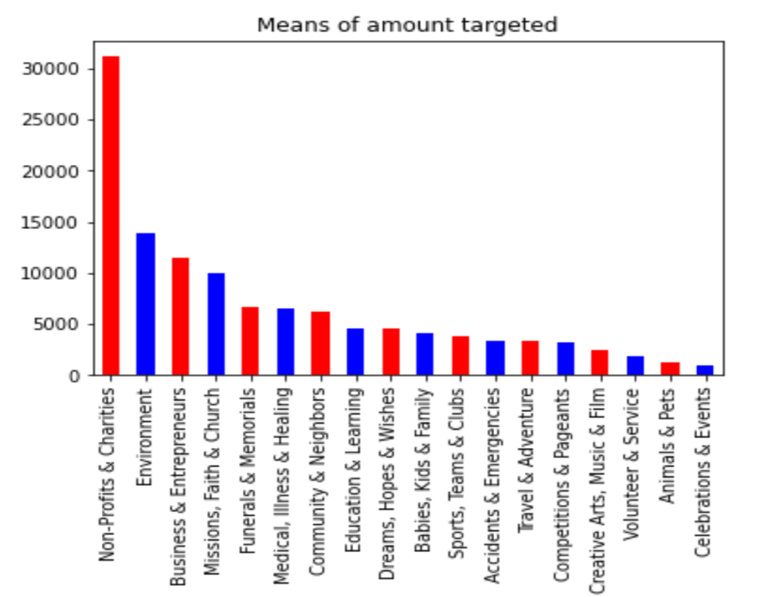
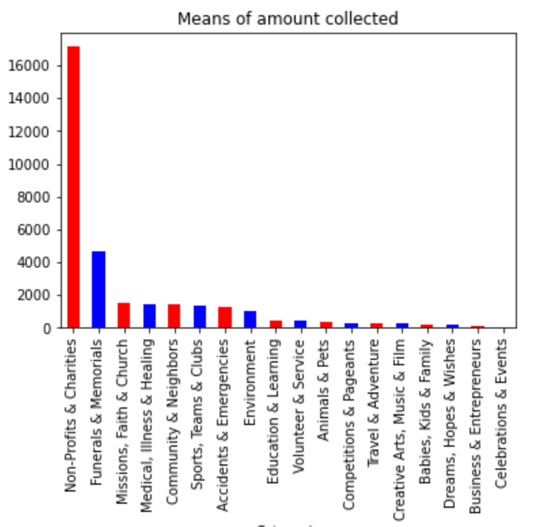


Figure 7 Means of the amount targeted by category

Figure 8 Distribution of the amount targeted

Figure 8 Distribution of the amount targeted

The *Figure 8* confirms clearly that most of the collects are “small collects” but *a*s we said, it’s irrelevant to compare the collect of different categories since they have different natures. As the *Figure 4* shows it, the mean targeted is 6 400€ but it really depends on the category: collect like ones for the animal (*Figure 7)* are mostly for small and individual projects compared to Environment projects.

**Une image contenant table

Description générée automatiquementAnalysis of the amount collected:**

Figure 9 Means of amount collected per category

Figure 10 Distribution of the amount collected

As we can see, collects are very inequal, for example “Funeral” collects are the second most successful (*Figure 9*) even though they are not the one of the collects requiring the most (*Figure 7)*. *The Figure 10* *and 4* inform us that most of collects are not funded (25% collect even 0€ and median at 90€), but some few are very successful (maximum at 245k€, even though we drop the 5% most requiring collects).

**ANALYSIS**

Our first question was: “What are the feature explaining the success of a collect ?”

Most of our analyze is then doing OLS regression and interpreting the covariates, their coef, their p-value to understand their influence. Now let’s look at the criteria of success : it could be the number of donors, the percentage raised or the amount collected. We chose the last one because if the amount requested is huge, although the collection is successful, it may have a low percentage raised. As for now, every OLS regression will have ‘Amount collected’ as Y. For our regression, we use statmodel because of its relevant table of summary. Now we have to choose the variables of interest, the X. Obviously, the more donors a collection has or the higher the average donation, the more successful it is (OLS very accurate with excellent R2 in appendix). What we are interested in is explaining the success of a fundraiser upstream, so we will look at the influence of its category, description, and city. These are the characteristics on which the fundraiser can really play. We do not consider the title because it is included in the description.

**Influence of the category:**

As the *Figure 9* informs us, among the 17 categories, some are more successful and, this is why, statistically, collects from categories like “charities”, “Funeral” are likely to succeed. This is proven in the *Figure 11*, summary of the regression. Such categories have a great coeff (17 00 for charities) and a very low p-value. Nonetheless the R2 is low (0.173) because as we said, each category has its behavior and it’s difficult to do a proper OLS solely between categories and the Amount collected. We used pd.dummy to use ‘Category’ that was a str column.

**Une image contenant table

Description générée automatiquement**

Figure 11 OLS regression with Category as variable

**Influence of the town:**

We use pd.dummy to use ‘town’, we can see it has very less impact and there is to much towns (824) so the R2 is excellent (0,9) but the p-value are mostly close to 1 because they are lot of towns which only have one study.Then It is not relevant to take the town into account. Cf Appendix

**Influence of the description:**

We used to points of view to tackle the problem: ScoreWord and dummies.

Our idea with our tutor was to find the keywords for each category. To do this, we took the most frequent words per category and created a list containing, for each category, among the words that occur the most in the descriptions, the 10 that we find interesting. The "ScoreWord" function then counts the occurrence of keywords in the project description to give it a ScoreWord. Our intuition was that the more keywords there are in a description (the bigger the ScoreWord), the more the project convinces donors and is therefore successful.

The second idea was to use pd.dummy to create a column for each keyword indicating whether it is present in the project description. For the OLS regression, we decided to focus only on one category, in this case “Non profit & Charities collects” because the tendencies are the same for this part (the OLS analysis for all categories are in appendix) Let’s look at the result.

Une image contenant texte, capture d’écran, reçu

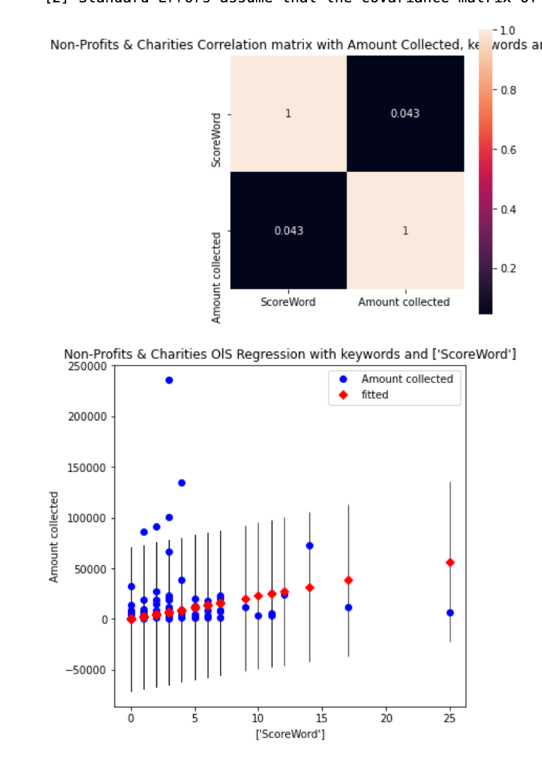
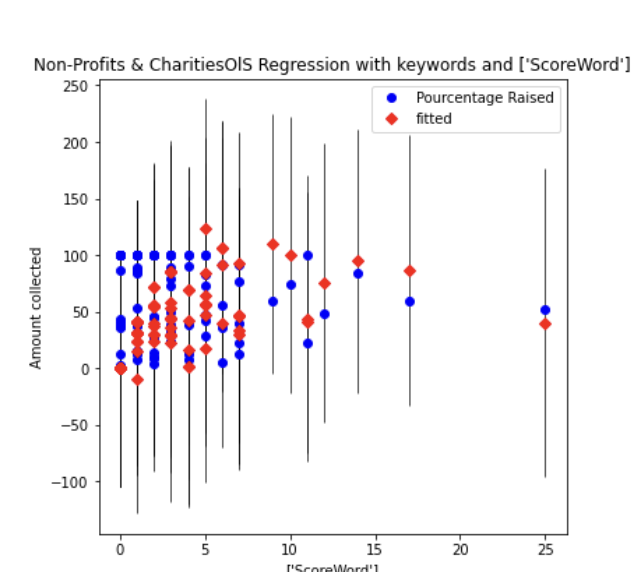
Description générée automatiquement

Figure 12 OLS Regression solely with 'ScoreWord'

Then, trying to predict the success of a collect only with its ScoreWord is not relevant (R2 of 0.108) and the correlation between these ‘ScoreWord’ and ‘Amount collected is very low (0.043)

Our second idea was now to do an OLS regression with the ‘ScoreWord’ and the presence or not of each keyword of the category in the description:

Une image contenant texte, reçu

Description générée automatiquement

Figure 13 OLS Regression with ScoreWord and keywords

This time, we have much convincing results: R2 of 0,545. The OLS summary indicates that keywords have inequal coef and p-value, for example children (coef=32 and p-value=0.057) is much more influent than foundat