**GOFUNDME : 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. We didn't have any capcha to deal with but when we didn't put a time.wait(), the loops failed. It was sometimes complicated to locate exactly the tag of such or such figure displayed on the screen. For the number of donors for example, we had to try with Selenium and then BeautifulSoup, going first through the class, then the xpath and then the selector or the CSS path. Sometimes projects have no donors for example and so the tag does not exist, we had to try with Try loops. 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 programme run longer to get more.

Une image contenant table

Description générée automatiquement

Figure 1: The Data freshly scraped