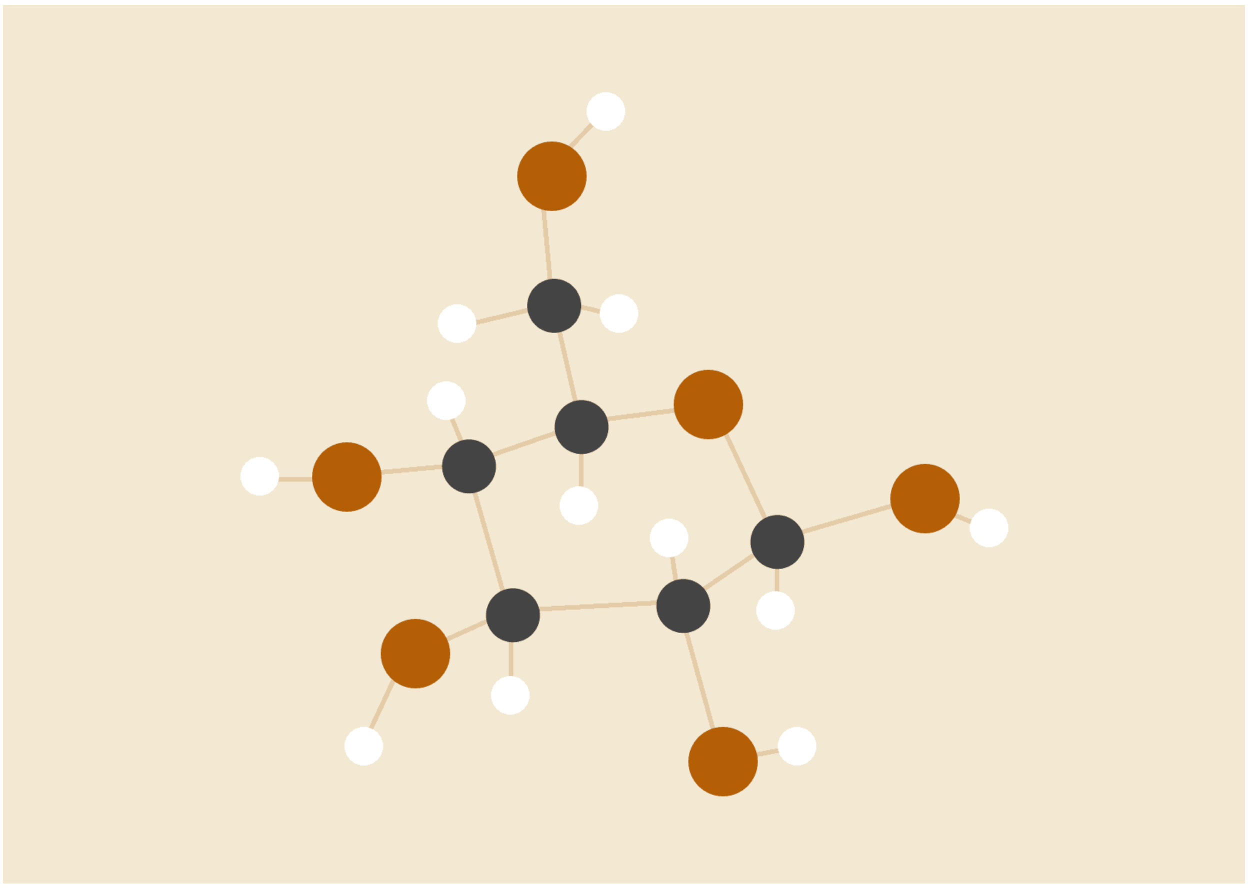
Web Scraping and Social Media Scraping - Project description -



Student

Dat Ngo (ID: 432395)
Anvesh Kotturi (ID: 428935)

May 2021

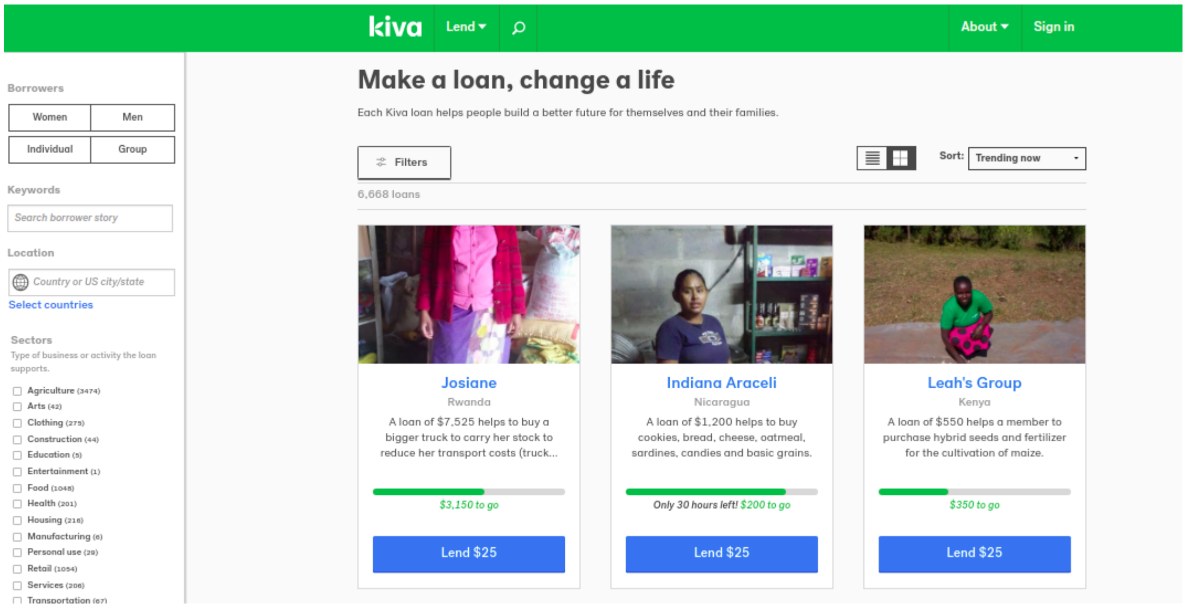
1. INTRODUCTION

In this project, we applied skills learned from class ***" Webscraping and Social Media Scraping"*** in crawling data of loans in the crowdfunding platform named Kiva (kiva.org). About Kiva, Kiva is an international nonprofit, founded in 2005 to expand financial access to help underserved communities thrive via crowdfunding loans. On Kiva's platform, everyone can lend as little as $25 to support the underserved.

The crawled data would help both borrowers and lenders have a comprehensive understanding of how Kiva organization is working in their country. Regarding borrowers, they can decide the appropriate loan, loan term and interest to apply based on peers. They can also search for key lending partners and networks who are active in their country as well as these ones' portfolios. In the perspective of lending partners and a social lending network. They can see the movement of loan amounts, percentage of successful campaigns, risk rating and changes in loan term, trend in loan sector to plan to support crowdfunding activities better.

2. SCRAPER MECHANICS

The project uses 3 tools (Selenium, Scrapy and Beautiful Soup) to crawl loan data started from the loan searching page (https://www.kiva.org/lend).



Make a loan, change a life

Each Kiva loan helps people build a better future for themselves and their families.

6,668 loans

Sectors
Type of business or activity the loan supports.

- Agriculture (3474)
- Arts (42)
- Clothing (275)
- Construction (44)
- Education (5)
- Entertainment (1)
- Food (1048)
- Health (201)
- Housing (216)
- Manufacturing (6)
- Personal use (29)
- Retail (1054)
- Services (206)
- Transportation (67)

Josiane
Rwanda
A loan of $7,525 helps to buy a bigger truck to carry her stock to reduce her transport costs (truck...
$3,150 to go

Indiana Araceli
Nicaragua
A loan of $1,200 helps to buy cookies, bread, cheese, oatmeal, sardines, candies and basic grains.
Only 30 hours left! $200 to go

Leah's Group
Kenya
A loan of $550 helps a member to purchase hybrid seeds and fertilizer for the cultivation of maize.
$350 to go

Because the loan searching page is dynamic, we cannot collect loan URLs on the page by Scrapy and Beautiful Soup. We firstly use Selenium to send the user's loan filers to the page (filters include: borrower's countries, borrower's genders, borrower's types group or individual, loan sectors, loan length in months, loan status and user's keywords). After that, we crawled the list of chosen loans' URLs by reading the website pages.

- Selenium's mechanics

- Script ***project_Selenium_URLs.py*** is used to crawl loan URLs based on user's input
- Script ***project_Selenium_loanDetails.py*** is used to crawl loan details for each loan URL

The main command we used is `driver.find_element_by_xpath`

The scraper firstly found web elements to send user's inputs and collect loan URLs. With each loan URL, we used the gecko driver to request the page's structure to crawl loan detail data.

For example:

- The scraper find necessary tags and send the user's gender option:

```
gender = driver.find_element_by_xpath("//div[@class = 'gender-button-box
triple-state-buttons']").find_element_by_xpath(".//*[text() =
'{0}']".format(genderDict[genderOption][0])) # find the selected gender tag

gender.click() # click the tag to select
```

- The scraper collects the filtered loan URLs:

```
loanURLs += [x.get_attribute('href')+'?minimal=false' for x in
driver.find_elements_by_xpath("//a[@class = 'loan-card-2-borrower-name']")]
# crawling loan URLs
```

- For each loan URL, the crawled loan info (eg: loan length):

```
loanLength = trim(driver.find_element_by_xpath('//a[text() = "Loan
length"]/../..//following-sibling::div').get_attribute('innerHTML'))
```

- Scrapy's mechanics

- Script ***project_Scrapy.py*** is used to crawl loan details for each loan URL. The command line $ scrapy crawl kivaLoans -O loanDetails_Scrapy.csv on terminal to run the scraper.

The main syntax we used is `x_path`

For example:

The scraper crawled time left for a loan using x_path:

```
timeLeft_xpath = '//div[re:match(@class, "^days-left-stat.*")]/text()'
l['timeLeft'] = trim(response.xpath(timeLeft_xpath).get())
```

- Beautiful Soup mechanics
- Script ***project_BeautifulSoup.py*** is used to crawl loans info based on loan URL.

Syntaxes use used include find, lambda and regular expressions

For example:

The scraper crawled country of borrowers:

```
country = trim(bs.find(lambda tag: tag.name == 'h2' and 'Country' in
tag.text).get_text().replace('Country: ',''))
```

3. TECHNICAL DESCRIPTION OF THE OUTPUT

We have the same output file using 3 methods. Following is the time consumed to crawl loan details from 100 URLs:

- Selenium: approximate **11 minutes**

- Beautiful Soup: approximate **6 minutes**

- Scrapy: approximate **40 seconds**

- Some comparison of 3 scrapers:

- Selenium ran for the longest period. Even the scraper is more flexible to select tags and can deal with some dynamic pages, it needs time to load pages and the website server might refuse to respond to the webdriver if a lot of requests are sent in a short period of time. Therefore, we set the driver to sleep for 2 - 5 seconds after each URL. This results in a high time consumed. In our opinion, selenium can be used to interact with pages and is more suitable to crawl a small number of pages.

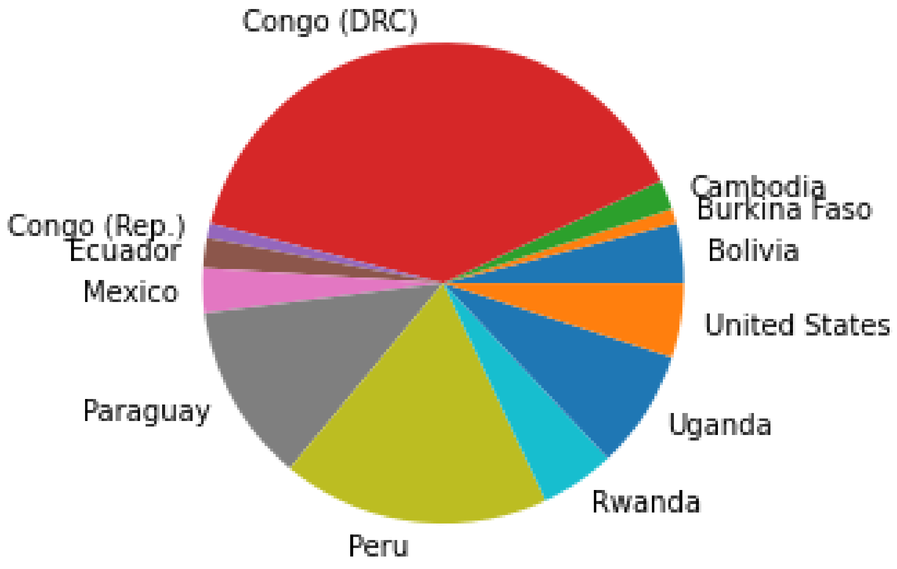
- Beautiful Soup ran faster than Selenium but it needs more complex syntaxes to find a tag. Besides, it cannot work with dynamic pages.

- Scrapy ran the fastest. Also, it automatically deals with exception errors and helps reduce time for coding. However, the scraper has many default settings and is not flexible for new users.

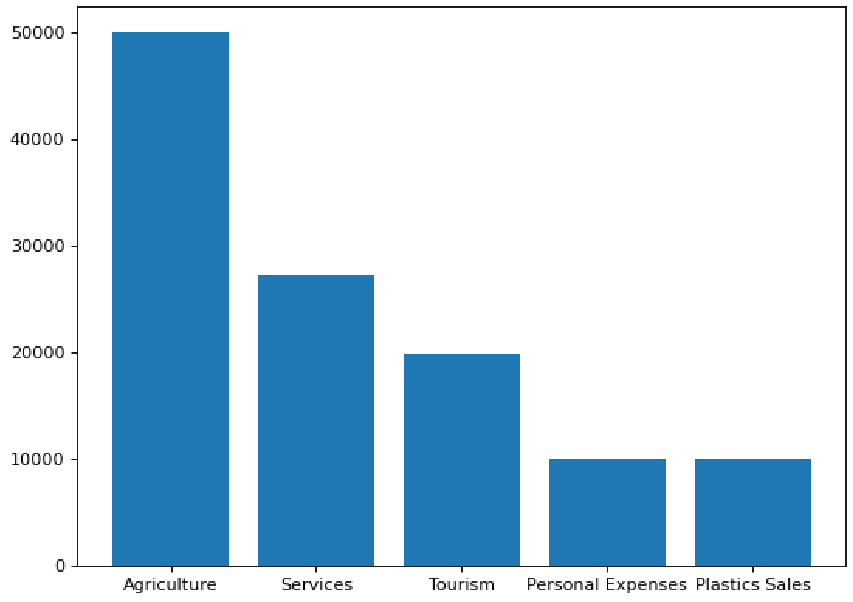
4. ELEMENTARY DATA ANALYSIS

Using the first 100 crawled loans from the page, we implemented some elementary analysis.

- Comparing countries regarding number of loans, most of loans are from Africa and South America countries including Congo, Peru and Paraguay, Uganda. Kiva focuses on these developing areas where people have many troubles and barrier to meet the finance resources.



- About loan sectors, we can see there is a big gap between the average loan amount of the first and the second sectors. Agriculture borrowers from Kiva usually have a larger loan because of the high cost for agriculture investment. However, agriculture loans also have high risk because the sector depends a lot on weather.
- With a larger data set, users can find many insights about loans such as the trend of loans, the appropriate loan term to borrow, the best lending partners to apply.



5. TEAM PARTICIPATION

Member	Activities
Dat Ngo	- Search for web scraping ideas - Write Selenium and Beatiful Soup script - Write the project description part 2, 3
Anvesh Kotturi	- Search for web scraping ideas - Write Scrapy script - Write the project description part 1, 4 - Test and review scripts