Project description – Justjoin.it

1. Short description of the topic of the web page.

Justjoin.it is a Polish start-up dealing with job offers from the IT industry. Primarily, this website is an 'intermediary' between the employer and job seekers. Anyone can place their job offer on the site for a fee. In turn, searching for offers from the user's level is free. The site is structured so that everyone can search for an offer with filters - location selection, ranges regarding expected salary, and a selection of interesting IT environments.

2. Justification of the choice of static/interactive methods in your project (interactive path requires scraping dynamic web page or filling forms during scraping process - you can use interactive path when it is not possible to use static methods).

We chose interactive methods in our project because Justjoin.it is a dynamic web page. There are some points that are not possible to handle without interactive methods – choosing filters like location and expected salary and scrolling down to obtain destined content of the page.

3. Short description of your scraper mechanics.

1) Declaration of initial values.

The user declares whether he wants to scrape a maximum 100 pages:

- if not – scraping all available pages;

The user declares whether he wants to choose a location:

- if yes – enters one of the available locations;
- if not – offers from all available locations are available;

The user declares whether he wants to choose ranges of expected salary:

- if yes:
  - the user declares minimum expected salary
  - the user declares maximum expected salary
- if not: expected salary in default website range: 0 – 50 000+ PLN.

2) Scraping

Selenium part

According to the user's declarations, the scraper has filtered the given request. First, to select a location, the desired city was clicked out of the available. In turn, the selection of

expected salary ranges was performed using ‘ActionChains’ in Selenium. It swipes horizontal slider due to user input (left-hand edge for minimum and right-hand edge for maximum).

The next step in the program is scraping links of offers (selected according to the values given by the user). Link gathering takes place through a loop that saves all offers on the page. Then using Key Page Down (JS query in Selenium did not work on this site so we came up with approach with Key Page Down) goes down saving subsequent links to offers until collecting links to 100 offers or going to the bottom of the page. Then, through the loop of links, scraper writes data to the lists, and finally creates a dataframe, which is saved to csv.

Scrapy part

We used Scrapy-Selenium middleware to handle justjoin.it website because in a static way practically it does not display anything but HTML header, so it was necessary to use middleware (as it is recommended in Scrapy documentation). Unfortunately, there was a problem with scrolling down the main page in the Scrapy, because justjoin.it’s dynamic front-end framework does not support JavaScript requests from the user. Furthermore, in Scrapy-Selenium it is not possible to trigger the Key Page Down (as it was in the Selenium part). To solve this problem we found a JSON file, which is a server response to the user request - entering the site. Thus, we were able directly to scrap all the offers links on this site. Scrapers are not exactly the same as in the Selenium, but the whole scraping process is analogous, except the part with gathering links to a specific job offers.

4. Short technical description of the output you get.

Scraped data consists of a total of 12 variables, represented in a tabular format:

- **offer_link** - URL link to a job offer. *(i.e.https://justjoin.it/offers/cybersecurity-studio-sp-z-o-o-backend-developer)*
- **offer_title** - a title of the position *(i.e. Backend Developer)*
- **company_name** - a name of a company *(i.e. Cybersecurity Studio sp. z o.o.)*
- **company_size** - a number o employees in a company, expressed as a range *(i.e. 40-45)*
- **employment_type** - a type of contract for a job *(i.e. b2b)*
- **experience_lvl** - an indicator to which experience level an offer is dedicated to *(i.e. mid)*
- **salary** - salary range, expressed as a range with a lower and upper boundary, currency, time period and income type *(i.e. 10 000 - 15 000 PLN net/month)*
- **place** - physical address of a company that offers a job opening, consisting of street, building number, town. *(i.e. Gdańska 2, Warszawa)*
- **tech_stack** - a list of dictionaries of technologies/skills requested by a job position, paired with a level of advancement. *(i.e. [{'.Net': 'regular'}, {'ELK Stack': 'regular'}, {'C#': 'regular'}])*

- **direct_apply** - boolean value, which takes a TRUE value, when on a job offer page exists a possibility to apply for this job position directly through the job offer page, by clicking the 'Direct Apply' button. *(i.e TRUE)*
- **company_page** - URL link to a page of a company, which has listed a job position. *(i.e. http://www.cyberstudio.systems/)*
- **offer_description** - description of a position in a form of long text. *(i.e 'Cybersecurity.studio to przede wszystkim zespół entuzjastów nowoczesnych technologii...'*

5. Extremely elementary data analysis - you need to prove, that collected data can be used for further analysis, but nothing more (histograms, means, chosen word/letter count is enough - no models here!).

An extension of this part, with codes for data wrangling and plotting, is available in the Jupyter Notebook Visualizations.ipynb.

a) Salary

First, by calculating the midpoint of each salary range ([lower_bound+upper_bound)/2], and then performing an analysis on obtained values we get the following results:

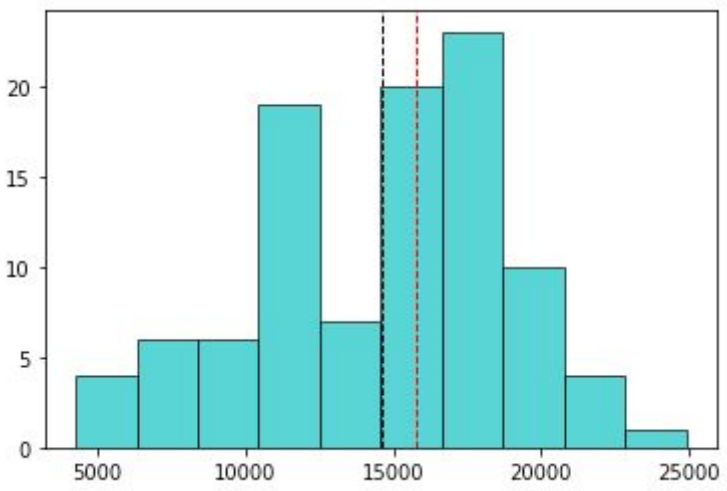

Figure 1. Histogram of midpoint values of salary ranges with a mean (black line) and median (red line).

Average salary: 14641 PLN.
Median salary: 15750 PLN
Min salary: 4250 PLN
Max salary: 24900 PLN.
Analyzing Figure 1., we can see cut-off values on 10 000 PLN, 15 000 PLN, and approx 17 500 PLN.

b) Experience

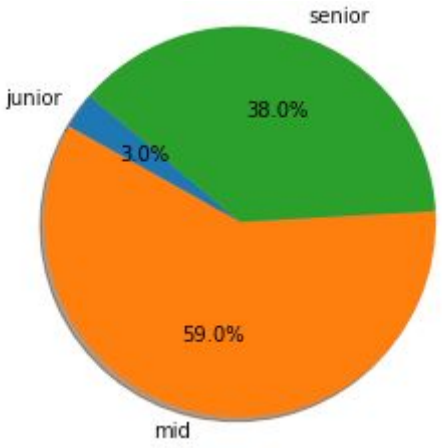

Figure 2. The percentage share of offers for different experience levels

The majority of offers are dedicated to mid's, following by 38% share of senior offers and only 3% of offers for juniors.

c) Company types

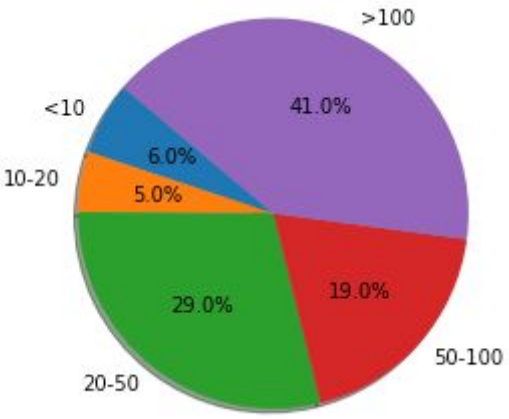

Figure 3. The percentage share of company sizes, measured in the number of employees

Over 40% of companies have 100 or more currently employed employees, suggesting that big corporations are the biggest employees in the IT sector.

d) Technological stack

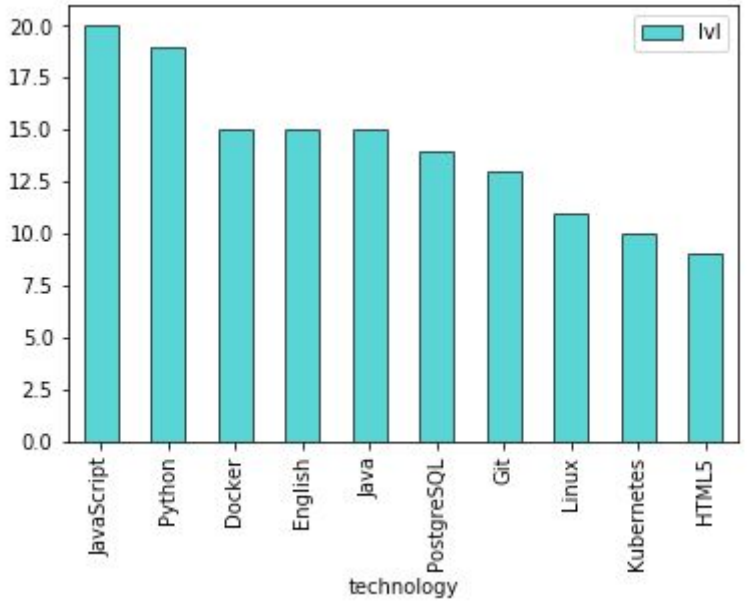

Figure 4. Prevalence of technologies in job offers (sorted by their count), top 10.

JavaScript and Python are the two most in-demand technologies. We can also see high importance of the English language as well as knowledge of containers technologies and Linux.