Data Analytics – Exercises

(Week 01)

These exercises are about the first three parts of the Data Analytics Process Model (highlighted in yellow in the figure below). The results must be uploaded as separate files (no .zip files) by each student on Moodle. For details on how submit the results see the tasks below.
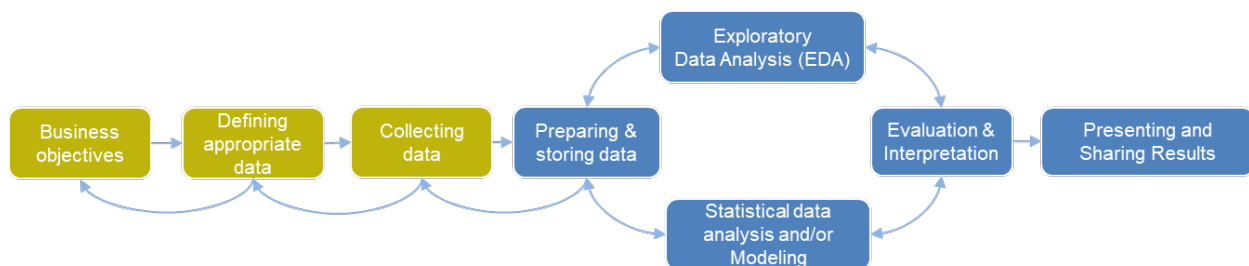

Figure 1: Data analytics process model (see lessons)

Note: It is assumed that basic knowledge of Python and Jupyter notebooks is available from the Statistics module.

Task 1

In this exercise, you must define business objectives and hypotheses for the two use cases defined below. Prepare a short description (bullet points, short notes) by using the following structure:

1. Problem definition (identify one or more potential problem(s)).

2. Objective (define a clear objective based on the problem definition).

3. Research question (formulate a research question based on the objective).

4. Hypothesis (formulate a clear hypothesis based on the research question).

5. Expected added value of the data analysis (business value of data analytics).

The use cases are:

Use-Case (1): Price prediction model for rental apartments in the canton of Zuerich

Use case (2): Demand and supply for supermarkets in Swiss municipalities

To be submitted on Moodle: nothing 😊 but discuss your findings in class!

Task 2

In this exercise, you must propose and describe suitable data for each of the use cases from task 1 and briefly describe the necessary data. You don't have to collect this data in this task; an overview of the data suitable for achieving your business goals is sufficient. Please note that you will need to combine data from multiple sources for the data analysis. For example, it may be helpful to supplement the apartment and supermarket data with municipality-level data. You can find municipality-level data on the homepage of the Swiss Federal Statistical Office (SFSO): "Regionalporträts 2021: Kennzahlen aller Gemeinden". Data sources for the use cases are:

Use-Case (1): Price prediction model for rental apartments in the canton of Zuerich

https://www.immoscout24.ch

Note: See which information is available per offer.

Municipality-level data: Regionalporträts 2021: Kennzahlen aller Gemeinden

Use-Case (2): Demand and supply for supermarkets in Swiss municipalities

https://wiki.openstreetmap.org/wiki/DE:Key:shop

Note: See which data is available about supermarkets on OpenStreetMap.

Municipality-level data: Regionalporträts 2021: Kennzahlen aller Gemeinden

To be submitted on Moodle: nothing 😊 but discuss your findings in class!

Task 3

In this exercise, you will practice retrieving data using a Web API and Web Scraping. For use case (2), the data can be obtained via a Web API. Please use the Jupyter notebook 'getting_supermarket_locations_WebAPI.ipynb' for this purpose. Web scraping is necessary for use case (1). In this Web Scraping exercise, we will use the city of Winterthur as a test area to scrape data of rental apartments. Therefore, install the 'Web Scraper' extension for Chrome in the Chrome browser. Then perform the following steps:

1. Use Google Chrome to navigate to the website immoscout24.ch/en and go to Rent -> Winterthur -> Show ... properties. Search for apartments only.

2. Open the Chrome Web Scraper extension via -> right mouse button in the browser window -> Explore -> Browser menu -> Web Scraper. Alternatively, use CTRL+SHIFT+I (Windows) or CMD+OPTION+I (macOS) to show the Web Scraper in the developer tools

menu of the Chrome browser. In Chrome DevTools, you need to switch to the correct tab (see Figure 2), and the DevTools must be displayed at the bottom, not on the right or left side.
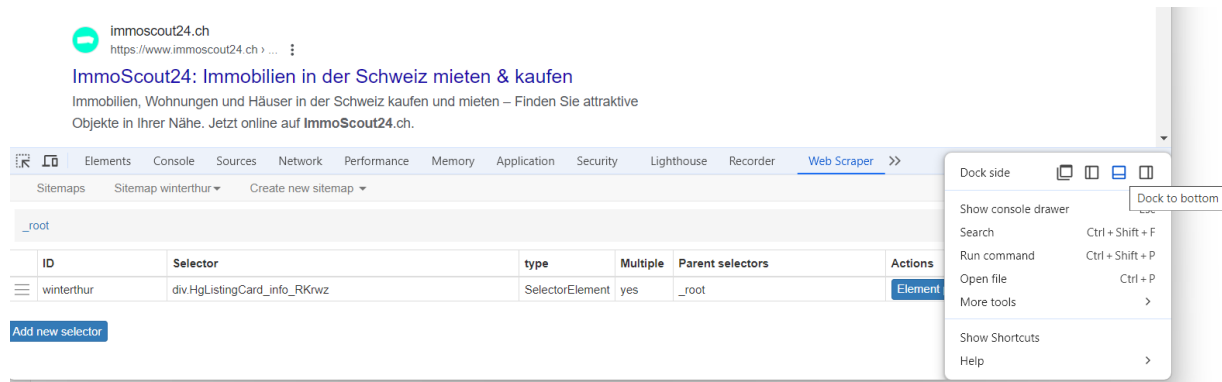

Figure 2: Switch to the correct tab and dock to bottom.

3. Create a new sitemap with the Web Scraper: Web Scraper menu -> Create new sitemap. Name the sitemap 'winterthur'. The start URL is: https://www.immoscout24.ch/de/wohnung/mieten/ort-winterthur?pn=[1-5]. Note that 'pn=[1-5]' stands for the number of web pages with rental apartments offers that must be scraped.

4. Add a new selector via 'Add new selector' and name it 'winterthur'.

5. Select 'Element' as the type.

6. Within the 'Element' selector, go to Selector -> Select and mark the first offer on the web page (whole box to the right of the photos of an offer). Then select a second and third offer until all offers on the first web page are highlighted. Then click on 'Done selecting! The CSS selector is now transferred to the 'Selector' field.

7. Additionally check 'Multiple'. Save the selector with 'Save selector'.

8. After saving, click on the line with the name of the selector ('winterthur'). This will take you to the second level of the selector.

9. Add a new selector by using 'Add new selector'. Name it 'rooms_area_price_raw' and select 'text' as type. The '_raw' at the end of the name indicates that this is raw data, which must be processed later. Click on Selector -> Select in the scraper. The first offer should now be highlighted. Click on the line with the number of rooms, living space and price and then on 'Done selecting'. The CSS selector should now appear in the selector window. Go to 'Data preview' and check that the number of rooms, area and price appear.

10. Add more text selectors in the same way: 'price_raw' for the rental price, 'address_raw' for the address, 'description_raw' for the description of the offer and 'text_raw' for all the text in the text box of the offer (see Figure 3).

3

Figure 3: Example of text selectors in the Web Scraper

11. Start the web scraper: Menu -> Sitemap winterthur -> Scrape -> start scraping.

After scraping (takes a few seconds) refresh the browser view using the 'refresh' button.

12. The data can be downloaded via Menu -> Sitemap winterthur -> Export data as .csv.

13. Export the sitemap via Menu -> Sitemap winterthur -> Export sitemap and save it as sitemap_winterthur.json. This can be done by using a text editor like Notepad++.

To be submitted on Moodle:

a) One json file with the supermarket data in Switzerland (supermarkets.json).

b) One csv-file with the scraped rental apartment data of the municipality of Winterthur (apartment_data_winterthur.csv).

c) One json-file with the Web Scraper sitemap (sitemap_winterthur.json).