**ANALISIS FOR A SCRAPY CARPROJECT EBAYKLEINEINZEIGEN**

Operations on links (filtering database):

https://www.kleinanzeigen.de/s-autos/mercedes\_benz/c216+autos.ez\_i:1990%2C2000+autos.fuel\_s:benzin+autos.km\_i:60000%2C150000+autos.marke\_s:mercedes\_benz+autos.model\_s:190+autos.power\_i:34%2C252+autos.schaden\_s:nein+autos.shift\_s:automatik

FIXED PART: <https://www.kleinanzeigen.de/s-autos/c216>

BRAND ASSIGNING:

/mercedes\_benz/ add in the middle of link

THE ORDER IN LINK: (THERE IS NONE)

(it is pretty simple and have NOT A fixed layout)

+autos.marke\_s:mercedes\_benz 🡪 brand of a car (SHOULD BE ASSIGNED PREVIOUSLY)

+autos.model\_s:190 🡪 model of a car (SHOULD BE ONE OF SOME – any other can make program kapput the same with brand,fuel and all JS variables that are strings)

+autos.km\_i:60000%2C150000 🡪 auto mileage (in km)

+autos.ez\_i:1990%2C2000 🡪 production year between 1990 – 2000

+autos.\_fuel\_s:benzin 🡪 as the name tells us – name of a fuel (there are some distinctive)

+autos.power\_i:34%2C252 🡪 power of auto between 34ps and 252ps

+autos.schaden\_s:nein 🡪 if auto is damaged

+autos.shift\_s:automatic 🡪 shift of auto

THAT’S ALL IN LINK – so we can make an assumption that everything on this page can be filtered only by making changes in link – in project I will try this method for filtering cars and all other stuff such as phones or computers (that’s for later)

The assumption turned out to be true so we can continue to work with ebay that way!

Ways of using program:

Mine first idea was to use this program only to extract prices about specific product, however it turned out that Scrapy is way more powerful than I thought so the second step is to display more complex information by entering sub webpage of every single product program parses. From there we can extract important information important for further Data Science projects and Data Analysis. Using thousands of examples of cars for sale in Poland (on olx or otomoto) we can make a

#FINISH!!!

Plan for updating program:

1. IMPORTANT: my first idea – a program built to check average price of a specific car with fixed parameters (mileage etc.) on Ebaykleineinzeigen (with of course fixed approximation error [in case of mileage or year of production]) – **that’s the first thing to do** (the next is to compare those prices to Polish, this idea assumes that you have a broad knowledge about Polish auto market and current prices – you just know them, program don’t need to check it for you) – Program is already created inside project documents folder
2. Update the existing Kleineinzeigen program: - DONE
   1. Display more useful data for Data Analysis
      1. Mileage
      2. Engine type
      3. Year of production
      4. Condition – NOT NECESSARY
      5. ?Description? (I don’t know yet if I will use it – NOT NECESSARY
   2. Change the way data is filtered – more must be done by Pipelines – DONE
      1. Especially the ‘VB’ thing - DONE
   3. Research the way it can have the data given by tkinter python program – TO DO
3. Create a Otomoto program:
   1. It must be completely independent spider that will save the data in the same file that Kleineinzeigen one but in different localization
   2. The same data must be displayed here
4. Create a OLX program:
   1. To start it I need to make research about CloudFront (that’ll probably require agreement with OLX admin, so the better way is to use fake user agents)
   2. Include a Fake user agent into the program and collect data from OLX
5. When all above done We can start a soft Data Analysis using Excel – making charts of prices in Poland and Germany (Y should be price and X – mileage, year of production or so.

For now there are going to be 2 independent versions of spider – the first one – displaying only prices of collected cars (it must have all important data about specific car – using that data analyses all prices in Germany) – FIRST VERSION SAVED IN first\_version folder inside the project documentation.

SECOND VERSION – inside the project.

**ISSUES:**

* Some of the pages are not about selling cars but about searching for a car – those pages have a special element next to mileage and year of production – this little field with “gesuch” is destroying our input not only for a single element but for a whole page we need to figure out why does it work this way – DONE
  + 1 idea – if “gesuch” found just DropItem - WORKS