

Furniture stores extraction

Irimia Octavian

1.	Introducere	3
2.	Date de intrare	3
3.	Abordarea problemei	3
3.1.	Extragerea url-urilor din fișier într-o listă	3
3.2.	Extragerea conținutului site-urilor pentru antrenarea modelului	3
3.3.	Pregătirea datelor de antrenare	4
3.4.	Pregătirea datelor de evaluare	4
3.5.	Antrenarea modelului	4
3.6.	Testarea modelului	4
4.	Evaluare	4
5.	Explicarea codului	5
6.	Antrenarea modelului	8
9.	Afișarea datelor	9
10.	Provocări	11
11.	Concluzie	12

1. Introducere

Acest document descrie un model de extracție a produselor de pe site-urile magazinelor de mobilă. Programul preia o listă de URL-uri și identifică numele produselor de tip mobilier disponibile pe fiecare pagină web.

2. Date de intrare

Programul primește un fișier de intrare de tip CSV numit "furniture_stores_pages.csv", care conține pe prima coloană adresele paginilor web ale magazinelor de mobilă.

Pentru curățarea textului extras de pe paginile web am mai introdus încă două fișiere numite "stop_words_english.txt" și "codes-all.csv". Primul fișier conține cuvinte uzuale, des întâlnite (in, out, our, me, etc.), iar al doilea conține coduri valutare (USD, CZK, EUR, etc.). Cel de-al doilea fișier a fost introdus pentru că am observat că în textul extras de pe site-uri se află și aceste coduri valutare care a putea îngreuna procesarea textului.

Pentru adnotarea datelor am folosit un fișier de tip text în care pe fiecare linie e trecut câte un nume de mobilier; de asemenea am inclus și numele obiectelor de mobilier la plural.

3. Abordarea problemei

3.1. Extragerea url-urilor din fișier într-o listă

3.2. Extragerea conținutului site-urilor pentru antrenarea modelului

Am extras primele 100 de url-uri din lista menționată anterior. Am accesat fiecare url, iar din cele ce au răspuns pozitiv accesării am extras conținutul. În acest conținut extras am verificat dacă mai există și alte url-uri. Am extras toate aceste url-uri și le-am filtrat în modul următor; dacă url-ul inițial era "<https://www.example.com/content/furniture/abc123>" am extras doar prima parte încât a devenit doar "https://www.example.com", astfel din url-urile extrase din cel inițial să le filtrez pe cele de care nu aveam nevoie (spre exemplu url-uri care duc spre anunțuri publicitare) și să rămân doar cu cele care se îndreaptă tot către url-ul inițial. Astfel din fiecare dintre cele 100 de url-uri inițiale am încercat (unele nu puteau fi accesate) să extrag url-uri pentru a-mi mări lista de date de antrenare.

După mărirea liste-i de url-uri am extras textul vizibil din fiecare site și l-am filtrat așa cum am menționat anterior. Am salvat textul filtrat într-un fișier text pentru a vedea cum arată.

3.3. Pregătirea datelor de antrenare

Pentru a realiza datele de antrenare am utilizat textul extras de pe site-uri și cu ajutorul fișierului în care aveam nume de mobilier am început să adnotez datele. Adnotarea a fost realizată tot cu ajutorul bibliotecii de expresii regulate. Pentru fiecare text din fiecare url am salvat poziția caracterelor de început și de sfârșit unde se găsește numele de mobilier i-am pus eticheta 'FURNITURE'. Aceste date le-am salvat și într-un fișier de tip JSON pentru a putea fi vizualizate, dar și în formatul necesar bibliotecii pentru antrenarea modelului.

3.4. Pregătirea datelor de evaluare

Am procedat la fel ca la datele de antrenare doar că datele de evaluare au fost generate din 50 de url-uri inițiale.

3.5. Antrenarea modelului

Antrenarea am realizat-o în versiunea gratuită a lui Google Colab. Pentru primul model antrenat am reușit să accesez și un GPU, însă la antrenările ulterioare nu am mai reușit să accesez niciun GPU în versiunea gratuită (probabil toate erau utilizate).

3.6. Testarea modelului

Aceasta a fost realizată cu utilizând url-urile rămase în fișierul inițial. Datele de ieșire au fost reprezentate sub formă de tabel, dar și într-un fișier de tip JSON.

4. Evaluare

Evaluarea modelului poate fi realizată manual comparând datele de ieșire cu o listă de produse extrasă manual de pe un site de mobilă.

5. Explicarea codului

- **load_text_file:**

- citește fișiere de tip text și returnează un set cu șiruri de caractere;
- în fiecare șir de caractere se găsește câte o linie din fișier;
- dacă fișierul nu este găsit se afișează o eroare și se închide programul.

- **load_currency_codes:**

- încarcă codurile valutare dintr-un fișier csv;
- codurile se găsesc pe coloana a treia din fișier, însă nu toate liniile au trei coloane (nu sunt coduri valutare pentru toate țările) așa că se verifică dacă sunt 3 coloane (dacă nu sunt cel puțin trei coloane se trece peste acesta) și se introduce într-un set codul u litere mici;
- în caz că nu există fișierul se afișează un mesaj de eroare.

- **extract_urls_from_csv:**

- trece peste prima linie pentru că aceasta conține altceva;
- ancarcă fiecare linie și elimină spațiile libere;
- afișează un mesaj de eroare în caz în care fișierul nu este găsit.

- **annotate_data:**

- primește ca intrare datele extrase de pe site-uri sub forma unui set de șiruri de caractere, unde fiecare șir de caractere reprezintă textul filtrat extras de pe fiecare site, datele de adnotare sub tot sub aceeași formă, calea unde să salveze datele adnotate și eticheta pusă datelor, în cazul acesta 'FURNITURE';
- se creează o listă goală (training_data) unde vor fi stocate datele adnotate sub format json pentru a putea fi vizualizate mai ușor;
- se încarcă un model spacy pre-antrenat pentru limba engleză;
- se creează un container pentru date de tip spacy;
- pentru fiecare regula din annotation_rules, adică pentru fiecare nume de mobilier se creează o expresie regulată pentru a identifica aparițiile cuvântului, apoi se caută potrivirile și pentru fiecare dintre acestea se extrage numărul caracterului la care începe potrivirea (start) și numărul caracterului la care aceasta se

sfârșește (end); se salvează într-o listă fiecare un tuple care conține start, end și eticheta, în cazul ăsta 'FURNITURE';

- dacă s-au găsit entități se salvează în training_data și se creează un document spacy din textul extras de la site;
- se creează un span de tip spacy unde se salvează start, end și eticheta și se adaugă la lista ents;
- se filtrează entitățile pentru a elimina eventualele suprapuneri cu modelul pre-antrenat pentru limba engleză (unele nume de mobilier pot fi marcate în modelul pre-antrenat cu o altă etichetă decât cea setată de mine);
- se adaugă documentul în containerul db;
- se salvează datele în format json și format spacy.

- **fetch_html_content:**

- date de intrare url-ul și un parametru boolean care dacă e setat pe true funcția va returna conținutul text, altfel conținutul brutș
- se face o cerere către url, dacă se va primi un răspuns pozitiv se va returna conținutul, altfel va apărea o excepție care va printa un mesaj.

- **is_visible:**

- verifică dacă textul este vizibil utilizatorului prin verificarea tag-ului; dacă tag-ul se află în lista din cod se va returna fals;
- se verifică și dacă elementul este de tip comentariu; dacă da se returnează fals, deoarece ne interesează doar textul site-ului, nu și cel adăugat de oameni prin intermediul comentariilor.

- **clean_and_filter_text:**

- date de intrare textul, stop_words și currency_codes;
- inițial se folosește o expresie regulată pentru a înlocui orice caracter care nu este literă cu un șir gol, mai pe scurt să îl elimine;
- se separă textul în cuvinte individuale și se elimină currency_codes și stop_words;
- se unesc din nou cuvintele după ce sunt eliminate duplicatele.

- **extract_text_from_url:**

- apelează funcțiile anterioare.

- **extract_associated_pages:**

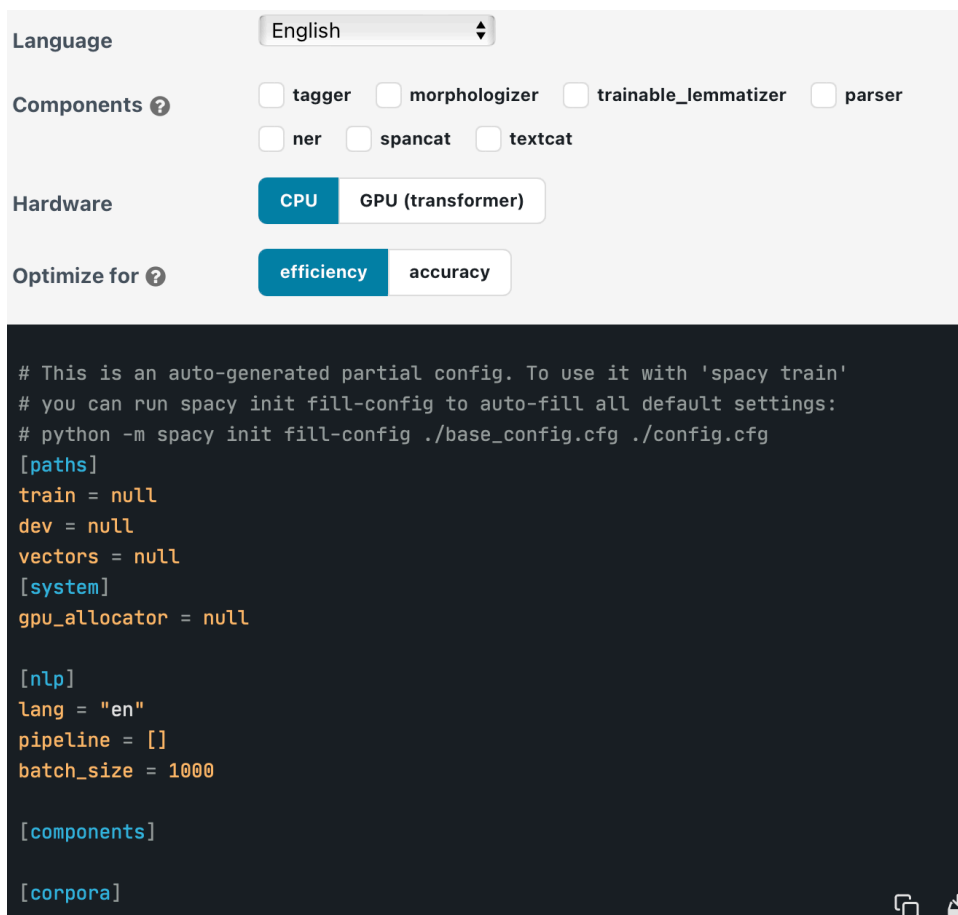
- date de intrare un url cu ajutorul căruia se apelează funcția `fetch_html_content`, care returnează conținutul brut al paginii;
- se creează un obiect de tip `BeautifulSoup` și se extrage domeniul url-ului inițial;
- folosind metoda `find_all` din `BeautifulSoup` se caută toate tagurile "a" care au un atribut de tip "href";
- se iterează prin fiecare link găsit și se verifică dacă valoare atributului "href" începe cu un șir specific de tipul "https://www.<domeniu extras>.<sufix domeniu>", asta pentru a ne asigura că extragem doar linkuri care au domeniul url-ului inițial, fără a exista posibilitatea să extragem linkuri către reclame sau altele;
- se returnează lista de link-uri extrase din url-ul inițial.

- **extract_website_data:**

- date de intrare lista de url-uri, `stop_words`, `currency_codes` și o cale pentru a salva datele;
- inițial se extrag toate url-urile asociate cu fiecare url în parte, adică toate url-urile care se află în conținutul brut al fiecărui url din lista inițială; pentru a îmbunătăți viteza extragerii acestora, am găsit o metodă prin care să se creeze mai multe procese în paralel care extrag link-urile asociate; viteza chiar s-a îmbunătățit considerabil;
- în același mod descris anterior se extrage și textul de pe site-uri;
- se salvează datele într-un fișier pentru a putea fi vizualizate și utilizate ulterior.

6. Antrenarea modelului

După crearea fișierelor de antrenare și evaluare de tip spacy (training_data.spacy, evaluation_data.spacy) a trebuit să încep antrenarea modelului. Citind prin documentația acestei biblioteci am aflat că voi avea nevoie de un fișier de configurație care poate fi generat pe site-ul acestora.



```
# This is an auto-generated partial config. To use it with 'spacy train'
# you can run spacy init fill-config to auto-fill all default settings:
# python -m spacy init fill-config ./base_config.cfg ./config.cfg
[paths]
train = null
dev = null
vectors = null
[system]
gpu_allocator = null

[nlp]
lang = "en"
pipeline = []
batch_size = 1000

[components]

[corpora]
```

Astfel în cazul meu am ales limba engleză, ner pentru named_entity_recognition, GPU având la dispoziție un laptop cu GPU și accuracy. După descărcarea acestui fișier (salvat ca base_config.cfg) a urmat o prelucrare ulterioară în terminal cu ajutorul comenzii “python -m spacy init fill-config base_config.cfg config.cfg” care a generat fișierul final de configurație necesar antrenării.

După configurarea laptopului (am instalat toate programele necesare pentru ML pe GPU, precum CUDA Toolkit) am încercat antrenarea modelului cu ajutorul comenzii “python -m spacy train config.cfg --output output --paths.train training_data.spacy --paths.dev evaluation_data.spacy --gpu-id 0”. Imediat după rularea comenzii am primit “blue screen” și mi s-a repornit laptopul. Am mai încercat de câteva ori având același rezultat, așa că am generat un fișier de configurație potrivit antrenării utilizând CPU, iar antrenarea a început în mod normal. Deși în trecut am mai rulat antrenări pe GPU și au funcționat, de data asta nu a mers.

Totuși dorindu-mi să cresc viteza de antrenare am decis să nu renunț la ideea de GPU așa că am intrat pe Google Colab și, spre surprinderea mea, am reușit să accesez un GPU în versiunea gratuită a Colab-ului (probabil nu erau ocupate cele disponibile pentru versiunea gratuită). Astfel în final prima mea antrenare a modelului a decurs integral pe un GPU în cloud. Totuși asta nu a mai funcționat ulterior, nereușind să mai accesez un GPU, antrenările ulterioare fiind executate exclusiv pe CPU.

După terminarea antrenării a rezultat un fișier în care erau două modele și anume “model_best” și “model_last”.

```
!python -m spacy train config.cfg --output ./output --paths.train /content/training_data.spacy --paths.dev /content/evaluation_d

✓ Created output directory: output
i Saving to output directory: output
i Using CPU

===== Initializing pipeline =====
✓ Initialized pipeline

===== Training pipeline =====
i Pipeline: ['tok2vec', 'ner']
i Initial learn rate: 0.001
E   #      LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
-----
0     0          0.00    100.33    0.00    0.00    0.00    0.00
0    200          9.27   1474.31    99.60    99.47    99.73    1.00
1     400          5.99    24.18    98.72    99.73    97.74    0.99
1     600         20.64    56.73    99.27    99.60    98.94    0.99
2     800         16.55    46.36    98.55    97.77    99.34    0.99
2    1000          7.89    12.81    99.53    99.47    99.60    1.00
3    1200         17.14    16.11    99.34    98.69   100.00    0.99
3    1400         24.92    33.36    99.73    99.87    99.60    1.00
4    1600          6.51     9.22    99.93    99.87   100.00    1.00
4    1800          5.65     8.47    99.80    99.60   100.00    1.00
5    2000         13.60    12.06    99.67    99.34   100.00    1.00
5    2200          3.15     4.12    99.47    98.95   100.00    0.99
6    2400          1.75     2.00    99.93    99.87   100.00    1.00
7    2600          0.00     0.00    99.93    99.87   100.00    1.00
8    2800          0.00     0.00    99.93    99.87   100.00    1.00
9    3000          0.00     0.00    99.93    99.87   100.00    1.00
10   3200          0.00     0.00    99.93    99.87   100.00    1.00
✓ Saved pipeline to output directory
output/model-last
```

9. Afișarea datelor

Datele sunt afișate în terminal sub următoarea formă: “link: produse găsite”. Spre exemplu “www.example.com: chair, table, desk”. Deși pot un obiect de mobilier este prezent de mai multe ori pe pagină el este afișat o singură dată.

Se afișează și un tabel în care prima pe prima coloană se află nume de produse de mobilier și pe a doua coloană pe câte pagini a apărut; dacă produsul apare de mai multe ori pe aceeași pagină el va fi luat în considerare o singură dată. Produsele vor fi afișate descrescător în funcție de numărul de apariții. Produsele care apar la singular nu vor fi cumulate cu cele care apar la plural, astfel primul produs care apare în tabel nu înseamnă că este cel mai des întâlnit.

The screenshot shows a Jupyter Notebook environment. The left sidebar contains a file explorer with the following structure:

- VERIDION
 - assets
 - output
 - output 2
 - output_initial
 - base_config.cfg
 - codes-all.csv
 - commands.txt
 - config.cfg
 - evaluation_data.json
 - evaluation_data.spacy
 - furniture_names.txt
 - furniture_stores_pages.csv
 - output_data.json
 - stop_words_english.txt
 - training_data.json
 - training_data.spacy
 - website_evaluation_data.txt
 - website_testing_data.txt
 - website_training_data.txt
 - documentation
 - src
 - __pycache__
 - extract_website_data.py
 - main.py
 - venv
 - test.py

The main notebook area displays a code cell with the following Python code:

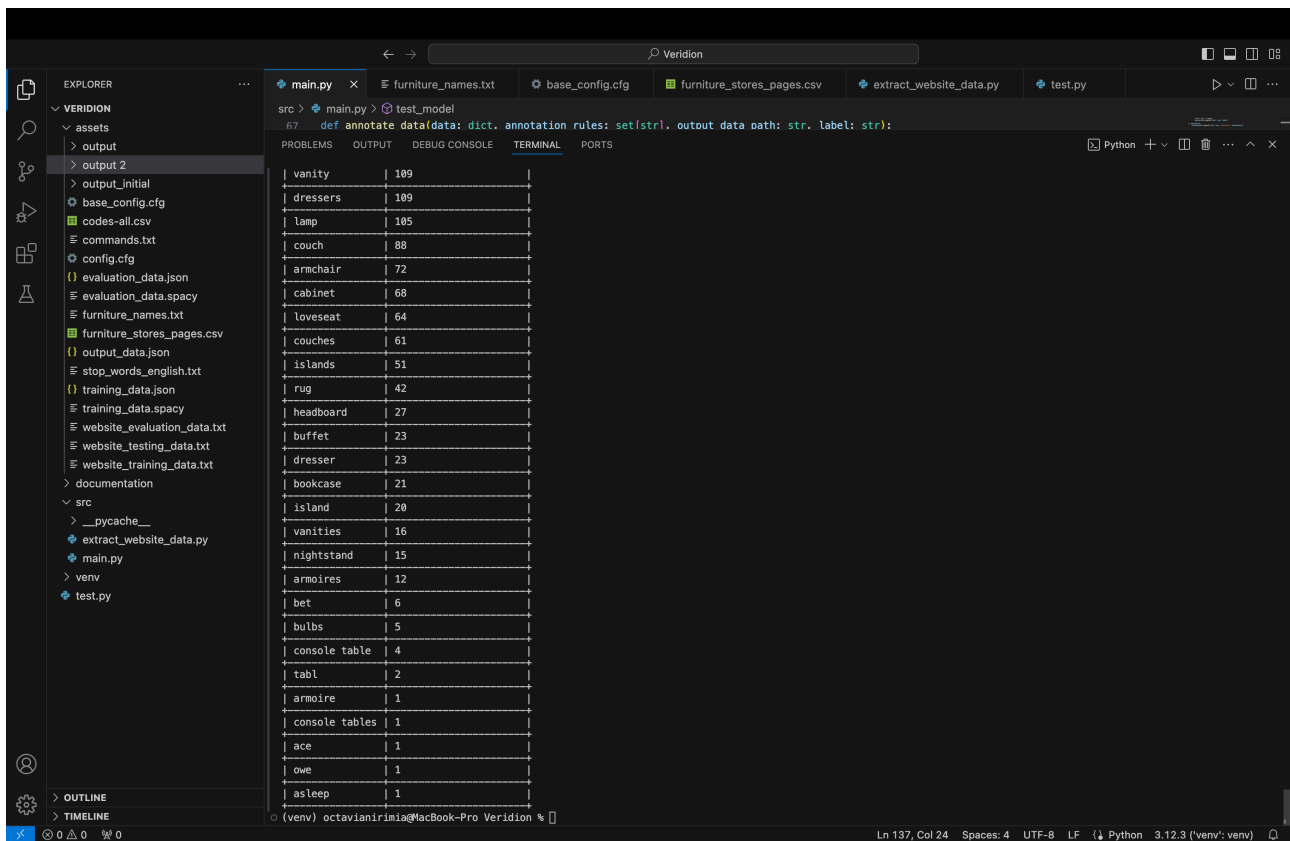
```
src > main.py > test_model
67 def annotate data:data: dict, annotation rules: set[str], output_data_path: str, label: str):
```

Below the code cell, a data cell shows a list of URLs related to furniture products and services, such as:

- <https://www.muuduufurniture.com/product/abrianna-brown-sofa/>: couch bed mattress couches table ottomans loveseat chairs sofa cabinets sofas beds tables
- <https://www.espasso.com/collection/78/>: console table nightstands armchairs ottomans benches chairs sofas desks tables
- <https://www.canvastextiles.com/desk-nightstand-table-ottomans-benches-chairs-dressers-sofa-island-chair-beds-lamp-tables-rugs>
- <https://www.furniturecontracts.com.au/product-category/office-sitstand-adjustable-height-desks/>: table buffets ottoman cabinets chairs desks bench tables
- <https://www.muuduufurniture.com/product/3pcs-taupe-microfiber-leather-sofa-set/>: couch mattress bed couches table ottomans loveseat chairs cabinets sofa sofas chair b
- <https://www.georgestreet.co.uk/delivery/>: mattresses bed mattress armchairs benches chairs sofa lamps headboards sofas beds tables rugs
- <https://www.thefurniturefactory.org.uk/products/warwick-dining-set/>: bookcases mattresses armchairs table ottomans benches sofa chairs cabinets headboards sofas dresse
- <https://www.onlydiningchairs.com.au/collections/mid-century-dining-chairs/>: armchair bet table chairs chair tables
- https://www.angliareliners.co.uk/wp-content/uploads/2019/07/4748_02.jpg
- <https://www.insarfa.com/pages/certifications/>: dresser mattresses bench benches chairs cabinets sofa rug sofas lamps desks beds lamp tables rugs
- <https://www.furniturecontracts.com.au/product-category/education-school-timber-frame-chairs/>: buffets bench table chairs cabinets ottoman desks chair tables
- <https://www.couchpotato.com.au/collections/in-stock-quickship/constraint-quickship/>: couch table benches chairs sofa rug sofas desks chair tables rugs
- <https://www.innerpace.net.au/products/ARTEK-ROCKEY-STOOL-34.htm>: chairs ottomans tables
- <https://www.espasso.com/collection/all?designers=171>: armchair nightstands armchairs ottomans benches chairs sofas desks tables
- <https://www.furniturecontracts.com.au/product-category/steel-frame-dining-chairs/>: table bench buffets ottoman cabinets chairs desks chair tables
- <https://www.onlydiningchairs.com.au/collections/luxury-dining-chairs/>: chairs chair table tables
- <https://www.espasso.com/collection/37/>: nightstands armchairs ottomans benches chairs sofas desks tables
- <https://www.modishstore.com/collections/trophy-heads/>: nightstands loveseat desks buffets sofa cabinets chair lamps tables table ottomans chairs lamp bookcases vanity
- <https://www.houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
- <https://www.georgestreet.co.uk/furniture/dining-room/benches/>: mattresses armchairs bench benches sofa chairs headboards sofas beds tables
- <https://houseofall.com/products/gift-card/>: mattresses bed table sofa cabinets chair beds
- <https://www.modholic.com/living-room/>: nightstands loveseat nightstand table ottomans ottoman chairs sofa chair tables
- <https://www.skandium.com/products/ch24-soft/>: chair sofa table desk
- <https://www.blandkandisel.co.za/products/catalogue/cushions-throws/emerald-ikat-cushion/>: bookcases tables bed armchairs vanity table ottomans benches sofa chairs cab
-

The screenshot shows a PyCharm IDE with a Python script named 'main.py' open. The script defines a function 'def annotate_data(data: dict, annotation rules: set[str], output data path: str, label: str):'. The IDE interface includes a file explorer on the left, a terminal at the bottom, and a table of furniture items and their occurrence counts.

Furniture	Number of occurrences
chairs	842
tables	745
sofas	596
table	569
sofa	441
chair	404
benches	402
ottomans	380
beds	373
armchairs	354
desks	353
cabinets	327
nightstands	223
rugs	219
headboards	216
bed	201
mattresses	195
buffets	177
lamps	175
desk	157
bench	152
bookcases	145
ottoman	144
mattress	139
loveseats	121
vanity	109



10. Provocări

- Site-uri web inaccesibile - provocare pe care am depășit-o destul de ușor utilizând un bloc try - except și afișând eroarea în terminal;
- Curățarea textului - a fost inițial un pas dificil însă am trecut bine peste el cu ajutorul expresiilor regulate, care e o metodă eficientă în curățarea textului, și cu ajutorul celor două fișiere enunțate anterior;
- Adnotarea datelor - inițial am ales o metodă mult mai complicată pentru adnotarea datelor, mult mai puțin eficientă, însă am realizat la un moment dat că pot utiliza expresii regulate ca la curățarea textului;
- Alegerea bibliotecii pentru ner - deși am avut o recomandare către două biblioteci nu am reușit să le configurez pentru sistemul meu așa că am decis să caut o altă soluție și anume biblioteca spacy;
- Antrenarea modelului - nefiind familiarizat cu biblioteca utilizată am petrecut destul de mult timp documentând-mă despre aceasta și despre modul de antrenare al unui model.

11. Concluzie

În această proiect, am prezentat un model pentru extragerea produselor de pe site-urile magazinelor de mobilă. Modelul a primit ca intrare o listă de URL-uri din fișierul “furniture_stores_pages.csv” și a fost capabil să extragă numele produselor de pe paginile care conțineau mobilă, chiar și în situația în care unele URL-uri erau nevalide sau nu conțineau mobilă.