*Weboldalak adatainak gépi feldolgozása*

(Automated processing of website data)

Anonim Hallgató

A weboldalak (pl. arukereso.hu) adatainak gépi feldolgozása napjaink egyik legfontosabb és leggyakrabban alkalmazott technológiája az adattudomány és az üzleti intelligencia területén. A különböző online áruházak, összehasonlító oldalak, közösségi platformok hatalmas mennyiségű információt szolgáltatnak, amelyeket manuálisan feldolgozni idő- és munkaigényes lenne. Ezért elengedhetetlen a webscraping és automatizált adatkinyerés alkalmazása, amely képes strukturált formában gyűjteni és feldolgozni a webes tartalmakat.

(vö. [https://miau.my-x.hu/mediawiki/index.php/FARAD-DAYS /](https://miau.my-x.hu/mediawiki/index.php/FARAD-DAYS%20/) <https://www.arukereso.hu/hutoszekreny-fagyaszto-c3168/fn:termekek-osszehasonlitasa:zanussi-zrb33103wa-p272930431,gorenje-rk4172anx-p408880113,beko-rcsa-330k31-w-p359309879,gorenje-rk4172anw-p408615135,indesit-lr7s2w-p341919636,indesit-lr6-s2-x-p347571984,indesit-lr6-s2-w-p347571982,beko-csa270m30w-p383036065,candy-cm-3354-x-p421328934,candy-cm-3354-w-p421328928/>)

Jelen dolgozatban bemutatott megoldás egy Selenium-alapú webscraper program, amely az Arukereso.hu videókártya összehasonlító oldaláról gyűjt adatokat, majd azokat Excel fájlban tárolja, ezzel támogatva a további elemzéseket és döntéstámogatást.

**A program működésének részletes bemutatása**

**1. Böngészőindítás és konfiguráció**

A program egy Chrome böngészőt indít el a Selenium WebDriver segítségével. A böngésző nem headless módban fut, ezáltal a futás során láthatóvá válik a böngészőablak, ami megkönnyíti a fejlesztők számára a hibakeresést és az adatkinyerési folyamat nyomon követését. A képernyőfelbontás 1920x1080 pixelre van állítva, biztosítva a weboldal optimális megjelenítését.

**2. Weboldal betöltése**

A böngésző megnyitja a megadott URL-címet, amely az Arukereso.hu videókártya összehasonlító aloldala. Ez a lépés kulcsfontosságú, hiszen a program így hozzáfér az oldal teljes tartalmához, beleértve a dinamikusan, JavaScript segítségével generált elemeket is.

**3. Oldalforrás mentése a hibakereséshez**

A program a teljes HTML-oldalforrást egy fájlba menti (debug\_page.html), amely később szolgálhat referenciaként a lekért adatok ellenőrzéséhez, illetve a scraping folyamat hibáinak felderítéséhez.

**4. Cookie-értesítés kezelése**

Az Arukereso.hu oldal sütikkel kapcsolatos értesítést jeleníthet meg a látogatók számára. A program automatikusan megkeresi az elfogadás gombot, és ha megtalálja, aktiválja azt. Ez a lépés biztosítja, hogy az értesítés ne zavarja az oldal többi elemének betöltődését és adatkinyerését.

**5. Várakozás az oldalelemek betöltődésére**

Mivel az összehasonlító termékek listája JavaScript segítségével töltődik be, a program explicit várakozást alkalmaz a terméknevek megjelenésére. Ez garantálja, hogy az adatok kinyerése csak akkor kezdődik el, ha az oldal már teljesen betöltődött.

**6. Termékek adatainak kigyűjtése**

A program kinyeri a termékek nevét, URL-jét és a hozzájuk tartozó képek linkjeit a weboldal DOM-jából. Ez a strukturált adatgyűjtés lehetővé teszi, hogy később a termékek azonosíthatóak és megjeleníthetőek legyenek.

**7. Specifikációk lekérése**

A program az összehasonlító táblázat sorait vizsgálja meg, és minden paraméter esetén kigyűjti az egyes termékekhez tartozó értékeket. Hiányzó adat esetén helyettesítő karaktert (-) alkalmaz, így az adatok táblázatos formája konzisztens marad.

**8. Adatok rendszerezése és tárolása**

Az összegyűjtött adatokat egy Python dictionary struktúrában rendezi, amelyet egy pandas.DataFrame objektumba konvertál. Ez az adatstruktúra könnyen exportálható és továbbfeldolgozható.

**9. Adatok exportálása Excel fájlba**

Az adatokat a program Excel fájlba (gpu\_comparison.xlsx) menti, amely széles körben elfogadott és könnyen kezelhető formátum. Ez megkönnyíti az adatok megosztását, elemzését és vizualizációját.

**10. Folyamat dokumentálása képernyőkép segítségével**

A sikeres adatkinyerést követően a program képernyőképet ment a böngészőről, amely hasznos lehet a folyamat dokumentálásához és a hibakereséshez.

**11. Böngésző erőforrásainak felszabadítása**

A program a végén bezárja a Chrome böngészőt, ezzel felszabadítja az erőforrásokat, és biztonságosan lezárja az alkalmazást.

**A program előnyei**

* **Hatékonyság és automatizáltság:** A folyamat emberi beavatkozás nélkül képes nagy mennyiségű adat összegyűjtésére.
* **Dinamikus tartalom kezelése:** A Selenium képes JavaScript által generált oldalak kezelésére is, szemben a hagyományos HTTP kérésekkel.
* **Strukturált adatexport:** Az Excel fájl megkönnyíti a további adatfeldolgozást és elemzést.
* **Rugalmas alkalmazhatóság:** A módszer adaptálható más weboldalakra és más típusú adatok kinyerésére is.

Microsoft Visual Studio Solution File, Format Version 12.00

# Visual Studio Version 17

VisualStudioVersion = 17.14.36121.58 d17.14

MinimumVisualStudioVersion = 10.0.40219.1

Project("{888888A0-9F3D-457C-B088-3A5042F75D52}") = "Python", "Python\Python.pyproj", "{637227DA-7242-48C5-95AB-2A4EEDA9ECFD}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{637227DA-7242-48C5-95AB-2A4EEDA9ECFD}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{637227DA-7242-48C5-95AB-2A4EEDA9ECFD}.Release|Any CPU.ActiveCfg = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {B955FB4E-2F5F-4F0D-8BE9-AEBD8DFD3EDE}

EndGlobalSection

EndGlobal



Lépés száma Folyamat lépése Mit csinál a program? Megjegyzés / Eredmény

1 Böngésző indítása ChromeDriver elindítása fejlesztői módban (nem headless) Megnyílik egy Chrome ablak

2 Weboldal betöltése Az adott URL megnyitása (az összehasonlító oldal) Megjelenik a teljes weboldal

3 HTML mentése Az oldal HTML kimentése debug\_page.html fájlba Segít hibakeresésnél

4 Cookie elfogadása Az elfogadás gomb megkeresése és kattintás Ha nincs gomb, ezt átugorja

5 Termékek betöltésének várása Várakozás, amíg betöltődnek a termékek nevei Így biztos, hogy vannak adatok a következő lépésekhez

6 Terméknevek kinyerése A DOM-ból kikeresi az összes termék nevét és linkjét Kinyeri a termékek nevét és URL-jét

7 Képek URL-jeinek lekérése Lekéri a termékekhez tartozó képek URL-jét Képek linkjei későbbi letöltéshez vagy megjelenítéshez

8 Specifikációk kinyerése Az összehasonlító sorok (paraméterek) kinyerése és tárolása Pl. memória méret, órajel, stb.

9 Adatok összerendezése Az összes adat összefűzése egy táblázatba (dictionary) Később DataFrame-be alakítás

10 Mentés CSV vagy Excel fájlba Az adatokat CSV vagy Excel fájlba írja Később Excel-ben megnyitható táblázat