

✓ 📊 중고 노트북 가격 크롤링 및 분석 프로젝트

✓ 문제 인식

- 중고 제품 가격이 판매자 주관에 따라 결정되다 보니, 합리적인 구매 판단이 어렵습니다.
- 신뢰할 수 있는 가격 데이터가 실시간으로 정리되어 있지 않아 구매자/판매자 모두 기준 설정이 힘들다.

✓ 사용 기술 및 도구 선정 이유

도구	사용 목적	선정 이유
Selenium	웹 자동화 및 데이터 크롤링	로그인 없이 접근 가능한 플랫폼에 적합
Fake UserAgent	봇 탐지 회피	반복 크롤링 시 차단 방지
pandas	데이터 정제 및 분석	표 형태로 다루기 편리
matplotlib, seaborn	데이터 시각화	가격 분포를 직관적으로 전달
Google Colab	실습 및 데모 환경	웹 기반으로 즉시 실행 가능하고 설치 부담 없음

✓ 기대 효과

- 지역별, 브랜드별 중고 노트북 시세를 빠르게 파악 가능
- 평균 가격 기준으로 **과하게 비싼 제품 혹은 저렴한 매물**을 필터링하여 정보 제공
- 사용자 맞춤형 자동화 분석 도구로 확장 가능 (예: 특정 브랜드 알림, 가격 추이 분석 등)

✓ 🛠️ 개발 환경 준비: 필수 라이브러리 설치

다음 !pip 명령어는 프로젝트 실행에 필요한 Python 패키지를 설치하기 위한 코드입니다.

```
1 !pip install selenium chromedriver_autoinstaller fake_useragent gradio matplo
```

↩

```
Downloading wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/py
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-pi
Downloading selenium-4.31.0-py3-none-any.whl (9.4 MB)
_____ 9.4/9.4 MB 26.1 MB/s eta 0:00:00
Downloading chromedriver_autoinstaller-0.6.4-py3-none-any.whl (7.6 kB)
Downloading fake_useragent-2.2.0-py3-none-any.whl (161 kB)
_____ 161.7/161.7 kB 9.2 MB/s eta 0:00:00
Downloading gradio-5.28.0-py3-none-any.whl (54.1 MB)
_____ 54.1/54.1 MB 8.7 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
_____ 322.9/322.9 kB 9.5 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
_____ 95.2/95.2 kB 5.0 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
_____ 11.5/11.5 MB 27.3 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
_____ 72.0/72.0 kB 4.3 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading trio-0.30.0-py3-none-any.whl (499 kB)
_____ 499.2/499.2 kB 12.8 MB/s eta 0:00:00
Downloading trio_websocket-0.12.2-py3-none-any.whl (21 kB)
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
_____ 62.5/62.5 kB 4.8 MB/s eta 0:00:00
Downloading ffmpeg-0.5.0-py3-none-any.whl (6.0 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Installing collected packages: pydub, wsproto, uvicorn, tomlkit, semantic-ver
Successfully installed aiofiles-24.1.0 chromedriver_autoinstaller-0.6.4 fake_
```

✓ 크롬 브라우저 & 드라이버 설치 (Google Colab 전용)

Colab 환경은 일반 PC와 달리 크롬 브라우저나 드라이버가 설치되어 있지 않기 때문에, 웹 자동화를 위해 수동으로 설치해줘야 합니다.

다음은 크롬과 드라이버를 설치하고, Selenium이 인식할 수 있도록 경로를 등록하는 코드입니다.

코드 설명

코드	설명
<code>!apt-get update</code>	시스템 패키지 목록을 최신 상태로 업데이트
<code>!apt install chromium-browser chromium-chromedriver</code>	크롬 브라우저와 크롬드라이버 설치
<code>!ln -sf ...</code>	드라이버를 Colab의 기본 실행 경로에 심볼릭 링크

```
import os
os.environ['PATH'] += ...
```

운영체제(OS) 기능을 사용하기 위한 모듈 임포트
크롬과 드라이버 경로를 시스템 환경 변수에 등록

```
1 !apt-get update > /dev/null
2 !apt install chromium-browser chromium-chromedriver > /dev/null
3 !ln -sf /usr/bin/chromedriver /usr/local/bin/chromedriver
4
5 import os
6 os.environ['PATH'] += ':/usr/lib/chromium-browser:/usr/bin/'
```

W: Skipping acquire of configured file 'main/source/Sources' as repository '[h](#)
WARNING: apt does not have a stable CLI interface. Use with caution in script:

✓ 🔍 중고 노트북 가격 수집 및 분석 (코드 요약)

1. 환경 설정 및 크롬 드라이버 자동화

- Selenium, FakeUserAgent, chromedriver_autoinstaller 등을 이용해 웹 브라우저 자동화 환경 구성
- 크롤링 지역은 '신당동', 스크롤은 3페이지까지 설정

2. 데이터 수집 (크롤링)

- 당근마켓에서 "노트북" 키워드로 검색된 게시글 정보를 수집
- 제목, 가격, 지역, 링크 정보를 추출
- Selenium으로 페이지 스크롤하면서 동적으로 게시글 목록 크롤링

3. 데이터 정제 및 전처리

- convert_price() 함수를 통해 가격 텍스트에서 숫자만 추출
- convert_date() 로 "몇일 전" 등의 텍스트를 실제 날짜로 변환
- 삼성, LG, 그램, 레노버, ASUS 등 브랜드 키워드가 포함된 모델명 추출

4. 데이터프레임 변환 및 분석

- 크롤링한 데이터를 pandas 의 DataFrame 으로 변환
- 평균 가격 계산 후, 평균보다 20% 이상 비싼 제품 / 싼 제품으로 나눠 필터링

5. 결과 출력

- 상위 가격 상품(Above Average)과 하위 가격 상품(Below Average) 각각 표로 출력
- 수집된 전체 데이터를 CSV로 저장 (daangn_laptops.csv)

6. 파일 다운로드 (Colab 환경)

- Google Colab 환경에서 자동으로 CSV 파일 다운로드 링크 제공

```
1 import time
2 import re
3 import datetime
4 import pandas as pd
5
6 from selenium import webdriver
7 from selenium.webdriver.common.by import By
8 from selenium.webdriver.chrome.options import Options
9 from selenium.webdriver.chrome.service import Service
10 from selenium.webdriver.support.ui import WebDriverWait
11 from selenium.webdriver.support import expected_conditions as EC
12 from fake_useragent import UserAgent
13
14 import chromedriver_autoinstaller
15
16 region = '신당동'
17 max_page = 3
18
19 ua = UserAgent()
20 options = Options()
21 options.add_argument('--headless')
22 options.add_argument('--no-sandbox')
23 options.add_argument('--disable-dev-shm-usage')
24 options.add_argument(f'user-agent={ua.random}')
25
26 chromedriver_autoinstaller.install()
27 driver = webdriver.Chrome(options=options)
28
29 region_param = region.replace(' ', '-') + '-28'
30 base_url = f'https://www.daangn.com/kr/buy-sell/?in={region_param}&search=노인'
31 driver.get(base_url)
32
33 SCROLL_PAUSE_TIME = 2
34 data = []
35
36 for _ in range(max_page):
37     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
38     time.sleep(SCROLL_PAUSE_TIME)
39
40     try:
41         WebDriverWait(driver, 5).until(
42             EC.presence_of_element_located((By.CSS_SELECTOR, 'a[data-gtm="se
43             )
44     except:
45         print("❌ 게시물 로딩 실패")
46         continue
47
48     items = driver.find_elements(By.CSS_SELECTOR, 'a[data-gtm="search_article
49     for item in items:
50         try:
```

```

51         title = item.find_element(By.CSS_SELECTOR, 'span.lm809sh').text
52         price_text = item.find_element(By.CSS_SELECTOR, 'span.lm809si').
53         region_elem = item.find_element(By.CSS_SELECTOR, 'span.lm809sj')
54         link = 'https://www.daangn.com' + item.get_attribute('href')
55         data.append([title, price_text, region_elem, link])
56     except:
57         continue
58
59 driver.quit()
60
61 def convert_price(price):
62     price = re.sub(r'^0-9', '', price)
63     return int(price) if price else None
64
65 def convert_date(text):
66     now = datetime.datetime.now()
67     if '일 전' in text:
68         days = int(re.search(r'(\d+)일 전', text).group(1))
69         return now - datetime.timedelta(days=days)
70     elif '시간 전' in text:
71         hours = int(re.search(r'(\d+)시간 전', text).group(1))
72         return now - datetime.timedelta(hours=hours)
73     else:
74         return now
75
76 records = []
77 for title, price, date, link in data:
78     model_match = re.findall(r'(삼성|LG|그램|레노버|ASUS|NT[0-9]+)', title)
79     model = model_match[0] if model_match else '기타'
80     records.append({
81         'title': title,
82         'model': model,
83         'price': convert_price(price),
84         'date': convert_date(date),
85         'url': link
86     })
87
88 df = pd.DataFrame(records)
89 print("총 게시글 수:", len(df))
90
91 if not df.empty:
92     print("🔪 예시:", df.iloc[0])
93 else:
94     print("❗ 크롤링 결과가 비어 있음")
95
96 df.dropna(subset=['price'], inplace=True)
97
98 avg_price = df['price'].mean()
99 threshold_upper = avg_price * 1.2
100 threshold_lower = avg_price * 0.8
101
102 above_avg = df[df['price'] > threshold_upper]
103 below_avg = df[df['price'] < threshold_lower]
104
105 print("\n👉 평균 이상 가격 상품:")

```

```
106 display(above_avg[['title', 'price', 'url']])
107
108 print("\n🔪 평균 이하 가격 상품:")
109 display(below_avg[['title', 'price', 'url']])
110
111 df.to_csv("daangn_laptops.csv", index=False)
112 print("✅ CSV 파일 저장 완료: daangn_laptops.csv")
113
114 from google.colab import files
115 files.download("daangn_laptops.csv")
```



	title	price	url
0	레노버 게이밍 노트북 판 매합니다.(리전5pro)(풀 박)	1000000.0	https://www.daangn.comhttps://www.daangn.com/k...
2	레노버 노트북 팝니다	400000.0	https://www.daangn.comhttps://www.daangn.com/k...
7	삼성 ■ 노트북 NT900X4C-A99	250000.0	https://www.daangn.comhttps://www.daangn.com/k...
12	HP 노트북 probook450	247000.0	https://www.daangn.comhttps://www.daangn.com/k...
17	아수스 젠북 UM425IA- AM001 라이젠 노트북	280000.0	https://www.daangn.comhttps://www.daangn.com/k...
19	삼성 노트북	244000.0	https://www.daangn.comhttps://www.daangn.com/k...
21	노트북	600000.0	https://www.daangn.comhttps://www.daangn.com/k...
22	인텔 듀얼 모니터 노트북 램 16gb/롬 256gb	1100000.0	https://www.daangn.comhttps://www.daangn.com/k...
25	델 게이밍 노트북 Dell G16 32GB 1TB	1200000.0	https://www.daangn.comhttps://www.daangn.com/k...
34	MSI 모던 14 노트북	340000.0	https://www.daangn.comhttps://www.daangn.com/k...
36	rog strix g713rm 3060 노트북 팝니다	650000.0	https://www.daangn.comhttps://www.daangn.com/k...
39	MSI Cyborg 15 A12VF 노 트북	1100000.0	https://www.daangn.comhttps://www.daangn.com/k...
41	서피스 랩탑4 13.5인치/라 이젠5/8GB/256GB	350000.0	https://www.daangn.comhttps://www.daangn.com/k...
53	Asus x560u 노트북	250000.0	https://www.daangn.comhttps://www.daangn.com/k...
55	ASUS 에이수스 비보북 고 15 OLED (E1504FA- R7525T) 노트북 판...	320000.0	https://www.daangn.comhttps://www.daangn.com/k...
57	lg 노트북 거의 새거	600000.0	https://www.daangn.comhttps://www.daangn.com/k...
60	레노버 게이밍 노트북 판 매합니다.(리전5pro)(풀 박)	1000000.0	https://www.daangn.comhttps://www.daangn.com/k...
62	레노버 노트북 팝니다	400000.0	https://www.daangn.comhttps://www.daangn.com/k...
67	삼성 ■ 노트북 NT900X4C-A99	250000.0	https://www.daangn.comhttps://www.daangn.com/k...
72	HP 노트북 probook450	247000.0	https://www.daangn.comhttps://www.daangn.com/k...

12	HP 노트북 probook450	247000.0	https://www.daangn.com https://www.daangn.com/k...
77	아수스 젠북 UM425IA-AM001 라이젠 노트북	280000.0	https://www.daangn.com https://www.daangn.com/k...
79	삼성 노트북	244000.0	https://www.daangn.com https://www.daangn.com/k...
81	노트북	600000.0	https://www.daangn.com https://www.daangn.com/k...
82	인텔 듀얼 모니터 노트북 램 16gb/롬 256gb	1100000.0	https://www.daangn.com https://www.daangn.com/k...
85	델 게이밍 노트북 Dell G16 32GB 1TB	1200000.0	https://www.daangn.com https://www.daangn.com/k...
94	MSI 모던 14 노트북	340000.0	https://www.daangn.com https://www.daangn.com/k...
96	rog strix g713rm 3060 노트북 팝니다	650000.0	https://www.daangn.com https://www.daangn.com/k...
99	MSI Cyborg 15 A12VF 노트북	1100000.0	https://www.daangn.com https://www.daangn.com/k...
101	서피스 랩탑4 13.5인치/라이젠5/8GB/256GB	350000.0	https://www.daangn.com https://www.daangn.com/k...
113	Asus x560u 노트북	250000.0	https://www.daangn.com https://www.daangn.com/k...
115	ASUS 에이수스 비보북 고 15 OLED (E1504FA-R7525T) 노트북 판...	320000.0	https://www.daangn.com https://www.daangn.com/k...
117	lg 노트북 거의 새거	600000.0	https://www.daangn.com https://www.daangn.com/k...
120	레노버 게이밍 노트북 판 매합니다.(리전5pro)(풀 박)	1000000.0	https://www.daangn.com https://www.daangn.com/k...
122	레노버 노트북 팝니다	400000.0	https://www.daangn.com https://www.daangn.com/k...
127	삼성  노트북 NT900X4C-A99	250000.0	https://www.daangn.com https://www.daangn.com/k...
132	HP 노트북 probook450	247000.0	https://www.daangn.com https://www.daangn.com/k...
137	아수스 젠북 UM425IA-AM001 라이젠 노트북	280000.0	https://www.daangn.com https://www.daangn.com/k...
139	삼성 노트북	244000.0	https://www.daangn.com https://www.daangn.com/k...
141	노트북	600000.0	https://www.daangn.com https://www.daangn.com/k...
142	인텔 듀얼 모니터 노트북 램 16gb/롬 256gb	1100000.0	https://www.daangn.com https://www.daangn.com/k...
145	델 게이밍 노트북 Dell G16 32GB 1TB	1200000.0	https://www.daangn.com https://www.daangn.com/k...
154	MSI 모던 14 노트북	340000.0	https://www.daangn.com https://www.daangn.com/k...
156	rog strix g713rm 3060 노트북 팝니다	650000.0	https://www.daangn.com https://www.daangn.com/k...
159	MSI Cyborg 15 A12VF 노트북	1100000.0	https://www.daangn.com https://www.daangn.com/k...

161	서피스 랩탑4 13.5인치/라 이젠5/8GB/256GB	350000.0	https://www.daangn.com https://www.daangn.com/k...
173	Asus x560u 노트북	250000.0	https://www.daangn.com https://www.daangn.com/k...
175	ASUS 에이수스 비보북 고 15 OLED (E1504FA- R7525T) 노트북 판...	320000.0	https://www.daangn.com https://www.daangn.com/k...
177	lg 노트북 거의 새거	600000.0	https://www.daangn.com https://www.daangn.com/k...

 평균 이하 가격 상품:

title	price	url
-------	-------	-----



시각화 구성 및 의미

Gradio를 통해 시각적으로 출력되는 구성 요소는 다음과 같습니다:



Price Histogram (가격 히스토그램)

: 가격 분포를 보여주는 그래프

- 각 중고 노트북 가격이 어느 범위에 몰려 있는지 시각적으로 확인할 수 있음
- 정규분포형인지, 치우쳐 있는지 등을 파악 가능
- 예: 대부분의 노트북이 30~50만원대에 분포한다면, 평균 가격이 적정하다는 신뢰가 생김



Price Boxplot (가격 박스플롯)

: 이상치와 가격 범위를 한눈에 볼 수 있는 그래프

- 최소값, 최대값, 사분위수(Q1, Q2, Q3)를 시각적으로 확인 가능
- 특히 극단적으로 비싼 제품이나 싸게 올라온 매물(outliers)을 빠르게 파악할 수 있음
- 예: 200만원 이상 노트북이 박스 밖으로 튀어나온다면 해당 매물은 비정상적일 수 있음



Top 10 Above-Average Price Listings (평균 이상 상위 10개 상품)

: 평균 가격보다 비싼 매물 TOP 10 리스트

- 가격이 평균보다 20% 이상 높은 상품만 필터링
- 프리미엄 제품, 신제품급 중고, 허위매물 의심 항목 등을 선별할 때 유용



Top 10 Below-Average Price Listings (평균 이하 하위 10개 상품)

: 평균 가격보다 저렴한 매물 TOP 10 리스트

- 가격이 평균보다 20% 이상 낮은 매물
- 가성비 좋은 제품, 빠른 거래를 원하는 매물 등 실구매자에게 유용한 정보 제공
- 예: 평균가 70만 원일 때, 50만 원 이하 제품만 표시됨



요약

시각화 항목	설명	목적
가격 히스토그램	가격 분포 시각화	평균가, 중심 범위 파악
가격 박스플롯	이상치 및 범위 표현	극단값 및 신뢰도 판단
평균 이상 TOP 10	비싼 매물 선별	프리미엄/비정상 매물 확인
평균 이하 TOP 10	저렴한 매물 선별	가성비 추천 또는 급매 탐색

```
1 import gradio as gr
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```