(비 **NCS)**머신러닝 데이터 수집과 분석 시각화

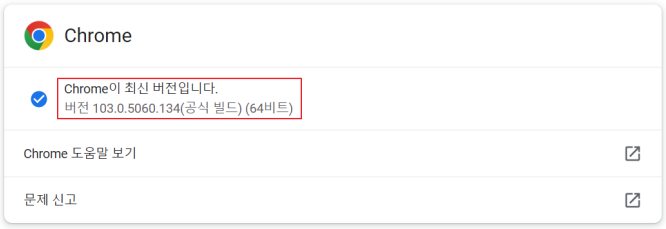
비트캠프 KDT 5기 김지혜

주제: 셀프 주유소는 저렴한가?

Selenium 사용하기
selenium 설치 - pip install selenium
from selenium import webdriver

크롬 드라이버 설치
크롬 드라이버란 크롬 브라우저를 제어하기 위한 드라이버이다.
다운로드 사이트에서 자신의 크롬버전 및 운영체제와 동일한 드라이버를 다운로드한다.

참조 https://chromedriver.chromium.org/downloads



Current Releases

- If you are using Chrome version 104, please download ChromeDriver 104.0.5112.29
- If you are using Chrome version 103, please download ChromeDriver 103.0.5060.134
- If you are using Chrome version 102, please download ChromeDriver 102.0.5005.61
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

현재 크롬버전은 103이므로 ChromeDriver 103으로 다운로드 받는다.

다운로드 받은 파일을 해당 프로젝트에 넣는다.

```
import time
from glob import glob
import pandas as pd
from matplotlib import font_manager, rc # 한글 시각화 패키지 설치
from selenium import webdriver
import matplotlib.pyplot as plt
import seaborn as sns
```

이 예제를 하기 위한 라이브러리이다.

- 데이터 크롤링

```
driver = webdriver.Chrome('./data/chromedriver.exe')
driver.get('https://www.opinet.co.kr/searRgSelect.do')
driver.get("http://www.opinet.co.kr/searRgSelect.do")
```

selenium 라이브러리에서 webdriver.Chrome함수를 사용하여 드라이버를 로드한다.

```
driver.find_element_by_id("SIDO_NM0").send_keys('서울특별시')
gu_list_raw =
driver.find_element_by_id("SIGUNGU_NM0")
time.sleep(4)
gu_list =
gu_list_raw.find_elements_by_tag_name('option')
# tag가 option
gu_names = [option.get_attribute("value") for option in gu_list] # option의 value들을 하나의 리스트로 생성
gu_names.remove('') # 생성된 리스트에서 빈 공백 제거
print(gu_names)
```

Opinet 사이트에서 사용자의 위치에 따라지역을 잡아주어 현재 서울로 위치하여 데이터가 크롤링된다.

```
for cnt in range(len(gu_names)):
  second_list_raw =
  driver.find_element_by_id("SIGUNGU_NM0")
   #id SIGUNGU_NM0가 순차적으로 들어간다.
  second_list_raw.send_keys(gu_names[cnt])
  # 키 입력을 차례대로 선택
  time.sleep(5)
  file_down =
  driver.find_element_by_id('glopopd_excel').click()
  # 25개의 피일을 순차적으로 다운로드
driver.close()
```

위 코드는 자동 반복으로 다운로드 하기 위한 것이다.
이 코드를 실행하면 엑셀파일 25개가 다운로드 된다.

```
def data_embedding(self):
      merged_list = glob('./data/지역*xls')
      # 생성한 엑셀파일 한 리스트에 모으기
      #print(merged_list)
```

```
    list_tabel = []  # 엑셀 내용을 담을 리스트
    for file_name in merged_list:
        tmp = pd.read_excel(file_name, header=2)
      # xls 파일을 편집하기 위해서 데이터프레임으로
      생성
        list_tabel.append(tmp) # list_tabel 리스트에 넣기
    #print(list_tabel)  # 25개의 테이블이 저장된 리스트
    total_gas_station = pd.concat(list_tabel)
    # 25개의 테이블을 하나의 리스트구조로 반환
    print(total_gas_station)
    return total_gas_station
```

생성한 엑셀파일을 glob 라이브러리를 사용하여 한 리스트에 모으고 각 엑셀 파일을 for문으로 풀어 데이터프레임으로 생성한다. concat 함수를 사용하여 25개의 테이블을 하나의 리스트구조로 반환한다.

```
      지역                                상호                                주소  ...   휘발유   경유  실내등유
0   서울특별시  재건에너지 재정제2주유소 고속셀프지점  서울특별시 강동구  천호대로 1246 (둔촌제2동)  ...  2065  2065     -
1   서울특별시                  구천면주유소          서울 강동구 구천면로 357 (암사동)  ...  2093  2117     -
2   서울특별시         (주)소모에너지 신월주유소         서울 강동구 양재대로 1323 (성내동)  ...  2125  2135  1600
3   서울특별시       지에스칼텍스㈜ 동서울주유소         서울 강동구 천호대로 1456 (상일동)  ...  2127  2105     -
4   서울특별시     현대오일뱅크㈜직영 명일셀프주유소           서울 강동구 고덕로 168 (명일동)  ...  2133  2163     -
..     ...                               ...                               ...  ...   ...   ...   ...
31  서울특별시         (주)소모에너지 펜트힐주유소                 서울 강남구 삼성로 335  ...  2495  2363     -
32  서울특별시                갤러리아주유소                서울 강남구 압구정로 426  ...  2495  2398  1750
33  서울특별시         (주)만정에너지 삼보주유소          서울 강남구 봉은사로 433 (삼성동)  ...  2638  2558  1778
34  서울특별시                  삼성주유소          서울 강남구 삼성로 521 (삼성동)  ...     -     -     -
35  서울특별시                  동우주유소      서울특별시 강남구  봉은사로 311 (논현동)  ...     -     -     -
```

- 데이터 전처리

```
def preprocessing(self):
    total_gas_station = self.data_embedding()
    gas_station = pd.DataFrame({'주유소명': # 스키마
이름 변경
    total_gas_station['상호'],\
    '경유가격': total_gas_station['경유'],\
    '셀프': total_gas_station['셀프여부'],\
    '브랜드': total_gas_station['상표'],\
    '주소': total_gas_station['주소']})
    print(gas_station)
    print(gas_station.info())
```

크롤링한 데이터의 스키마를 변경한다.

```
                주유소명  경유가격 셀프      브랜드                                주소
0   재건에너지 재정제2주유소 고속셀프지점  2065  Y  현대오일뱅크  서울특별시 강동구  천호대로 1246 (둔촌제2동)
1                 구천면주유소  2117  N  현대오일뱅크          서울 강동구 구천면로 357 (암사동)
2        (주)소모에너지 신월주유소  2135  N   GS칼텍스         서울 강동구 양재대로 1323 (성내동)
3      지에스칼텍스㈜ 동서울주유소  2105  Y   GS칼텍스          서울 강동구 천호대로 1456 (상일동)
4    현대오일뱅크㈜직영 명일셀프주유소  2163  Y  현대오일뱅크            서울 강동구 고덕로 168 (명일동)
..                 ...  ... ..      ...                               ...
31       (주)소모에너지 펜트힐주유소  2363  N   GS칼텍스                 서울 강남구 삼성로 335
32              갤러리아주유소  2398  N   SK에너지                서울 강남구 압구정로 426
33       (주)만정에너지 삼보주유소  2558  N   GS칼텍스          서울 강남구 봉은사로 433 (삼성동)
34                삼성주유소     -  N   SK에너지           서울 강남구 삼성로 521 (삼성동)
35                동우주유소     -  N   SK에너지      서울특별시 강남구  봉은사로 311 (논현동)
```

```
[446 rows x 5 columns]
<class 'pandas.core.frame.DataFrame'>
Int64Index: 446 entries, 0 to 35
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   주유소명    446 non-null    object
 1   경유가격    446 non-null    object
 2   셀프      446 non-null    object
 3   브랜드     446 non-null    object
 4   주소      446 non-null    object
dtypes: object(5)
memory usage: 20.9+ KB
None
```

info() 함수를 사용하면 데이터의 정보를 확인할 수 있다.

```
gas_station = gas_station[gas_station['경유가격'] != '-']
#경유가격이 없는 데이터 삭제
print(gas_station.info())
```

경유가격이 -로 되어 있는, 즉 경유가격이 없는 데이터를 삭제한다.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 435 entries, 0 to 33
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   주유소명    435 non-null    object
 1   경유가격    435 non-null    object
 2   셀프      435 non-null    object
 3   브랜드     435 non-null    object
 4   주소      435 non-null    object
dtypes: object(5)
memory usage: 20.4+ KB
None
```

가격 없는 데이터를 제거하기 전 데이터 정보와 비교를 하면 11개의 데이터가 삭제된 것을 확인 할 수 있다.

```
gas_station['경유가격'] = [float(value) for value in
gas_station['경유가격']]
# 가격 정보를 실수형으로 변환
# print(gas_station.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 435 entries, 0 to 33
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   주유소명    435 non-null    object
 1   경유가격    435 non-null    float64
 2   셀프      435 non-null    object
 3   브랜드     435 non-null    object
 4   주소      435 non-null    object
dtypes: float64(1), object(4)
memory usage: 20.4+ KB
None
```

가격 데이터를 float 형으로 변경한다.

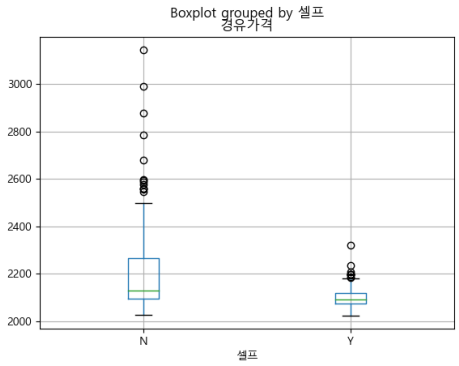
```
gas_station.reset_index(inplace=True)
#print(gas_station.head())
return gas_station
```

	index	주유소명	경유가격	셀프	브랜드	주소
0	0	재건에너지 재정제2주유소 고속셀프지점	2065.0	Y	현대오일뱅크	서울특별시 강동구 천호대로 1246 (둔촌제2동)
1	1	구천면주유소	2117.0	N	현대오일뱅크	서울 강동구 구천면로 357 (암사동)
2	2	(주)소모에너지 신월주유소	2135.0	N	GS칼텍스	서울 강동구 양재대로 1323 (성내동)
3	3	지에스칼텍스㈜ 동서울주유소	2185.0	Y	GS칼텍스	서울 강동구 천호대로 1456 (상일동)
4	4	현대오일뱅크㈜직영 명일셀프주유소	2163.0	Y	현대오일뱅크	서울 강동구 고덕로 168 (명일동)

.reset_index() 함수를 사용하여 전처리한 데이터를 재정렬한다.

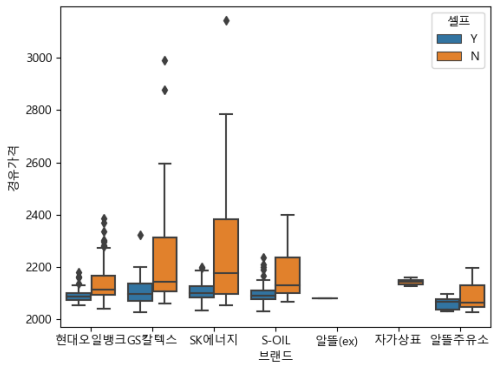
```
def visualization(self):
        self.korean_print() # 폰트 설정
        gas_station = self.preprocessing()
       #처리한 데이터 가져옴
        gas_station.boxplot(column='경유가격', by='셀프')
       # 셀프 vs 비셀프 가격 비교
        plt.show()
```

셀프 주유소는 정말 저렴한지 boxplot으로 확인



비셀프는 셀프에 비해 가격 평균과 분산이 크고 outlier가 많이 존재한다. 즉, 비셀프의 경우 주유소가 가지고 있는 환경에 따라 가격이 심하게 차이가 날 수 있다는 추측을 해볼 수 있다.

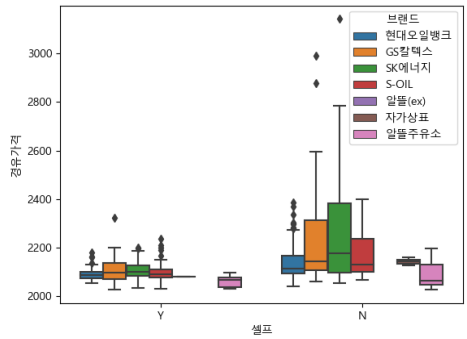
```
sns.boxplot(x='브랜드', y='경유가격', hue='셀프', data=gas_station)
# 브랜드별 가격 분포
plt.show()
```



주류 주유 브랜드의 가격 분포는 알뜰 주우소보다 더 큰 가격 분포를 가지고 있으며, outlier 또한 많이 존재한다.

그리고 특히 SK 에너지가 가장 넓은 분포의 주유 값과 가장 큰 가장 큰 주유값의 주유소를 가지고 있다.

```
sns.boxplot(x='셀프', y='경유가격', hue='브랜드', data=gas_station)
# 브랜드별 셀프 vs 비 셀프 가격 비교
plt.show()
```



브랜드별 셀프와 비셀프의 가격을 비교할 수 있다. 비셀프에서 SK 에너지의 가격이 제일 비싼것을 알 수 있다.

```
def minmax(self):
    gas_station = self.preprocessing()
    print(gas_station.sort_values(by='경유가격',
    ascending=False).head(10))
    # 최고가격 10곳
```

	index	주유소명	경유가격	셀프	브랜드	주소
278	9	서남주유소	3143.0	N	SK에너지	서울 중구 통일로 30
244	12	서계주유소	2990.0	N	GS칼텍스	서울특별시 용산구 청파로 367 (청파동)
277	8	필동주유소	2879.0	N	GS칼텍스	서울 중구 퇴계로 196 (필동2가)
242	10	한석주유소	2785.0	N	SK에너지	서울 용산구 이촌로 164
243	11	동자동주유소	2680.0	N	SK에너지	서울 용산구 한강대로104길 6 (동자동)
274	5	신당동주유소	2598.0	N	SK에너지	서울 중구 다산로 242 (신당동)
220	26	지에스칼텍스(주) 국회대로주유소	2595.0	N	GS칼텍스	서울 영등포구 국회대로 746 (여의도동)
276	7	약수주유소	2589.0	N	GS칼텍스	서울 중구 다산로 173
241	9	(주)남경주유소	2578.0	N	SK에너지	서울 용산구 녹사평대로11길 24
218	24	KH여의도주유소	2560.0	N	SK에너지	서울 영등포구 국회대로 794 (여의도동)

해당 코드로 최고 가격인 10곳의 주유소를 확인할 수 있다.

```
print(gas_station.sort_values(by='경유가격',
ascending=True).head(10))
# 최저가격 10곳 출력
```

	index	주유소명	경유가격	셀프	브랜드	주소
196	2	지에스칼텍스㈜ 화일주유소	2025.0	Y	GS칼텍스	서울 영등포구 도림로 415
279	0	뉴신정주유소	2027.0	N	알뜰주유소	서울 강서구 곰달래로 207 (화곡동)
172	3	현대주유소	2028.0	Y	S-OIL	서울 양천구 남부순환로 372 (신월동)
169	0	양천구주유소	2031.0	Y	알뜰주유소	서울 양천구 국회대로 275 (목동)
280	1	목화주유소	2031.0	Y	알뜰주유소	서울 강서구 국회대로 251 (화곡동)
281	2	지에스칼텍스㈜ 경인고속주유소	2032.0	Y	GS칼텍스	서울 강서구 국회대로 225 (화곡동)
173	4	개나리Self주유소	2033.0	Y	SK에너지	서울 양천구 남부순환로 442 (신월동)
174	5	형산석유(주)원주유소	2038.0	N	현대오일뱅크	서울 양천구 남부순환로 408
282	3	화곡역주유소	2039.0	Y	알뜰주유소	서울 강서구 강서로 154 (화곡동)
56	3	지에스칼텍스(주)홍제동주유소	2044.0	Y	GS칼텍스	서울 서대문구 통일로 372

해당 코드로 최저 가격인 10곳의 주유소를 확인할 수 있다.