```
# 사용할 라이브러리

import pandas as pd
import numpy as np
import os
import warnings
warnings.filterwarnings(action='ignore')

import datetime
import importlib
import json
from datetime import datetime

import webbrowser

!apt install chromium-chromedriver
!pip install selenium

from selenium import webdriver
options = webdriver.ChromeOptions()
options.add_argument('--headless')
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')

from bs4 import BeautifulSoup
import numpy as np
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
chromium-chromedriver is already the newest version (85.0.4183.83-0ubuntu0.18.
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
Requirement already satisfied: selenium in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: urllib3 in /usr/local/lib/python3.6/dist-packag
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call
```

```
# 크롤링에 사용할 함수 구현

genre_dic = {}

def makeUrl(dateList,i, kinds):
  if kinds == 'playstore':
    url = 'https://www.mobileindex.com/app/get_rank_all?rt=r&mk=2&c=kr&t=app&rs=100
  else:
    url = 'https://www.mobileindex.com/app/get_rank_all?rt=r&mk=1&c=kr&t=app&rs=100
  return url

def getGenreUrl(url):
  driver = webdriver.Chrome('chromedriver',options=options)
```

```
  driver.get(url)

  html = driver.page_source
  soup = BeautifulSoup(html, 'html.parser')
  genreURL = soup.select('div.item-info > a')
  return genreURL

def get_genre(url,appname):
  if appname in genre_dic:
    return genre_dic[appname]
  dr = webdriver.Chrome('chromedriver',options=options)
  dr.get(url)
  ht = dr.page_source
  so = BeautifulSoup(ht, 'html.parser')
  genre = so.select('tr.text-center')[0].text
  index1 = genre.find('\n',1)
  if genre[index1+1] == '\n':
    genre_dic[appname] = '기타'
    return '기타'
  index2 = genre.find('\n',14)
  genre_dic[appname] = genre[(index1+1):index2]
  return genre[(index1+1):index2]

def separateAppRank(rank):
  free_rank = []
  pay_rank = []
  sales_rank = []
  for j in range(len(rank)):
    if len(rank[j]) == 100:
      free_rank.append(rank[j][0::2])
    elif len(rank[j]) == 150 or len(rank[j]) == 300:
      free_rank.append(rank[j][0::3])
  return free_rank

def getAppRank(url, year):
  driver = webdriver.Chrome('chromedriver',options=options)
  driver.get(url)

  html = driver.page_source
  soup = BeautifulSoup(html, 'html.parser')
  notices = soup.select('span.appname')

  tmp = []
  for n in notices:
    tmp.append(n.text.strip())
  return tmp


# 크롤링

from tqdm import tqdm
#2019
appRank_p = []
appRank_a = []
date_2019 = ['2019-01-31','2019-02-28','2019-03-31','2019-04-30','2019-05-31','2019
```

```
date_size = len(date_2019)
kinds = ['playstore']


for kind in kinds:
  appRank = []
  appGenre = []
  for i in tqdm(range(date_size)):
    url = makeUrl(date_2019,i,kind)
    tmp = getAppRank(url,2019)
    appRank.append(tmp)
    genreURL = getGenreUrl(url)
    genreList = []
    for j in range(len(genreURL)):
      genreList.append(get_genre('https://www.mobileindex.com/'+genreURL[j]['href']
    appGenre.append(genreList)
    print(kind,' data of ',date_2019[i],' is done')
  if kind == 'playstore':
    appRank_p = appRank[:]

  p_appRank_free_2019 = separateAppRank(appRank)
  print('p_appRank is done')
  p_appGenre_free_2019 = separateAppRank(appGenre)
  print('p_appGenre is done')



#2020
appRank = []
date_2020 = ['2020-01-31','2020-02-29','2020-03-31','2020-04-30','2020-05-31','2020
date_size = len(date_2020)
kinds = ['playstore']

for kind in kinds:
  appRank = []
  appGenre = []
  for i in tqdm(range(date_size)):
    url = makeUrl(date_2020,i,kind)
    tmp = getAppRank(url,2020)
    appRank.append(tmp)
    genreURL = getGenreUrl(url)
    genreList = []
    for j in range(len(genreURL)):
      genreList.append(get_genre('https://www.mobileindex.com/'+genreURL[j]['href']
    appGenre.append(genreList)
    print(kind,' date of ',date_2020[i],' is done')

  p_appRank_free_2020 = separateAppRank(appRank)
  print('p_appRank is done')
  p_appGenre_free_2020  = separateAppRank(appGenre)
  print('p_appGenre is done')
```

```
  0%|          | 0/12 [00:00<?, ?it/s]
  8%|▊         | 1/12 [07:30<1:22:40, 450.96s/it]playstore  data of  2019-01-3
 17%|█▋        | 2/12 [09:47<59:26, 356.64s/it]  playstore  data of  2019-02-
 25%|██▌       | 3/12 [11:53<43:06, 287.34s/it]playstore  data of  2019-03-31
 33%|███▎      | 4/12 [13:31<30:45, 230.69s/it]playstore  data of  2019-04-3
 42%|████▏     | 5/12 [14:23<20:39, 177.12s/it]playstore  data of  2019-05-3
 50%|█████     | 6/12 [16:08<15:32, 155.47s/it]playstore  data of  2019-06-
 58%|█████▊    | 7/12 [21:26<17:00, 204.02s/it]playstore  data of  2019-07-
 67%|██████▋   | 8/12 [23:52<12:26, 186.64s/it]playstore  data of  2019-08-
 75%|███████▌  | 9/12 [25:16<07:48, 156.08s/it]playstore  data of  2019-09-
 83%|████████▎ | 10/12 [26:55<04:37, 138.97s/it]playstore  data of  2019-1
 92%|█████████▏| 11/12 [28:18<02:02, 122.08s/it]playstore  data of  2019-1
100%|██████████| 12/12 [29:35<00:00, 147.97s/it]

  0%|          | 0/9 [00:00<?, ?it/s]playstore  data of  2019-12-31  is done
p_appRank is done
p_appGenre is done

 11%|█         | 1/9 [01:31<12:14, 91.83s/it]playstore  date of  2020-01-31
 22%|██▏       | 2/9 [04:00<12:42, 109.00s/it]playstore  date of  2020-02-29
 33%|███▎      | 3/9 [05:41<10:38, 106.35s/it]playstore  date of  2020-03-31
 44%|████▍     | 4/9 [07:10<08:25, 101.16s/it]playstore  date of  2020-04-30
 56%|█████▌    | 5/9 [08:27<06:15, 93.95s/it] playstore  date of  2020-05-31
 67%|██████▋   | 6/9 [09:44<04:26, 88.83s/it]playstore  date of  2020-06-30
```

```
 78%|███████▊  | 7/9 [10:43<02:40, 80.03s/it]playstore  date of  2020-07-31

 89%|████████▉ | 8/9 [12:24<01:26, 86.41s/it]playstore  date of  2020-08-3

100%|██████████| 9/9 [13:47<00:00, 91.91s/it]playstore  date of  2020-09-1
p_appRank is done
p_appGenre is done
```

```
# 크롤링 -> csv로 변환

p_appRank_free = list()
for i in range(len(p_appRank_free_2019)):
  p_appRank_free.append(p_appRank_free_2019[i])
  p_appRank_free.append(p_appGenre_free_2019[i])

col = []
for i in range(12):
  col.append('19_'+str(i+1))
  col.append('19_'+str(i+1)+'_gen')

appRank_free_playstore_df = pd.DataFrame(p_appRank_free)
appRank_free_playstore_df = appRank_free_playstore_df.T
appRank_free_playstore_df.columns = col
appRank_free_playstore_df.to_csv('/content/gdrive/My Drive/빅콘/app/appRank_free_pla

p_appRank_free = list()
for i in tqdm(range(len(p_appRank_free_2020))):
  p_appRank_free.append(p_appRank_free_2020[i])
  p_appRank_free.append(p_appGenre_free_2020[i])

col = []
for i in tqdm(range(9)):
  col.append('20_'+str(i+1))
  col.append('20_'+str(i+1)+'_gen')

appRank_free_playstore_df = pd.DataFrame(p_appRank_free)
appRank_free_playstore_df = appRank_free_playstore_df.T
appRank_free_playstore_df.columns = col
appRank_free_playstore_df.to_csv('/content/gdrive/My Drive/빅콘/app/appRank_free_pla
```

```
100%|██████████| 9/9 [00:00<00:00, 57896.83it/s]

100%|██████████| 9/9 [00:00<00:00, 64527.75it/s]
```

```
pd.read_csv('/content/gdrive/My Drive/빅콘/app/appRank_free_playstore_2019.csv')
```

	Unnamed: 0	19_1	19_1_gen	19_2	19_2_gen	19_3	19_3_gen	19
0	0	국세청 홈택스	도구	Magic Booster - 제일 좋아하는 청리 도구	도구	Samsung Notes	생산성	배달의
1	1	TikTok 틱톡	동영상 플레이어/편집기	배달요기요	식음료	NH스마트뱅킹	금융	SRT - 고속 (NE
2	2	Netflix(넷플릭스)	엔터테인먼트	커넥츠: 무료 공부질문 앱-500만 다운로드 (문제해설, 노하우, 과외, 인강)	교육	삼성 음성 녹음	도구	Super Ma Cleaner 드로이드 리너 휴 쿨러 안 이
3	3	NH스마트뱅킹	금융	카카오톡 KakaoTalk	커뮤니케이션	카카오톡 KakaoTalk	커뮤니케이션	네이 NAV
4	4	카카오뱅크 - 같지만 다른 은행	금융	Super Junk Cleaner - 휴대폰 쿨러, 안드로이드 클리너, 배터리 절약	도구	토스	금융	카카 Kakao
5	5	카카오톡 KakaoTalk	커뮤니케이션	네이버 지도, 내비게이션	여행 및 지역 정보	카카오뱅크 - 같지만 다른 은행	금융	NH스마
6	6	V3 Mobile Plus 2.0	도구	V3 Mobile Plus 2.0	도구	네이버 - NAVER	도서/참고자료	국세청 홈
7	7	원더쇼핑: 인기쇼핑몰/핫딜/여행쿠폰/웹툰/위메프 투어/위메프페이	쇼핑	네이버 - NAVER	도서/참고자료	Samsung Smart Switch Mobile	도구	Sams Sm Sw Mo
8	8	쿠팡 (Coupang)	쇼핑	NH스마트뱅킹	금융	V3 Mobile Plus 2.0	도구	SODA 소 Natu Bea Cam
				Samsung		미세미세 - 미세먼지		

9	9	네이버 - NAVER	도서/참고자료	Smart Switch Mobile	도구	미세먼지, WHO기준, 알람, 위젯, 날씨, 지도	날씨	(Coupa
10	10	Samsung Smart Switch Mobile	도구	쿠팡 (Coupang)	쇼핑	Samsung Members	도구	당근마 우리 동네 고 직거래 룩장
11	11	NAVER VIBE (바이브)	음악/오디오	한컴오피스 viewer	생산성	바이오인증 공동앱	금융	Turbo V -무제한 무 VPN & 배 보안 프로
12	12	배달의민족	식음료	T map 택시 - 티맵택시,T맵택시	지도/내비게이션	쿠팡 (Coupang)	쇼핑	V3 Mo Plus
13	13	당근마켓 - 우리 동네 중고 직거래 벼룩장터	쇼핑	나침반	도구	BNK부산은행 모바일 뱅킹	금융	카카오뱅 같지만 다 은
14	14	위메프 - 특가대표 (특가 / 쇼핑 / 쇼핑앱 / 쿠폰 / 배송)	쇼핑	TikTok 틱톡	동영상 플레이어/편집기	한컴오피스 viewer	생산성	Netflix(넷 릭
15	15	SODA 소다 - Natural Beauty Camera	사진	카카오뱅크 - 같지만 다른 은행	금융	한국투자증권 (스마트폰 계좌개설)	금융	네이버 지 내비게이
16	16	Instagram	소셜	당근마켓 - 우리 동네 중고 직거래 벼룩장터	쇼핑	밴드	소셜	배달요
17	17	뱅크샐러드	금융	Netflix(넷플릭스)	엔터테인먼트	TikTok 틱톡	동영상 플레이어/편집기	TikTok 틱
18	18	KB국민은행 스타뱅킹	금융	카카오 T - 택시, 대리운전, 주차, 내비, 카풀, 바이크	지도/내비게이션	네이버 지도, 내비게이션	여행 및 지역정보	Instagr
19	19	한컴오피스 viewer	생산성	Instagram	소셜	Droid-X III 백신 (기업용)	도구	C
				SODA 소다 - Natural				Ulike유라 크 - 스타

20	20	배달요기요	식음료	Natural Beauty Camera	사진	배달요기요	식음료	고급 필터 티 카
21	21	토스	금융	Google Play 게임	엔터테인먼트	당근마켓 - 우리 동네 중고 직거래 벼룩장터	쇼핑	4sha
22	22	네이버 파파고 - AI 통번역	도구	국세청 홈택스	도구	Instagram	소셜	Faceb
23	23	네이버 지도, 내비게이션	여행 및 지역 정보	배달의민족	식음료	AliExpress - 스마트한 쇼핑, 더즐거운생활	쇼핑	
24	24	밴드	소셜	BNK부산은행 모바일 뱅킹	금융	Facebook	소셜	카카오 KakaoP
25	25	SSG.COM	쇼핑	카카오맵 - 대한민국 No.1 지도 앱 (지도 / 내비게이션 / 대중교통 / 로드뷰)	여행 및 지역 정보	SODA 소다 - Natural Beauty Camera	사진	일기 2020 - 간
26	26	Facebook	소셜	밴드	소셜	카카오맵 - 대한민국 No.1 지도 앱 (지도 / 내비게이션 / 대중교통 / 로드뷰)	여행 및 지역 정보	카카오 대한 No.1 지 (지도 / 게이션 중교통 드
27	27	튜브박스 - 음악 동영상 다운로드	동영상 플레이어/편집기	Total Cleaner	도구	네이버 파파고 - AI 통번역	도구	
28	28	Google Play 게임	엔터테인먼트	Intra	도구	마이 홈플러스	라이프스타일	메가 (MEGAB
		Cut Cut - 사진편집어						카카오 택시, 대