

A large pile of plastic waste, including bottles, bags, and containers, is shown in the foreground. In the background, a large industrial building with a corrugated metal facade is visible. The building has a logo and Korean text on it. A green semi-transparent box is overlaid on the center of the image, containing text.

i:i 에이콘아카데미

대소형 폐기물 이미지 분류

4조 녹색지구대

목 차

1. 프로젝트 소개

- 주제 선정 이유 및 목적
- 프로젝트 기간 및 개발환경

2. 업무 분담

3. 데이터 설명

- R데이터 및 이미지 데이터

4. R분석 보고서

5. 웹사이트 구현 및 설명

6. 모델 생성 및 Django 코드 설명

7. 한계점 및 보완점

8. Q & A

1. 프로젝트 소개





이건 버리는데
얼마지?

이폐기물은
어디다 버려야 하지?

뉴스 > IT/의학 > 정보통신

"열심히 한 분리수거, 재활용 안돼.." 헛수고?

"지금의 재활용 문제는 지난 20년 태만의 대가"

[인터뷰] 자원순환사회경제연구소 홍수열 소장

HOME > 사회

"한국, 분리수거 선진국 맞지만...질 낮은게 문제"

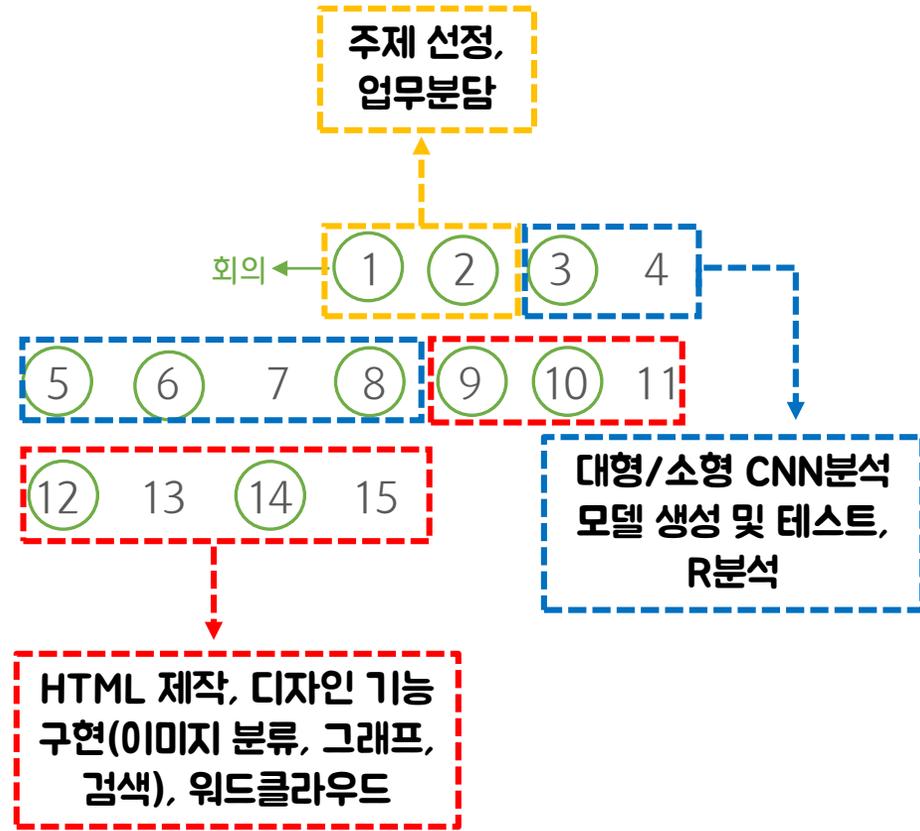
[개발목적]

대형폐기물은 분리수거업체에 문의를 통해 스티커를 붙여야 하는
수고가 있고 소형폐기물은 분리수거를 해야 하는지,
일반쓰레기인지 **헛갈리는** 경우가 많다.
이러한 불편함을 해소, 건전한 분리수거 문화 정착,
환경오염 감소, 일정 이상의 경제효과 창출 등
수많은 기대효과를 창출할 수 있다.

프로젝트 기간 및 개발환경

[프로젝트 기간]

9월 01일~9월 15일(총 15일)



[개발환경]



2. 업무 분담



대형 폐기물

이경* (팀장)

프로젝트총괄
대형폐기물 이미지수집 및 크롤링
대형 이미지데이터 전처리 및 모델링
R을 활용한 분석
Django 제작

문규*

대형 폐기물 이미지수집 및 크롤링
대형 이미지데이터 전처리 및 모델링
R을 활용한 분석
Django 제작

권기*

PPT 제작
대형 폐기물 이미지수집 및 크롤링
대형 이미지데이터 전처리 및 모델링
R을 활용한 분석 및 보고서 작성
환경 관련 통계 시각화
Django 제작

소형 폐기물

최은* (부팀장)

PPT 제작
소형 폐기물 이미지수집
소형 이미지데이터 전처리 및 모델링
환경 관련 통계 시각화
Django 제작

이용*

소형 폐기물 이미지수집
소형 이미지데이터 전처리 및 모델링
Django 제작

김혜*

크롤링 및 워드클라우드 제작
소형 폐기물 이미지수집
소형 이미지데이터 전처리 및 모델링
Django 제작

효율적인 인력 운영을 위해 대형 폐기물 3명, 소형 폐기물 3명으로 배치

참고) 완전 다른 팀이 아닌 정보와 방법론 공유 및 프로젝트 공동 진행하며 최종 결과물은 함께 수행

3. 데이터 설명



4. R 분석 보고서



개요

- **연구 동기** : 폐기물 배출량의 지역, 소득, 교육수준 간 차이에 관한 분석 목적
- **연구 목적** : 폐기물 재활용 증대 및 폐기물 관련 정책 참고 자료
- **연구 내용** : 인구, 1인당 총생산, 교육 수준에 따른 서울 지역구 별 폐기물량 분석
- **분석 단위** : 서울 25개 지역구 전수 자료(폐기물, 1인당 총생산, 교육 수준, 인구)
- **연구 방법** : R을 이용한 다중공선성, 단순회귀분석, 다중회귀분석
- **레퍼런스**
 - '폐기물 배출량의 지역간 차이에 관한 분석 (이용우, 1998, '대한지리학회지 제33권 2호 209~224')
 - 다중회귀분석을 이용한 경기도 폐기물 발생량 예측 (2019년도 추계학술연구발표회 논문집)

```
> head(garbage, 3)
  dist people  GRDP highedu Gamount
2 종로구   161 173117   0.50   284.3
3 중구     128 371717   0.44   434.3
4 용산구   228 43216   0.53   240.1
```

데이터 요약 설명

- people : 서울시 각 구별 인구수(단위 천명)
- GRDP : 각 구별 1인당 총 생산
- Highedu : 각 구별 고학력자(대학 이상 졸업자) 비율
- Gamount : 각 구별 폐기물 총 양(톤/일)

변수 분류

- 종속 변수 : Gamount 각 구별 폐기물 총 양
- 독립 변수 : People 각 구별 인구수
 GRDP 각 구별 1인당 총 생산
 Highedu 각 구별 고학력자 비율

분석

```
test_model <- lm(Gamount ~ people + GRDP + highedu, data = garbage)
```

```
> vif(test_model)
  people      GRDP  highedu
1.864249 1.678848 1.314620
> reduced.model=step(test_model,direction = "both")
Start:  AIC=210.75
Gamount ~ people + GRDP + highedu
```

| | Df | Sum of Sq | RSS | AIC |
|-----------|----|-----------|--------|--------|
| <none> | | | 83189 | 210.75 |
| - highedu | 1 | 24868 | 108057 | 215.29 |
| - GRDP | 1 | 59120 | 142309 | 222.17 |
| - people | 1 | 64610 | 147799 | 223.12 |

요약

- VIF 결과(분산팽창지수 결과)

3개의 독립변수 모두 vif 값이 10을 넘지 않으므로 다중공선성을 가지고 있지 않다. 따라서, 독립변수 간 상관관계가 없다고 확인 할 수 있다.

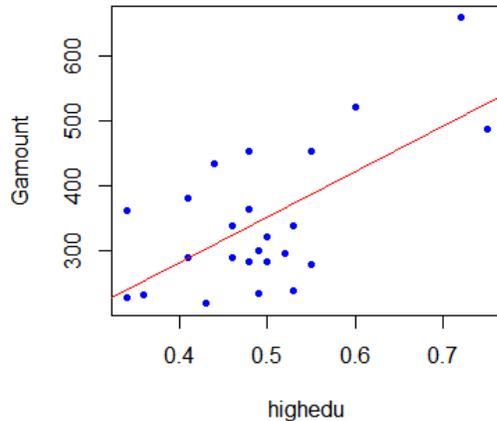
- 모델 선택 방법 결과

변수 선택 방법으로는 Stepwise를 선택하였다. 결과적으로 3개의 변수 모두 선택 된 것을 확인할 수 있다.

분석

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.76      87.52   0.032 0.975111
garbage$highedu  698.24    174.64   3.998 0.000565 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 83.21 on 23 degrees of freedom
Multiple R-squared:  0.41,    Adjusted R-squared:  0.3844
F-statistic: 15.98 on 1 and 23 DF,  p-value: 0.0005652
```



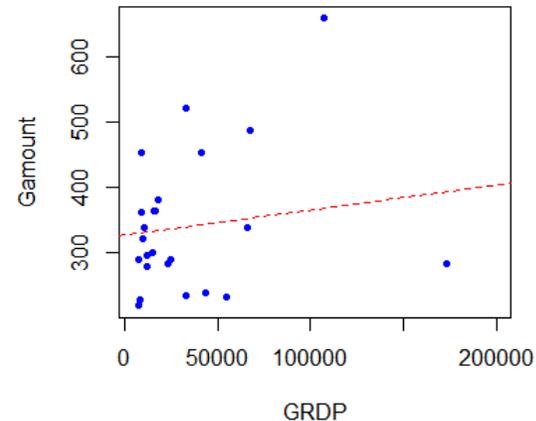
단순 회귀분석 요약(폐기물/교육수준)

- p-value: 0.0005652 < 0.05

귀무가설 기각 : 학력 별 폐기물 배출량의 차이가 있다.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.282e+02  2.458e+01  13.353 2.55e-12 ***
garbage$GRDP 3.809e-04  2.745e-04   1.387  0.179
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104.1 on 23 degrees of freedom
Multiple R-squared:  0.07723,    Adjusted R-squared:  0.03711
F-statistic: 1.925 on 1 and 23 DF,  p-value: 0.1786
```



단순 회귀분석 요약(폐기물/인당 총 생산)

- p-value: 0.1786 > 0.05

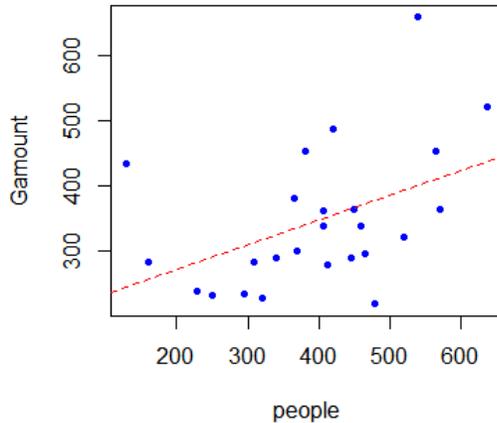
귀무가설 채택 : 1인당 총 생산 별 폐기물 배출량의 차이가 없다.

분석

```

Coefficients:
(Intercept)      Estimate Std. Error t value Pr(>|t|)
garbage$people    0.3790   0.1569   2.415  0.02410 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

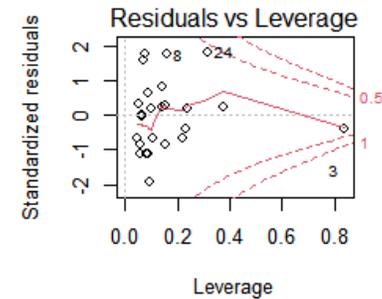
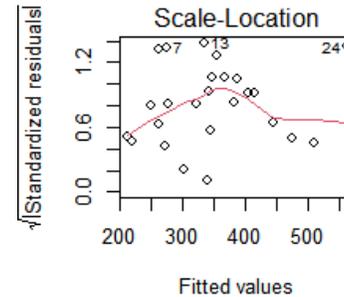
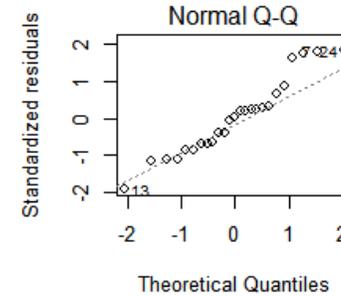
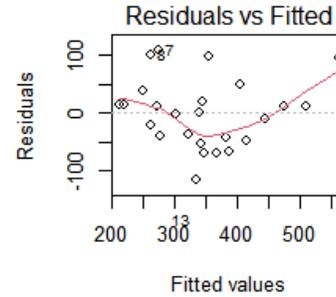
Residual standard error: 96.76 on 23 degrees of freedom
Multiple R-squared:  0.2022,    Adjusted R-squared:  0.1676
F-statistic: 5.831 on 1 and 23 DF,  p-value: 0.0241
    
```



단순 회귀분석 요약(폐기물/지역별 인구)

- p-value: 0.0241 < 0.05

귀무가설 기각 : 인구 별 폐기물 배출량의 차이가 있다.



다중 회귀분석 요약

- 왼쪽 상단 그래프에서 선형성 가정 만족
- 오른쪽 위의 그래프에서 잔차의 정규성 만족
- 왼쪽 아래 그림에서 등분산성 만족(잔차의 분산이 일정)
- 다소 이상치가 존재함

분석

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.035e+02  7.117e+01  -1.454 0.160636
people      5.629e-01  1.394e-01   4.039 0.000593 ***
GRDP       8.312e-04  2.151e-04   3.863 0.000901 ***
highedu    3.795e+02  1.515e+02   2.506 0.020532 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 62.94 on 21 degrees of freedom
Multiple R-squared:  0.6918,    Adjusted R-squared:  0.6478
F-statistic: 15.71 on 3 and 21 DF,  p-value: 1.377e-05
```

다중 회귀분석 요약

- p-value: $1.377e-05 < 0.05$
귀무가설 기각 : 인구/인당 총생산/학력 별 폐기물 양의 차이가 있다.

```
> AIC(model, model3, model4, mul_model)
      df      AIC
model  3 295.9295
model3  3 307.1121
model4  3 303.4726
mul_model  5 283.6967
```

AIC(모형 비교)

- AIC 모형 비교 결과 다중 회귀분석(mul_model) 모델 값이 제일 낮게 나타난다.
- AIC값이 가장 낮은 다중 회귀분석 모델이 제일 적합하다고 판단 된다.

결론

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.035e+02  7.117e+01 -1.454 0.160636
people       5.629e-01  1.394e-01  4.039 0.000593 ***
GRDP         8.312e-04  2.151e-04  3.863 0.000901 ***
highedu      3.795e+02  1.515e+02  2.506 0.020532 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 62.94 on 21 degrees of freedom
Multiple R-squared:  0.6918,    Adjusted R-squared:  0.6478
F-statistic: 15.71 on 3 and 21 DF,  p-value: 1.377e-05
```

- 다중회귀분석 공식은 $y = -1.035e+02 + 5.629e-01 * \text{people} + 8.312e-04 * \text{GRDP} + 3.795e+02 * \text{highedu}$
- 다른 변수를 보정하였을 때 people은 약 0.5629 증가하였고, GRDP는 약 0.0008312 증가하였으며, highedu는 약 379.5 증가한 것을 확인 할 수가 있다.
- 모든 독립 변수는 p-value가 유의수준 0.05보다 작으므로 통계적으로 유의하다.

부족한 점

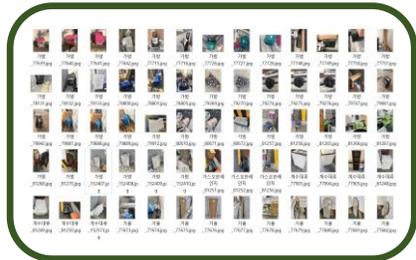
- 2015년도 데이터 이므로 다소 현재와 차이가 있을 수 있다는 점과 최신 데이터를 구하지 못한다는 점이 제일 아쉬운 점으로 꼽힌다.
- 데이터 양의 부재 (1개 년도만 할 수 밖에 없었다는 점)
- 데이터가 각각 수집된 데이터이며, 면적대비 인구수의 보정이 없어서 다양한 편향성(Bias)이 발생 될 수 있다는 점

5. 웹사이트 시연

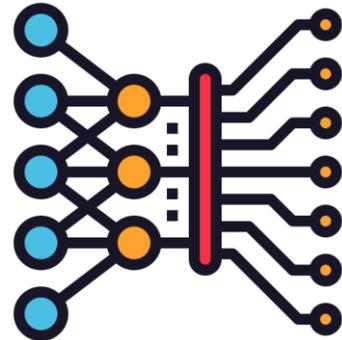


6. 코드 설명

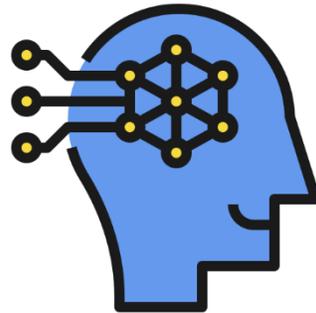




이미지 데이터



CNN



모델 생성



예측 이미지



결과 출력

< 데이터 전처리 >

```

train_dir = "500data/train"
categories = ["can", "glass", "paper", "plastic", "st", "vinyl"]
# 파일명 : can 22, paper 21, st 16, vinyl 15
nb_classes = len(categories)

image_w = 64
image_h = 64
X = [] # 사진 배열화 시킨거
y = [] # label [1 0 0 0 0 0] ...

for idx, cate in enumerate(categories):
    # one-hot
    label = [0 for i in range(nb_classes)]
    label[idx] = 1

    img_dir = train_dir + "/" + cate
    files = glob.glob(img_dir + "/*.jpg")
    #print(cate, " 파일 갯수:", len(files))

    for i, f in enumerate(files): # i(순서) f(파일경로명)
        img = Image.open(f)
        img = img.convert("RGB")
        img = img.resize((image_w, image_h))
        data = np.asarray(img) # 배열화

        X.append(data)
        y.append(label)

X = np.array(X) # 4차원으로!
y = np.array(y)

```

< train / test 분리 >

```

X_train, X_test, y_train, y_test = train_test_split(X, y)
xy = (X_train, X_test, y_train, y_test)
np.save("data/train_data_1.npy", xy)

```

→ 파일로 저장

< 정확도 향상을 위한 데이터 증강 및 일반화 >

```

augment_size = 10000 # 변형된 이미지 1만개 만들기
randidx = np.random.randint(X_train.shape[0], size = augment_size)
print('randidx : ', len(randidx)) # 10000
x_augment = X_train[randidx].copy()
y_augment = y_train[randidx].copy()

x_augment = img_gen.flow(x_augment,
                        np.zeros(augment_size),
                        batch_size = augment_size,
                        shuffle = False).next()[0]

X_train = np.concatenate((X_train, x_augment))
y_train = np.concatenate((y_train, y_augment))
print(X_train.shape) # (11875, 28, 28, 3)
print(y_train.shape) # (11875, 9)

# 일반화
X_train = X_train.astype(float) / 255
X_test = X_test.astype(float) / 255

```

< 모델 생성 >

```
# 모델
model = Sequential()

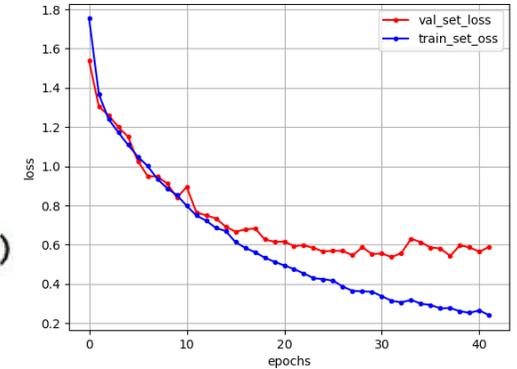
model.add(Conv2D(32, (3,3), padding="same", input_shape=X_train.shape[1:], activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3,3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model_path = "model/model_1.hdf5"
checkpoint = ModelCheckpoint(filepath=model_path , monitor='val_loss', verbose=0, save_best_only=True)
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

history = model.fit(X_train, y_train, batch_size=128, epochs=200, validation_split=0.2, \
                    callbacks=[checkpoint, early_stopping])
```



소형 폐기물 모델 :

loss: 0.3410 - accuracy: 0.8762

val_loss: 0.5506 - val_accuracy: 0.8290

대형 폐기물 모델 :

loss: 0.2100 - accuracy: 0.9214

val_loss: 0.4061 - val_accuracy: 0.8740

→ 과적합 방지

< 예측 결과 출력 >

```

for i in prediction:
    pred_lb = i.argmax() # 예측 레이블
    #print(i) # 배열
    #print(pred_lb) # 숫자
    pred_lb_name = ''
    if pred_lb == 0:
        pred_lb_name = "can"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    elif pred_lb == 1:
        pred_lb_name = "glass"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    elif pred_lb == 2:
        pred_lb_name = "paper"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    elif pred_lb == 3:
        pred_lb_name = "plastic"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    elif pred_lb == 4:
        pred_lb_name = "st"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    else:
        pred_lb_name = "vinyl"
        print("해당 "+filenames[cnt].split("\\")[1]+" 이미지는 "+pred_lb_name+"로 추정됩니다.")
    pred_lb_name_vali.append(pred_lb_name) # 예측된 라벨 저장
    cnt += 1
    
```

올바른 분리수거 방법/폐기 업체 정보와
환경 관련 뉴스 및 통계 자료 제공



" 종이 "

- 종류 : 박스, 종이팩(우유팩, 음료수팩, 1외용 종이컵), 기타 종이류(복사지, 광고전단지, 포장지, 종이 소빙 팩, 달력 등)
- 분리배출요령 :
 - 종이팩 : 내용물을 비우고 물로 한번 헹군 후 납작하게 눌러 병뚜에 넣거나 만대 묶어서 보관 배출한다.
 - (그대로 밀린 후 납작하게 눌러 묶어서 보관한다.)
 - 기타 종이류 : 물이나 오물에 젖지 않게 모아 30cm 정도로 묶어 보관 배출한다.
- 주의사항 :
 - 다른 종이류와 섞이지 않도록 별도의 분리 수거함을 마련하도록 한다.
 - 우유팩이나 음료수팩 중 장기 보존용인 테트라팩은 재생이 안되므로 제외한다.
 - 비닐 코팅지나 소빙팩의 비닐 끈, 비닐담개, 달력 스프링 등을 제거한 후 배출한다.
 - 박지, 나염지, 인쇄지(사진)는 제외한다.

필요정보기
기술정보기

"환경" 관련 키워드를 입력해보세요!
검색하기

* 대형

* 소형

```
def wordcloudFunc(request):  
    query = request.GET.get('search')  
    query = query.replace(' ', '+')  
  
    url = "https://search.naver.com/search.naver?where=news  
        &sm=tab_jum&query={}"  
    req = requests.get(url.format(query))  
    #res = req.urlopen(url)  
    soup = BeautifulSoup(req.text, "html.parser")
```

```
a_tags = soup.find_all("a", {'class':"news_tit"})
```

→ BeautifulSoup 이용
뉴스 기사 스크랩

```
sentences_tag = okt.pos(text) → 형태소 분석
```

```
for word, tag in sentences_tag:  
    if tag in ['Noun', 'Adjective']:  
        noun_adj_list.append(word) → 명사와 형용사인 단어
```

```
counts = Counter(noun_adj_list)  
tags = counts.most_common(100) → 가장 많이 나타나는  
단어 100개 저장
```

```
wc = WordCloud(font_path=r'C:/work/psou/final/app/NanumGothic-  
Regular.ttf', mask=mask, background_color="white",  
max_font_size=60)  
cloud = wc.generate_from_frequencies(dict(tags))
```

→ 워드클라우드 출력



메인으로 돌아가기

<관련 뉴스 기사를 확인해보세요!>

1. 환경부-교육부, '2021 환경공동선언식' 개최
<http://www.etnews.com/20210913000163>
2. 금강청, 추석 연휴 기간 환경오염 예방 '특별감시' 실시
http://www.newsis.com/view?id=NISX20210913_0001581306&clID=10807&plD=10800
3. 삼성바이오로직스, 기후환경리스크 관리 모형 개발 프로젝트 참여
<http://www.fnnews.com/news/202109131219562932>
4. [에코백] 현대차 수소선언, 환경단체가 비판하는 이유는?
<https://hankookilbo.com/News/Read/A2021090914240004953?did=NA>

* 검색어 : 환경

6. 한계점 및 보완점

1. 폐기물을 더욱 다양한 카테고리로 나누지 못한 점
2. 낮은 컴퓨터 성능으로 인해 더 많은 이미지를 학습시키지 못한 점
3. 서비스 이용자들의 접근성 향상을 위해 모바일 App으로 구현하고 싶었으나 시간상의 제약으로 인해 웹으로만 구현한 점

Q & A

감사합니다!