[빅데이터 프로그래밍 Project]

Article Python Project

담당교수 : 서경희 교수님

20141542 심찬양

개발환경: **Anaconda Jupyter notebook VM &**

Python 3.7.0 / Mac OS 10.14.1

공유수(shares)가 많은 기사의 특징은 무엇일까?

자료(Source) 소개

- UCI Machine Learning repository Online News Popularity Data Set

  From Kelwin Fernandes .Portugal/Universidade 외 4 인

- 이 Data Set 은 www.mashable.com 에 2015 년 1 월까지 포스트된 기사들로 부터 저자가 61 가지 속성(feature)을 .CSV 파일형태로 추출한 것입니다.

  Attribute Information:
  0. url: URL of the article (non-predictive)
  1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
  2. n_tokens_title: Number of words in the title
  3. n_tokens_content: Number of words in the content
  4. n_unique_tokens: Rate of unique words in the content
  5. n_non_stop_words: Rate of non-stop words in the content
  6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content

  .

  .

  .

  56. title_subjectivity: Title subjectivity
  57. title_sentiment_polarity: Title polarity
  58. Abs_title_subjectivity: Absolute subjectivity level
  59. abs_title_sentiment_polarity: Absolute polarity level
  60. shares: Number of shares (target)

공유수(shares)가 많은 기사의 특징은 무엇일까?

1. 전처리 과정(Preprocessing Process)

사용목적에 맞게 Data 를 다듬고 그룹을 나누는 과정입니다.

- 공유수 축소

```
for i in my_news.index:
    my_news.loc[i,' shares']/=10000
```

- Low 그룹과 High 그룹으로 나눔 (공유수 기준)

```
for i in my_news[' shares']:
    if i <= 0.1: # 1000이하는 Low
        sha_grade.append("Low")
    elif i >=1.5: # 15000이상은 High
        sha_grade.append("High")
    else:
        sha_grade.append("Mid")
my_news[' sha_grade'] = sha_grade # columns 추가
```

- 기사의 Category 와 Day feature 정리

```
for i in my_news.index:
    if int(my_news.loc[i,' data_channel_is_lifestyle']) == 1:
        channel.append("1") #life
        continue
    elif int(my_news.loc[i,' data_channel_is_entertainment']) == 1:
        channel.append("2") #enter
        continue
    elif int(my_news.loc[i,' data_channel_is_bus']) == 1:
        channel.append("3") #bus
        continue
    elif int(my_news.loc[i,' data_channel_is_socmed']) == 1:
        channel.append("4") #socmed
        continue
    elif int(my_news.loc[i,' data_channel_is_tech']) == 1:
        channel.append("5") #tech
        continue
    elif int(my_news.loc[i,' data_channel_is_world']) == 1:
        channel.append("6") #world
        continue
    else:
        channel.append("7") #etc
        continue
```

공유수(shares)가 많은 기사의 특징은 무엇일까?

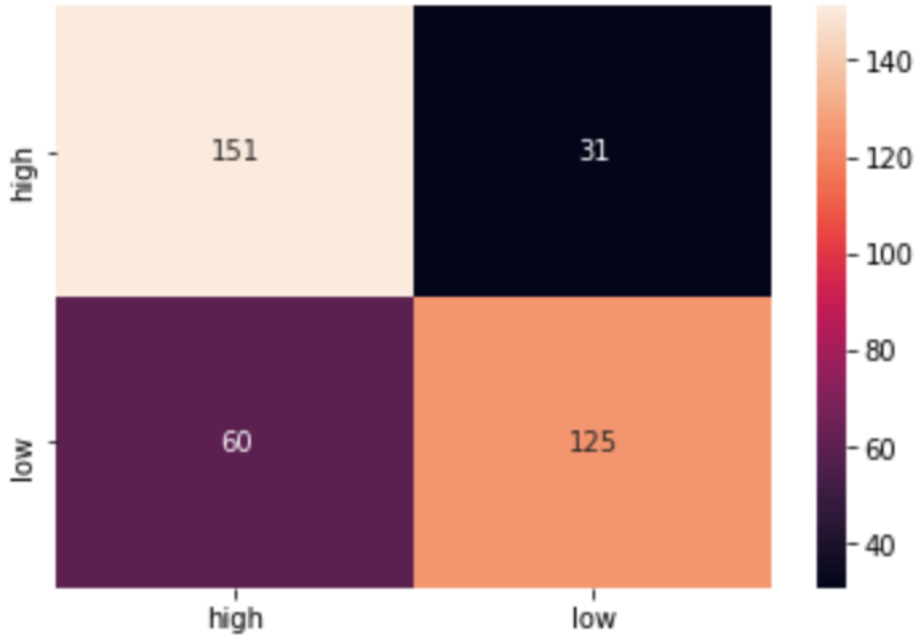
2. 기계 학습(Machine Learning)

Sklearn 모듈을 사용하여 Machine Learning 과 Deep Learning 을 이용한 결과를 얻고, 어떤 Feature 가 크게 기여했는지 알아 보겠습니다.

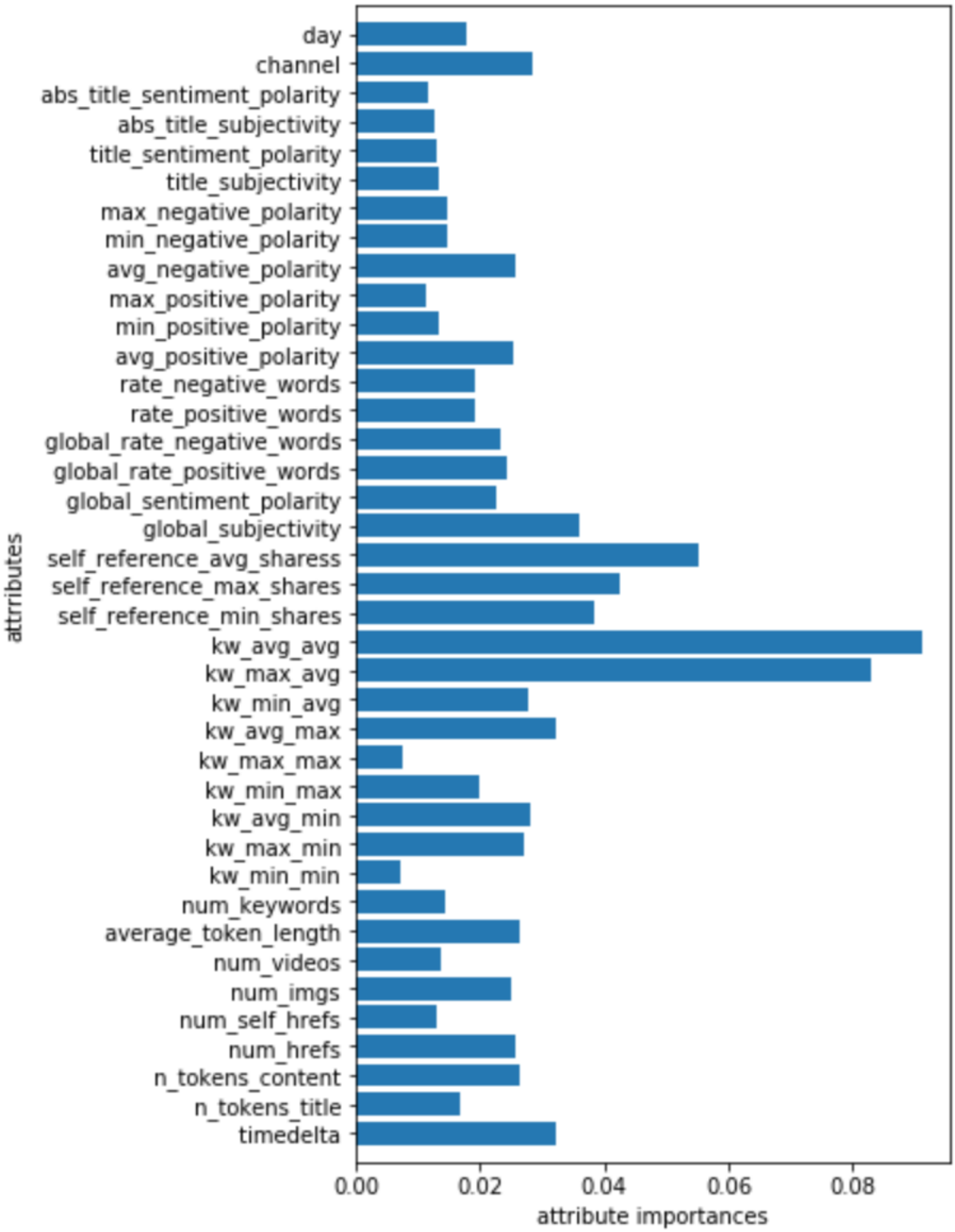
- Machine Learning

  데이터를 Training 데이터와 Test 데이터로 나누고 RandomForest 알고리즘을 이용해서 훈련된 결과를 평가합니다. (Test data = 15%)

	precision	recall	f1-score	support
High	0.72	0.83	0.77	182
Low	0.80	0.68	0.73	185
accuracy			0.75	367
macro avg	0.76	0.75	0.75	367
weighted avg	0.76	0.75	0.75	367



예측한 결과를 나타낸 Heat Map



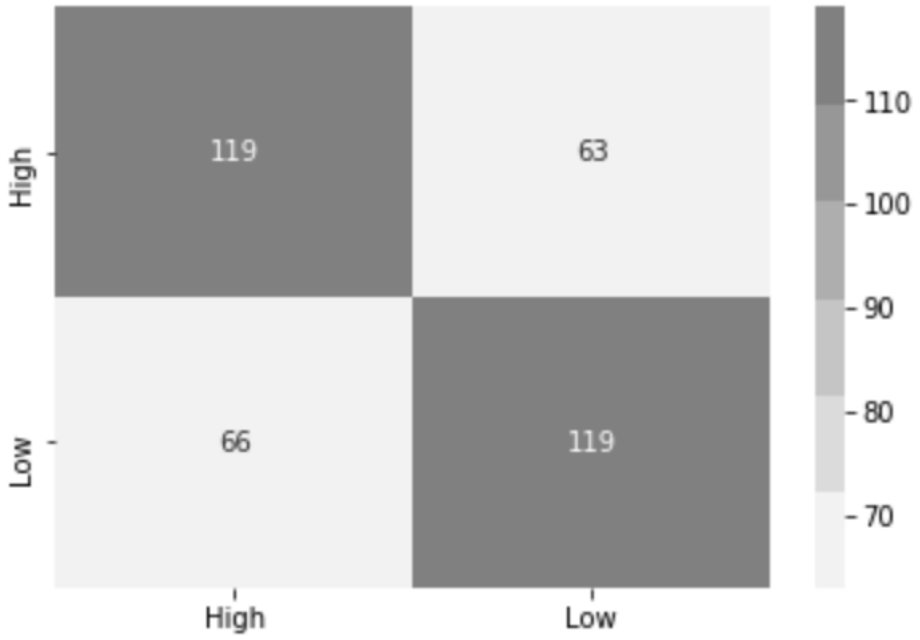
평가된 결과에서 RandomForest 가 판단한 각 feature 의 중요도

1. kw_avg_avg: Avg. keyword (avg. shares)
2. kw_max_avg: Avg. keyword (max. shares)

- Deep Learning

  데이터를 Training 데이터와 Test 데이터로 나누고 RandomForest 알고리즘을 이용해서 훈련된 결과를 평가합니다. (Test data = 15%)

	precision	recall	f1-score	support
High	0.64	0.65	0.65	182
Low	0.65	0.64	0.65	185
accuracy			0.65	367
macro avg	0.65	0.65	0.65	367
weighted avg	0.65	0.65	0.65	367

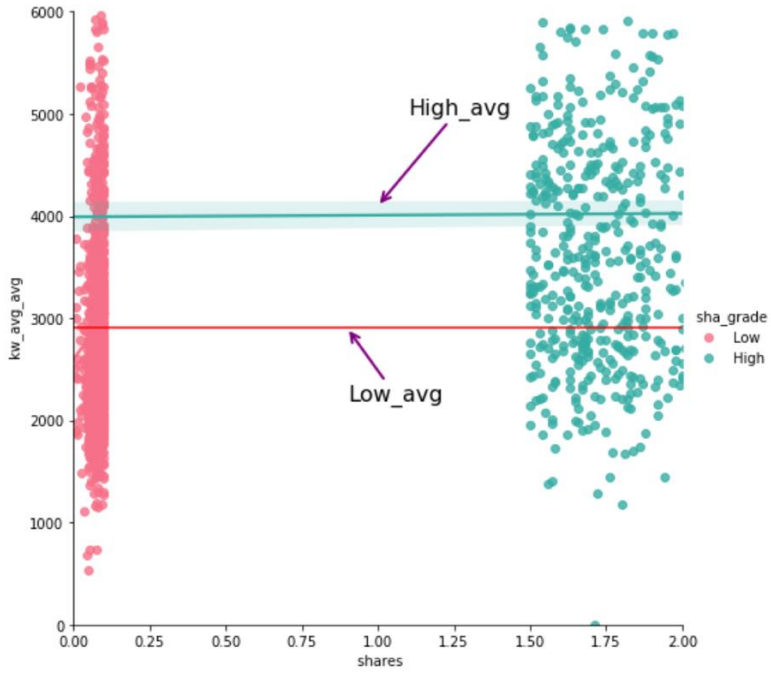


공유수(shares)가 많은 기사의 특징은 무엇일까?

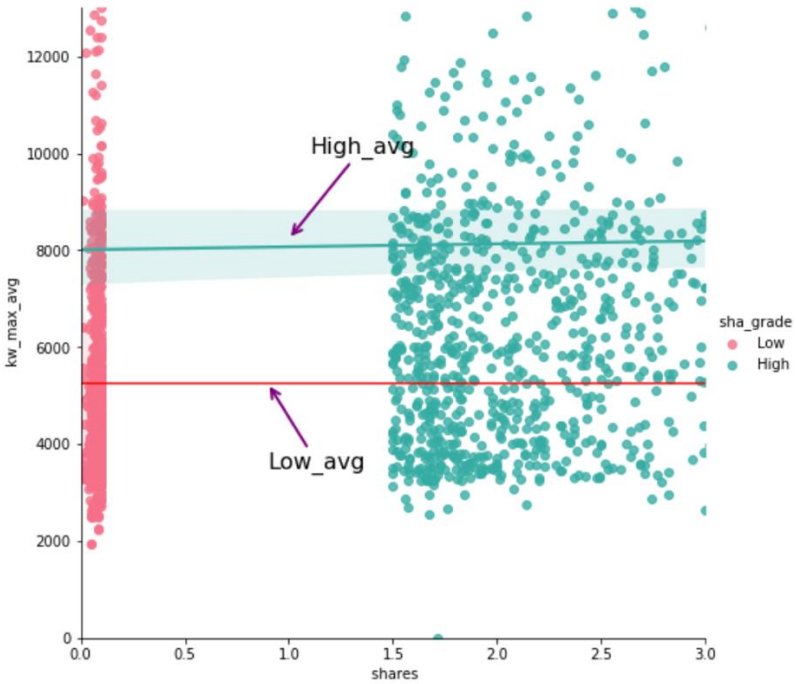
3. 시각화(Visualization)

2 에서 얻은 feature 별 중요도를 참고 하여 Matplotlib, Seaborn 모듈을 이용해 그래프를 만들었습니다.

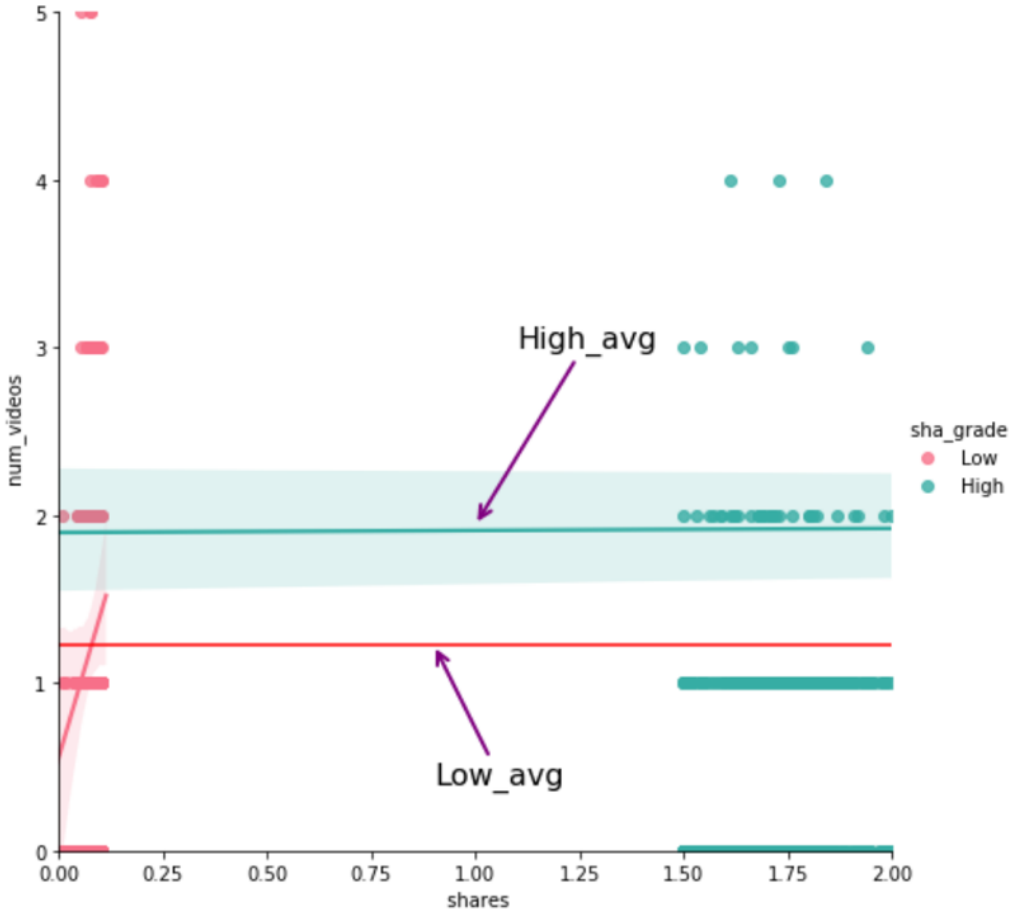
- 비교적 높은 기여도 feature



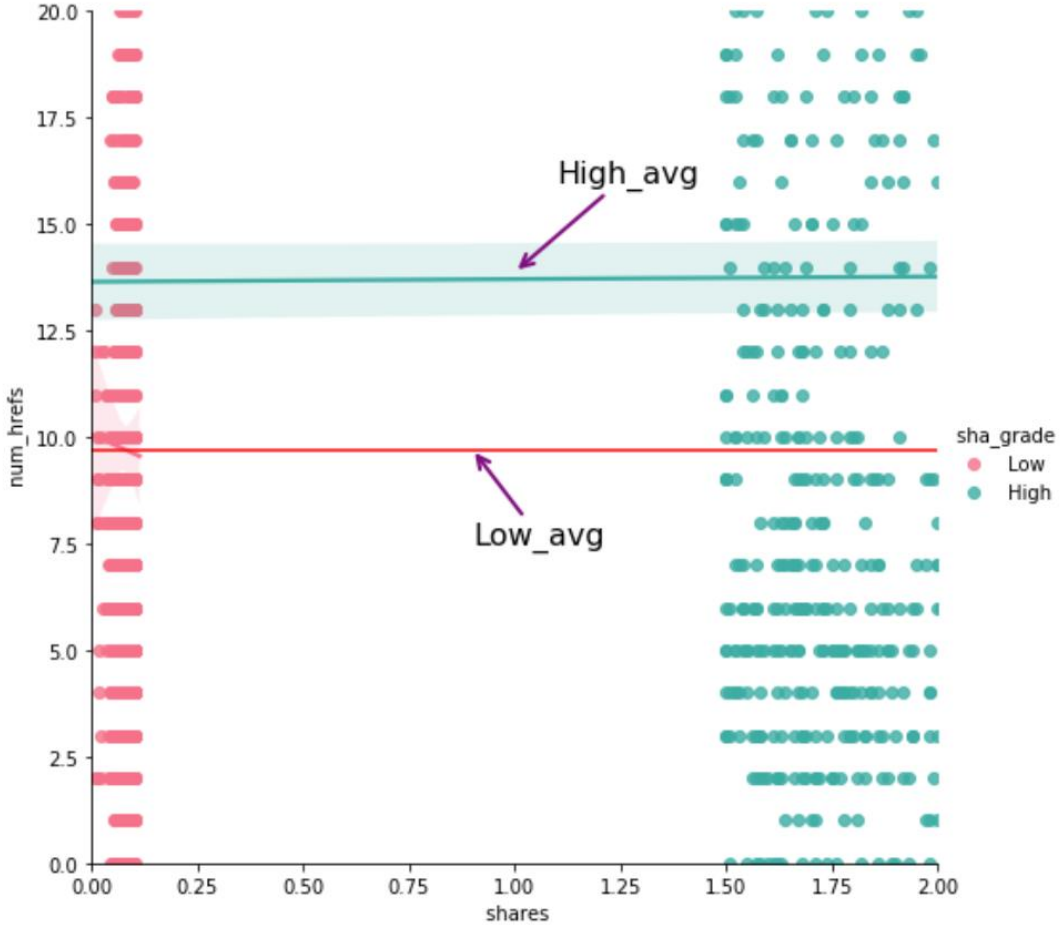
kw_avg_avg 와 kw_max_avg



- 비교적 낮은 기여도 feature



Videos 와 Hrefs



공유수(shares)가 많은 기사의 특징은 무엇일까?

4. 직접 찾아보자.. (Crawling & Regular Exp)

Requests 와 bs4 모듈을 이용해서 기사의 Title 을 crawling 하고 그룹(share)별 빈도수 높은 단어를 골라보자.

- Crawling

```
for i in craw_news.index:
    if craw_news.loc[i,' sha_grade']=="Low": # Low
        res = requests.get(my_news.loc[i,'url'])
        soup = BeautifulSoup(res.content,'html.parser')
```

Data 의 url 을 입력하여 해당 url 기사의 html 내용을 긁어옵니다.

```
title_p = re.compile(r'(?=<title>)(.*)(?<=</title>)',re.M)
word_p = re.compile(r'\b[a-zA-Z]{3,15}\b',re.M)
```

```
m = title_p.search(str(soup))
m = re.sub(r"title|[/]title","",m.group(0))
words = word_p.findall(m)
```

위 과정을 통해 <title> 과 </title> 사이 내용을 3 자에서 15 자 사이의 단어들로 저장합니다.

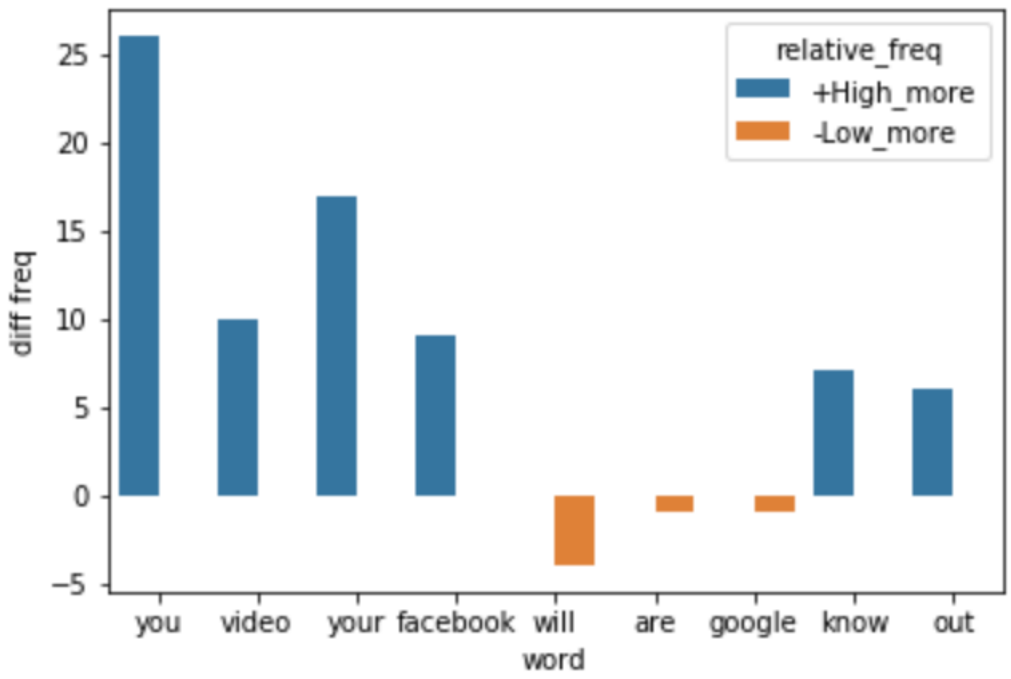
```
for word in words:
    word = word.lower()
    if (word =="the" or word =="for" or
        word =="and" or word =="with" or
        word == "from" or word =="that"or
        word == "this"):
        continue;
    if word in low_dic:
        low_dic[word] += 1
    else:
        low_dic[word] = 1
```

- 시각화

Low shares	
video	35
you	27
your	24
will	18
twitter	16
new	16
apple	15
more	15
are	13
google	13
now	11
how	11
best	10
facebook	9
live	9
game	9
time	8
its	8
what	8
star	8

High shares	
you	53
video	45
your	41
new	20
facebook	18
how	16
twitter	16
will	14
are	12
google	12
like	12
iphone	12
time	11
app	11
what	11
know	11
out	10
game	10
about	10
first	10

공유수가 낮은 그룹과 높은 그룹에서 나온 title 의 단어들의 같은 단어의 빈도수 차를 y 값으로 가지는 그래프를 만들었습니다.



결론: 1. AI 로도 공유수가 높은 기사의 특징은 찾기가 쉽지 않다.

2. 대신 you, your 라는 단어로 기사제목을 쓴다면 공유수가 높을 지도 모르겠습니다ㅎㅎ..