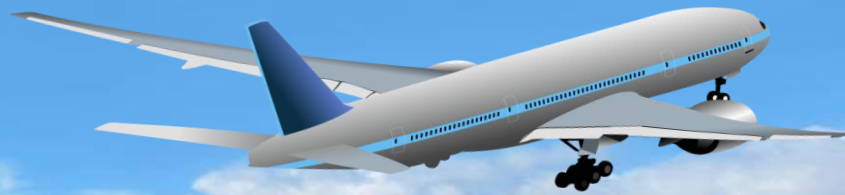
빅데이터 마케팅
Crawling Team Project

정재학 교수님

20160796 박보성
20150904 이예인
20141582 정필립

목차

1. 주제선정
2. 1차 크롤링
3. 데이터 수집 목표 수립
4. 항공사별 2차 크롤링
5. 매체별 결과 비교분석
6. 데이터 정제
7. 결과 분석
8. 시사점



1. 주제 선정

- 주제선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 특성
- 데이터 정제
- 결과 분석
- 시사점 도출

FSC 항공사

KOREAN AIR

ASIANA AIRLINES

LCC 항공사

AIR SEOUL

AIR BUSAN

JEJUair

t'way

JINAIR

EASTAR JET

LCC항공사 브랜드 증가(2019년 4월 4개 항공사 추가 승인)
FSC와 차별화된 LCC항공사의 소비자 인지도 조사 필요

2. 방향 설정

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 특성
- 데이터 정제
- 결과 분석
- 시사점 도출

포괄적 키워드[항공사]로 1차적 텍스트 크롤링 하기

►항공사에 대한 전반적 소비자 인식 조사

Twitter: 1887개
Naver블로그 : 1774개
Youtube: 22045개 댓글

매체 선정

▪트위터
-솔직한 리뷰/의견

▪네이버 블로그
-항공사 상세 리뷰
-정보전달 목적 글

▪Youtube
-항공사 관련 리뷰 다양
-댓글의 상호작용 활발

Main.py 메인 구현 코드

```
import sys
import naver_crawler
import youtube_crawler
import twitter_crawler

MAX_MENU = 3
NAVER_CRAWLING = 1
YOUTUBE_CRAWLING = 2
TWITTER_CRAWLING = 3

def menu():
    print('Crawling Program >> Please select menu')
    print('=======================================')
    print('1. Naver Blog Crawling')
    print('2. Youtube Comment Crawling')
    print('3. Twitter Crawling')
    print('=======================================')

if __name__ == "__main__":

    menu()
    # 메뉴 버튼 잘못 눌렀을 경우의 예외처
    try:
        menu_num = int(input('Menu : '))

    except :
        print('Please press valid menu number')
        sys.exit(1)
    if(not( 1 <= menu_num <= MAX_MENU)):
        print('Please press valid menu number (1~3)')
        sys.exit(1)

    # 네이버 블로그 크롤링
    if(menu_num == NAVER_CRAWLING):
        naver_crawler.crawling()

    # 유튜브 댓글 크롤링
    elif(menu_num == YOUTUBE_CRAWLING):
        #youtube_crawler.video_url_crawling()          # 유튜브에 있는 동영상 제목, url 전부 가져옴
        youtube_crawler.video_comment_crawling()     # 유튜브 각 동영상에 있는 댓글 전부 가져옴

    # 트위터 게시글 크롤
    elif(menu_num == TWITTER_CRAWLING):
        twitter_crawler.crawling()
```

```python
import sys
import urllib
import time
import pandas as pd
import re
from pandas import DataFrame, Series
from urllib.request import urlopen
from bs4 import BeautifulSoup
from selenium import webdriver as wd
from selenium.webdriver.common.keys import Keys


TWITTER_URL = 'https://twitter.com/search?l=&q='


def crawling():
    data = []
    keyword = input('Keyword : ')

    driver = wd.Chrome('./tool/chromedriver.exe')
    driver.maximize_window()

    driver.get(TWITTER_URL + keyword)

    print('The scroll is starting to move bottom')

    # 페이지 스크롤을 끝날 때까지 계속 내림
    # 스크롤을 내리기 전의 화면 높이와 내렸을 때의 화면 높이가 같다면 더 이상 내려갈 곳이 없다는 의미이므로 무한 루프를 탈출함.
    last_height = driver.execute_script("return document.body.scrollHeight")
    while True:
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight)")
        time.sleep(1)
        new_height = driver.execute_script("return document.body.scrollHeight")

        if new_height == last_height:
            # Wait to load page
            time.sleep(1)
            new_height = driver.execute_script("return document.body.scrollHeight")

            if(new_height == last_height):
                break

        last_height = new_height
    print('Arrived at the end of the page')
    print('Start twitter crawling')
    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

동적데이터:
스크롤을 끝까지 내리는 것을 코드로 구현
▶모든 데이터 긁어올 수 있게 함


```
# 트위터의 게시글에는 각각의 고유의 아이디가 있어서 트위터 사이트에서 모든 게시글 아이디를 리스트형으로 가져옴
pattern = re.compile('stream-item-tweet-\d+')
items = pattern.findall(str(soup))
for item in items:
    # 위에서 가져온 고유 아이디를 이용하여 게시글 본문을 css selector를 이용하여 가져
    text = driver.find_element_by_css_selector('#'+ item +' > div > div.content > div.js-tweet-text-container > p').text

    # 특수기호를 없애는 작
    for idx in range(len(text)):
        if not ((0 <= ord(text[idx]) < 128) or (0xac00 <= ord(text[idx]) <= 0xd7af)):
            text = text.replace(text[idx], ' ')

    data.append(text)

driver.close()

print('Finish crawling')
print('The data is being written to the csv file.')
dataframe = pd.DataFrame(data, columns=["content"])
dataframe.to_csv('../data/twitter_comment.csv', mode = 'a', encoding='cp949')
print('Finish working')
```

NAVER

```
import sys
import urllib
import time
import pandas as pd
import re
from pandas import DataFrame, Series
from urllib.request import urlopen
from bs4 import BeautifulSoup
from selenium import webdriver as wd
from selenium.webdriver.common.keys import Keys


NAVER_URL = 'https://section.blog.naver.com/Search/Post.nhn?pageNo=%%PAGE_NUM%%&rangeType=ALL&orderBy=sim&keyword='


def crawling():
    data = []
    BLOG_COUNT_PER_PAGE = 7
    keyword = input('Keyword : ')
    try:
        page_num = int(input('Page Num : '))
    except:
        print('Please press valid page number (int)')
        sys.exit(1)

    driver = wd.Chrome('./tool/chromedriver.exe')
    driver.maximize_window()

    # 블로그 제목에 해당하는 css selector -> 중간에 BLOG_NUM에 따라 계속 바뀜
    _const_title_selector = '#content > section > div.area_list_search > div:nth-child(%%BLOG_NUM%%) > div > div.info_post > div > a.desc_inner > strong > span.title'

    # 블로그 미리보기 내용에 해당하는 css selector -> 중간에 BLOG_NUM에 따라 계속 바뀜
    _const_content_selector = '#content > section > div.area_list_search > div:nth-child(%%BLOG_NUM%%) > div > div.info_post > div > a.text'

    temporary_storage_num = 1
    for PAGE_NUM in range(page_num + 1):
        print(f'Start crawling page {PAGE_NUM}')

        # page 번호에 해당하는 url을 계산
        link = NAVER_URL.replace('%%PAGE_NUM%%', str(PAGE_NUM)) + keyword
        driver.get(link)
```

NAVER

```
    # page 로딩 시간 기다림
    time.sleep(2)
    for BLOG_NUM in range(1, BLOG_COUNT_PER_PAGE + 1):
        title_selector = _const_title_selector.replace('%%BLOG_NUM%%',str(BLOG_NUM))
        content_selector = _const_content_selector.replace('%%BLOG_NUM%%', str(BLOG_NUM))

        # 제목과 내용을 가져옴
        title  = driver.find_element_by_css_selector(title_selector).text
        content = driver.find_element_by_css_selector(content_selector).text

        # 특수기호 없애는 작업
        for idx in range(len(title)):
            if not ((0 <= ord(title[idx]) < 128) or (0xac00 <= ord(title[idx]) <= 0xd7af)):
                title = title.replace(title[idx], ' ')

        for idx in range(len(content)):
            if not ((0 <= ord(content[idx]) < 128) or (0xac00 <= ord(content[idx]) <= 0xd7af)):
                content = content.replace(content[idx], ' ')

        data.append([title, content])

    # 컴퓨터 메모리 때문에 100 page씩 데이터를 저장하고 버퍼를 지운다.
    if temporary_storage_num % 100 == 0:
        dataframe = pd.DataFrame(data, columns=["title", "content"])
        dataframe.to_csv('./naver_comment.csv', mode='a', encoding='cp949')
        data = []

    temporary_storage_num += 1

driver.close()

print('Finish crawling')
print('The data is being written to the csv file.')
dataframe = pd.DataFrame(data, columns=["title", "url"])
dataframe.to_csv('../data/naver_comment.csv', mode='a', encoding='cp949')
print('Finish working')
```

```python
import sys
import urllib
import time
import pandas as pd
import re
from pandas import DataFrame, Series
from urllib.request import urlopen
from bs4 import BeautifulSoup
from selenium import webdriver as wd
from selenium.webdriver.common.keys import Keys


YOUTUBE_URL = 'https://www.youtube.com/results?search_query='


def video_url_crawling():

        data = []
        keyword = input('Keyword : ')

        driver = wd.Chrome('./tool/chromedriver.exe')
        driver.maximize_window()

        driver.get(YOUTUBE_URL+keyword)


        print('The scrolling starts moving to the bottom of the main page.')
        body = driver.find_element_by_tag_name("body")
```

함수1. video_url_crawling
: url 제목 가져오기

```
while True:

    ### id = '#message' 에 해당하는 값을 가져옴.
    ### 유튜브에서는 id = '#message'는 scoll을 맨 끝까지 내렸을 때 나타나는 값임.
    items=driver.find_elements_by_css_selector('#message')

    ### scroll을 끝까지 내렸다면 반복문 탈출
    if(len(items)):
        break
    body.send_keys(Keys.PAGE_DOWN)

    ### 로딩 시간 기다림
    time.sleep(0.1)

print('Arrived at the end of the main page')
print('Start to get the url of the video')

### 동영상 제목, url을 가지고 있는 class를 가져옴.
items = driver.find_elements_by_css_selector('#video-title')

for idx in items:
    if(idx.get_attribute('href') is not None):
        ### 한글 깨짐 방지
        text = idx.text

        for i in range(len(text)):
            if not ((0 <= ord(text[i]) < 128) or (0xac00 <= ord(text[i]) <= 0xd7af)):
                text = text.replace(text[i], ' ')
        data.append([text, idx.get_attribute('href')])

driver.close()

print('Finish previous working')
print('The data is being written to the csv file.')

### csv 파일에 저장 [동영상 제목, 동영상 url]
dataframe = pd.DataFrame(data, columns=["title", "url"])
dataframe.to_csv('../data/youtube_url_collection.csv', encoding='cp949')

print('Finish working')
```

```python
def video_comment_crawling():
    data = []
    df = pd.read_csv('../data/youtube_url_collection.csv', encoding = 'cp949')

    driver = wd.Chrome('./tool/chromedriver.exe')
    driver.maximize_window()

    temporary_storage_num = 1
    for i in range(len(df.index)):
        title = df['title'][i]
        link = df['url'][i]

        print(f'Start comment crawling : title = {title}')

        driver.get(link)
        time.sleep(2)

        count = 0
        body = driver.find_element_by_tag_name("body")

        print('The scrolling starts moving to the bottom of the comment page.')

        ### 댓글 데이터를 가져옴
        last = driver.find_elements_by_css_selector('#content-text')

        while True:
            body.send_keys(Keys.PAGE_DOWN)
            time.sleep(0.4)
            new = driver.find_elements_by_css_selector('#content-text')

            if new == last:
                if count == 10:
                    break
                count += 1

            else:
                count = 0

            last = new

        print('Arrived at the end of the comment page')
```

함수2. video_comment_crawling
:영상 안의 댓글 긁어오기

```
    for idx in new:
        ### 한글 깨짐 방지
        text = idx.text

        for idx in range(len(text)):
            if not ((0 <= ord(text[idx]) < 128) or (0xac00 <= ord(text[idx]) <= 0xd7af)):
                text = text.replace(text[idx], ' ')

        data.append([title, text])

    if temporary_storage_num % 100 == 0:
        dataframe = pd.DataFrame(data, columns=["title", "content"])
        dataframe.to_csv('../data/youtube_comment.csv', mode = 'a', encoding='cp949')
        data = []

    temporary_storage_num += 1


driver.close()
print('Finish comment crawling')
print('The data is being written to the csv file.')

### 댓글 정보를 csv 파일에 저장
dataframe = pd.DataFrame(data, columns=["title", "content"])
dataframe.to_csv('../data/youtube_comment.csv', mode = 'a', encoding='cp949')

print('Finish working')
```

3. Data 수집 목표 수립

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 특성
- 데이터 정제
- 결과 분석
- 시사점 도출

2차 크롤링

키워드: ★

KOREAN AIR
ASIANA AIRLINES
→ FSC항공사:
소비자 인식 차별점

JEJUair
→ LCC항공사
매출 1위 항공사

데이터 정제

twitter NAVER YouTube

매체별 특성 파악

매체별 노이즈 분석

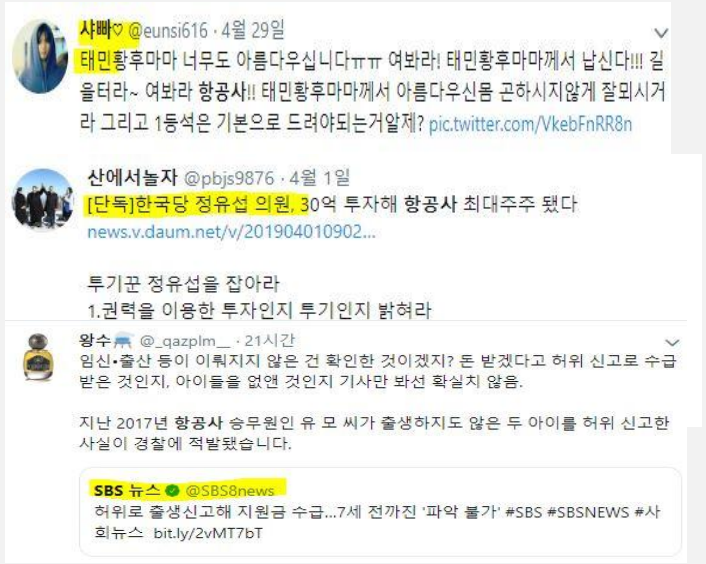
현재 타겟 텍스트에 적합한
SPAM dictionary 제작

텍스트(data)에서 유의미한 정보(information) 얻기

4. 매체별 비교 (1)

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- **매체별 비교**
- 데이터 정제
- 결과 분석
- 시사점 도출

1. TWITTER



►경험 중심 리뷰 (정보전달적)

►정치적 의견 드러내는 글 정제 필요
(비속어, 관련 없는 욕설)

►아이돌 관련 글 정제 필요
(관련 없는 '덕질' 게시물들)

►Re-Tweet을 이용한 광고성 글 정제 필요

4. 매체별 비교 (2)

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

2. 네이버 블로그

►가장 경험 중심 리뷰 (정보전달적)

►경험 이외의 핵심 이슈들 잘 전달 (기사리뷰 etc)

► 승무원 입사 관련 글 정제 필요

4. 매체별 비교 (3)

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- **매체별 비교**
- 데이터 정제
- 결과 분석
- 시사점 도출

3. YOUTUBE

돈 얘기 좀 시작해 볼까?[항공사 때려친 언니들]

달콤한크루들 • 조회수 108만회 • 5개월 전

안녕하세요. 항공사 때려친 언니들입니다^.^ 승무원만 누릴 수 있는 꿀 혜택부터 항공사별 월급 비교까지 ~! 이거 알면 승무원 하고 싶을 ...

베트남항공 A350-900 비즈니스석 리뷰 / Vietnam Airlines A350-900 Business Class Review

PrestigeGorilla • 조회수 7만회 • 5개월 전

"김사원 사과해!!! 감히 우릴 무시하다니💥" #베트남항공 A321 타고 다낭 출장을 떠났던 지난 날😩 웃펐던(?) 김사원의 첫 비즈니스석 ...

'보잉 737 맥스' 또 추락..한국 항공사들도 대량 구입 / 한국경제TV

한국경제TV ✓ 조회수 1.7천회 • 1개월 전

미국 항공 대기업 보잉의 중소형 여객기인 737 맥스 8이 추락하는 참사가 또 일어났습니다. 중장거리에 특화된 737 기종을 개량한 ...

►과시적, 자극적 영상 지배적
(구독수를 늘리기 위한 고가 항공권 이용 후기)

►승무원 입시 관련 영상 정제 필요

►항공사 브랜드 이미지와
관련성이 다소 낮은 뉴스 기사들 혼재

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

5. 데이터 정제

타겟 텍스트에 적합한 SPAM dictionary 제작 필요

각 SNS 별 특징 잘 반영하는
SPAM list dictionary 형성

원본 Crawled text file에서
SPAM dictionary에 해당하지 않는 데이터만
따로 추출 ► 파일 형성

코드 제작

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

5. 데이터 정제

Dictionary 제작하기 _ 총 103개

광고성메세지 관련 RT rt Rt 링크	**정치관련** 청와대 문재인 북한 정부 최순실 뉴스 국회의원 공무원 새누리당 사장 장관 부사장	**유튜버/댓글 관련** 홀수 짝수 릴카 릴 존예 릴하 미미
아이돌 동방신기 BTS RM		
욕설 및 기타 샹* 인성 김치 새* 시* 예약기간 작성일		**승무원 취업** 토익 정보 감사 스펙 채용

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

5. 데이터 정제

Dictionary 제작하기 _ 총 103개

승무원취업
합격
지상직
인터뷰
공지
발표
서류
학원
기출
상담
과외
취업
면접
진로
정책
인턴

면허
사업
취준생
컨설팅
입사
대졸
교육
특강
잘생긴
계약
자기소개서
일자리
지원

유튜브 관련 키워드
구독자
음악
구독
편집
영상
발음
편집자
목소리
브금
댓글

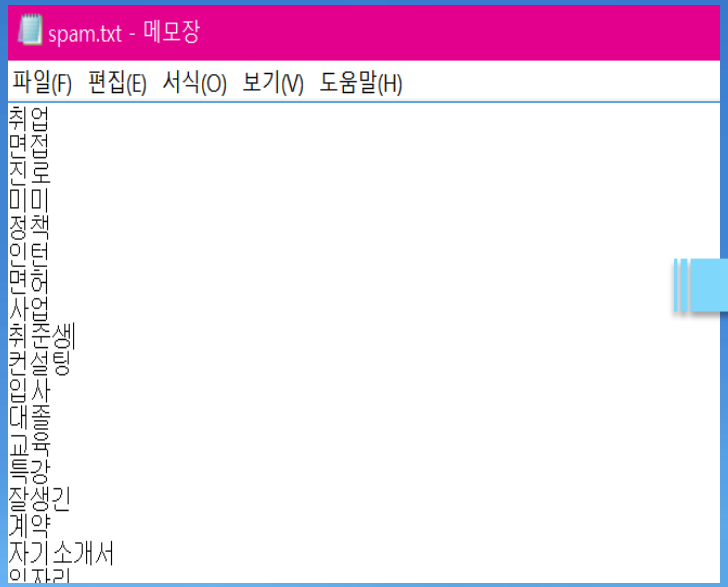
승무원 관련
힘들
연봉
월급
급여
기업
서비스직

승무원
박봉
외모
취집
땅콩
시집
언니
누나
누님
유니폼
연예인
메이저/메이져
승부심
결혼
못생
스튜어디
응원
이뻐 /예쁘/예쁜

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- **데이터 정제**
- 결과 분석
- 시사점 도출

5. 데이터 정제

데이터 정제 코딩



SPAM dic 에 있는 단어들을
SPAM.txt 파일에 모두 저장

필요에 따라 3가지 필터 코드 제작
►spam.txt에 저장되어 있는 단어가 포함되어 있지 않은 데이터만 추출
►Spam 과 별개로 특정 데이터만 포함 되어 있는 데이터 추출 (and)
►특정 데이터만 포함되어 있는 데이터 추출(or)

1. spam.txt에 저장되어 있는 단어가 포함되어 있지 않은 데이터만 추출

```
import csv
import pandas as pd

def manual():
    print('********The order of execution of this program*******')
    print('1. Read data')
    print('2. Read spam keyword that needs to be removed')
    print('3. Read the data one line at a time and remove it if it contains spam keyword.')
    print('**************************************************')


if __name__ == '__main__':

    manual()
    print('Read data')

    # 정제할 단어 목록을 읽음. 단어를 추가하고 싶으면 spam.txt파일 맨 아래에 추가
    filter_f  = open('./except_spam/spam.txt','r', encoding = 'cp949')
    naver_f   = pd.read_csv('./naver_comment.csv', encoding = 'cp949')
    twitter_f = pd.read_csv('./twitter_comment.csv', encoding = 'cp949')
    youtube_f = pd.read_csv('./youtube_comment.csv', encoding = 'cp949')

    print('Finish read data')

    # 정제할 단어 목록을 리스트형으로 변환함.
    filtering_data = filter_f.read().split('₩n')

    print('Start filtering NAVER BLOG')
    data = []
    flag = 1

    # 네이버 블로그 게시글 각각에 대하여 반복문 실행
    for content in naver_f['content'].dropna():
        flag = 1
        for keyword in filtering_data:

            # 어떤 spam keyword 중 하나라도 게시글에 포함되어 있으면 반복문 탈출
            if keyword in content:
                flag = 0
                break
        # 어떤 spam keyword라도 게시글에 포함되지 않았다면 해당 게시글을 따로 저장
        if flag:
            data.append(content)
```

NAVER

1. spam.txt에 저장되어 있는 단어가 포함되어 있지 않은 데이터만 추출

```
print('Finish filtering NAVER BLOG')
print('Start filtering TWITTER POSTS')

# 트위터 게시글 각각에 대하여 반복문 실행
for content in twitter_f['content'].dropna():
    flag = 1

    # 어떤 spam keyword 중 하나라도 게시글에 포함되어 있으면 반복문 탈출
    for keyword in filtering_data:
        if keyword in content:
            flag = 0
            break
    # 어떤 spam keyword라도 게시글에 포함되지 않았다면 해당 게시글을 따로 저장
    if flag:
        data.append(content)

# 네이버 블로그와 트위터를 하나로 합쳐서 한 파일에 저장함
dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./except_spam/naver_twitter_except_spam.csv', mode= 'w', encoding = 'cp949')
print('Finish filtering TWITTER POSTS')
print('Start filtering YOUTUBE COMMENTS')
youtube_data = []
for content in youtube_f['content'].dropna():
    flag = 1
    # 유튜브 댓글 각각에 대하여 반복문 실행
    for keyword in filtering_data:
        # 어떤 spam keyword 중 하나라도 게시글에 포함되어 있으면 반복문 탈출
        if keyword in content:
            flag = 0
            break

    # 어떤 spam keyword라도 게시글에 포함되지 않았다면 해당 게시글을 따로 저장
    if flag:
        youtube_data.append(content)
        data.append(content)
dataframe = pd.DataFrame(youtube_data, columns=['content'])
dataframe.to_csv('./except_spam/youtube_except_spam.csv', mode= 'w', encoding = 'cp949')
dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./except_spam/total_except_spam.csv', mode= 'w', encoding = 'cp949')

print('Finish filtering YOUTUBE COMMENTS')
filter_f.close()
```

twitter

- 트위터, 네이버:
  유사한 성격 ▶ filtered data 합쳐서 저장
- Youtube:
  성격 다름 ▶ 따로 저장

YouTube

2. Spam 과 별개로 특정 데이터만 포함 되어 있는 데이터 추출 (and)

```
import csv
import pandas as pd

def manual():
    print('****************************************')
    print('1. Input keyword until you enter 0')
    print('2. This program preform AND operation on the keywords')
    print('3. Create three csv files. (naver&&twitter, youtube, total)')
    print('****************************************')

if __name__ == '__main__':

    manual()

    flag = 1
    keyword = []

    # 사용자가 0을 입력하기 전까지 keyword를 계속 입력받는다.
    while True:
        tmp = input('keyword : ')
        if tmp == '0':
            break
        keyword.append(tmp)

    print('Read data')

    naver_twitter_f = pd.read_csv('./except_spam/naver_twitter_except_spam.csv', encoding = 'cp949')
    youtube_f = pd.read_csv('./except_spam/youtube_except_spam.csv', encoding = 'cp949')

    print('Finish read data')
```

2. Spam 과 별개로 특정 데이터만 포함 되어 있는 데이터 추출 (and)

```
print('Start filtering NAVER BLOG && TWITTER POSTS')
data = []

for content in naver_twitter_f['content']:
    flag = 1
    for i in keyword:
        # 네이버 블로그&&트위터 게시글에 keyword가 하나라도 포함되지 않는다면 flag를 set하고 반복문 탈출
        if i not in content:
            flag = 0
            break
    # 게시글에 모든 keyword가 있다면 따로 저장
    if flag:
        data.append(content)

dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./include_keyword/naver_twitter_include_and_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')


print('Finish filtering NAVER BLOG && TWITTER POSTS')
print('Start filtering YOUTUBE COMMENTS')

youtube_data = []
for content in youtube_f['content']:
    flag = 1
    for i in keyword:
        # 유튜브 댓글에 keyword가 하나라도 포함되지 않는다면 flag를 set하고 반복문 탈출
        if i not in content:
            flag = 0
            break
    # 댓글에 모든 keyword가 있다면 따로 저
    if flag:
        youtube_data.append(content)
        data.append(content)

dataframe = pd.DataFrame(youtube_data, columns=['content'])
dataframe.to_csv('./include_keyword/youtube_include_and_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')
dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./include_keyword/total_include_and_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')

print('Finish filtering YOUTUBE COMMENTS')
```

3. Spam 과 별개로 특정 데이터만 포함 되어 있는 데이터 추출 (or)

```
mport csv
mport pandas as pd

ef manual():
    print('****************************************')
    print('1. Input keyword until you enter 0')
    print('2. This program preform OR operation on the keywords')
    print('3. Create three csv files. (naver&&twitter, youtube, total)')
    print('****************************************')

f __name__ == '__main__':

    manual()

    keyword = []

    # 사용자가 0을 입력하기 전까지 keyword를 계속 입력받는다.
    while True:
        tmp = input('keyword : ')
        if tmp == '0':
            break
        keyword.append(tmp)

    print('Read data')

    naver_twitter_f = pd.read_csv('./except_spam/naver_twitter_except_spam.csv', encoding = 'cp949')
    youtube_f = pd.read_csv('./except_spam/youtube_except_spam.csv', encoding = 'cp949')


    print('Finish read data')


    print('Start filtering NAVER BLOG && TWITTER POSTS')
    data = []
```

3. Spam 과 별개로 특정 데이터만 포함 되어 있는 데이터 추출 (or)

```
for content in naver_twitter_f['content']:

    for i in keyword:
        # 네이버 블로그&&트위터 게시글에 keyword가 하나라도 포함되면 해당 게시글을 따로 저장하고 반복문 탈출
        if i in content:
            data.append(content)
            break
dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./include_keyword/naver_twitter_include_or_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')


print('Finish filtering NAVER BLOG && TWITTER POSTS')
print('Start filtering YOUTUBE COMMENTS')

youtube_data = []

for content in youtube_f['content']:

    for i in keyword:
        if i in content:
            # 유튜브 댓글에 keyword가 하나라도 포함되면 해당 게시글을 따로 저장하고 반복문 탈출
            youtube_data.append(content)
            data.append(content)
            break
dataframe = pd.DataFrame(youtube_data, columns=['content'])
dataframe.to_csv('./include_keyword/youtube_include_or_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')
dataframe = pd.DataFrame(data, columns=['content'])
dataframe.to_csv('./include_keyword/total_include_or_' + keyword[0] + '.csv', mode= 'w', encoding = 'cp949')

print('Finish filtering YOUTUBE COMMENTS')
```

5. 데이터 정제 Test (1차 크롤링)

데이터 정제 전후 비교

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- **데이터 정제**
- 결과 분석
- 시사점 도출

NAVER

필터링 전

필터링 후

5. 데이터 정제 Test (1차 크롤링)

데이터 정제 전후 비교

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- 데이터 정제
- 결과 분석
- 시사점 도출

twitter

필터링 전

필터링 후

5. 데이터 정제 Test (1차 크롤링)

데이터 정제 전후 비교

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- **데이터 정제**
- 결과 분석
- 시사점 도출

필터링 전

필터링 후

6. 결과 분석

주제 선정
1차 크롤링
Data 수집 목표 수립
2차 크롤링
매체별 비교
데이터 정제
결과 분석
시사점 도출

FSC-LCC 소비자 관심도
(빈도수가 높을수록 관심도가 높다고 볼 수 있다)

소비자가 고려하는 속성들 분류하기

서비스
- 전반적 친절도
- 기내식 품질
- 기본 제공 서비스 (어메니티, 간식 등)

마일리지
- 마일리지 제휴사
- 좌석 승급 혜택
- 라운지 제공 여부
- 연동 카드사 유무

가격
- 항공권의 액면가
- 항공권 할인행사

노선
- 노선의 다양성

"네 가지 속성 모두 상관관계가 높을 수 있음"

대한항공	아시아나항공	저가항공사(LCC)	제주항공
마일리지	마일리지	노선	이스타항공
기내식	스타얼라이언스	많은	티웨이항공
KE	많은	제주항공	아시아나항공
스카이팀	진에어	진에어	진에어
터미널	인천공항	가장	에어부산
에어프랑스	스카이팀	일본	취항
프레스티지석(모닝캄, 비즈니스, 퍼스트)	터미널	아시아나	에어서울
라운지	티웨이항공	저렴한	가장
델타항공	에어부산	자유여행	인천공항
KLM네덜란드항공	다른	아시아	다양한
진에어	이스타항공	에어아시아	보잉
여객터미널	에어서울	세계	맥스
제주항공	기내	대형	마일리지
터미널	오늘은	에어서울	여행
LCC	외국	carrier	매출액
중국	노선	에어부산	대표
가장	현재	티웨이항공	최고의
현재	인천공항	대표	터미널
아시아나항공	수하물	이스타항공	과징금
프레스티지석	미국	국적	영업이익
저가	소속	기내	많은
티웨이항공	대형	서비스	일본
기내	해외	가격	LCC항공사
대형	일본	미국의	에어아시아
스타얼라이언스	에어프랑스	수화물	서비스
여행	국적기	마일리지	캐리어

7. 시사점 도출

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- 데이터 정제
- 결과 분석
- **시사점 도출**

FSC 항공사

서비스 품질

마일리지 혜택

대한항공, 아시아나와 같은 FSC항공사에게 소비자는 보다 더 높은 수준의 '서비스'와 '마일리지 혜택'을 기대하며, 관심도 높다

따라서, FSC 항공사는 서비스와 멤버십을 강조한 마케팅을 통해서 Retention rate을 높이는 전략을 취해야 한다.

LCC 항공사

노선의 다양성

항공권 액면가

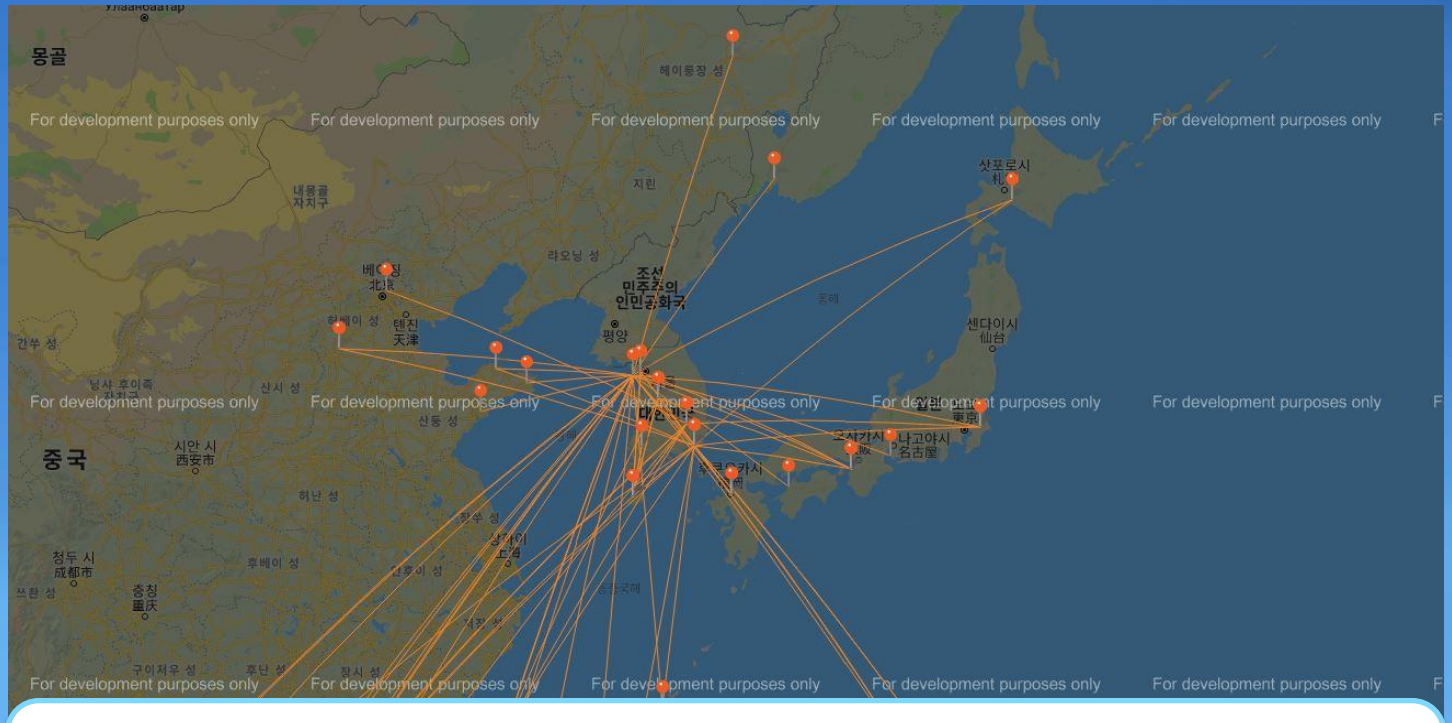
제주항공은 LCC항공사 중 매출 1위이다.
LCC항공사 관련해서는 '노선'과 '항공권 가격'에 소비자 관심도가 높다.

따라서, 제주항공을 비롯한 LCC항공사들은 FSC의 고객 층성도를 극복하기 위해, 노선과 가성비를 강조한 전략을 사용해야 한다.

FSC-LCC 브랜드 이미지 구축 차별화 필요

7. 시사점 도출 (제주항공 노선표)

- 주제 선정
- 1차 크롤링
- Data 수집 목표 수립
- 2차 크롤링
- 매체별 비교
- 데이터 정제
- 결과 분석
- **시사점 도출**



LCC항공사 **매출 1위**인 제주항공: **가장 다양한** 노선 보유함

▶LCC항공사들 중에서도 효과적인 차별화를 위해서는
다양하고 **유일**한 항공노선 취득하는 것이 중요

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

제주항공 나리타 제3터미널 후기	안녕하세요 :) 개강 전 마지막으로 4박 5일간 도쿄 여행 다녀왔었어요 반 년마다 매번 떠났던 여행이지만 이번에는 못가나 싶어서 조금 울적했었는데 어떻게 또 가게 되었네용 히히 **제주항공이 지방발 노선을 다양하게** 신규 취항한 덕분에 드디어 대구에서도 제주항공으로 갈 수 있는 여행의 선택...
한창 제주항공 찜특가 때문에 난리였잖아요 꽤나 다양한 노선을 팔고 있었는데 저는 인천-방콕 노선이랑 인천-세부 노선 둘 다 실패했었거든요 10시 되서 들어가면 늘 이 모양이었어요 ^^... 10시 반에도 이렇게 로딩이 길어서 그냥 포기하자 하고 있었는데 제주항공 찜특가 성공 하신 분이...	
일본자유여행 **제주항공 단독노선 마쓰야마** 특가 항공권 요즘 제가 일본에 갈때 자주 이용하는 제주항공 일본 노선 왕복 7번을 타면 1번 공짜인 일본패스 스탬프 이벤트로 최근 일본노선은 제주항공에 올인하고 있는데요 벌써 3번 이용해서 곧 후쿠오카와 오사카를 이용하면 5번째 스탬프가...	
제주항공, 무안공항 출발 도쿄·마카오 등 3개 노선 신규취항 (서울=뉴스1) 임해중 기자 | 제주항공은 오는 3월부터 **무안국제공항 기점의 도쿄, 블라디보스토크, 마카오** 등 3개 노선을 신규 운영한다고 21일 밝혔다.제주항공은 지난해 4월 오사카를 시작으로...	
안녕하세요 :) 개강 전 마지막으로 4박 5일간 도쿄 여행 다녀왔었어요 반 년마다 매번 떠났던 여행이지만 이번에는 못가나 싶어서 조금 울적했었는데 어떻게 또 가게 되었네용 히히 **제주항공이 지방발 노선을 다양하게** 신규 취항한 덕분에 드디어 대구에서도 제주항공으로 갈 수 있는 여행의 선택...	
제주항공이 3월 20일부터 인천-오사카 정기노선 취항을 시작했더라구요 인터넷 예약 할인가 199,000(TAX불포함)으로 오사카 왕복할 수 있다는 완전 멋진 가격때문에 주저없이 선택한 제주항공!! 역시 탁월한 선택이었답니다 :) 제주항공과 함께한 고고씽의 일본 오사카 여행기!! 개봉박두! 이른...	
제주항공 특가항공권-김포~제주, 일본, 홍콩, 마카오, 태국, 괌, 사이판, 블라디보스톡 등 > 제주항공 JJ멤버스위크 특가 프로모션 제주항공 국내선 및 국제선 항공권 출발일 8월 1일 ~ 8월 31일 발권일 4월 9일 ~ 4월 15일 항공사 제주항공 노선 제주항공 국내선 및 국제선(일본, 중화권...	
근데 그 때 플래카드로 **청주-오사카 노선이** 들어온다는 것을 보고 '오사카 한 번 가야겠구만~'하고 있었는데 어제 하트시그널보고(요즘 푹빠짐) 인터넷으로 하트시그널 평을 보고 있는데 제주항공 광고배너가 딱! 청주-오사카 노선이 생겼단다!!!!!! 그래서 갑자기 제주항공 어플을 깔고 항공권을...	
베트남 여행, 제주항공 취항 나트랑 정보 얻어보자 날씨가 추워지다보니 점차 따뜻한 남쪽 나라로 여행을 떠나는 분들이 많은 것 같다. 그중에서도... 참고로 이번에 제주항공 취항으로 가볼 수 있는 선택의 폭이 넓어졌다. 서울 인천발 나트랑 깜란행 노선으로 화수목토일 요일이 있고 귀국일도...	
오늘은 제주항공 을 타고 괌으로 여행 가는 분들에게 유용할 포스팅을 들고 왔어요 ! **제주항공 부산 괌 노선** 비행기 내부는 어떠한지 괌 입국신고서 작성방법과 무료로 제주항공 좌석지정 방법 등 궁금해하셨을 내용을 알려드리려고 해요 ~ 부산에서 출발하는 괌 노선 비행기는 인천공항에 비해...	

주제 선정

1차 크롤링

Data 수집 목표 수립

2차 크롤링

매체별 비교

데이터 정제

결과 분석

시사점 도출

새롭게 무안공항에 제주항공 국제선이 개항했다는 소식을 들었어요. 우와 대박. 저는 **군산에 살고 있어서 대구나 청주 국제공항**까지 가기가... 마침 제주항공의 모델이기도 하죠 무안 공항에서 바로 탈 수 있는 제주항공 국제선 노선들은 타이베이, 다낭, 방콕, 오사카 입니다. 가격이 얼마나 싼지...

티티키 시코쿠의 **마쓰야마노선은 아시아나항공이 운항하다가** 중단한것을 제주항공이 주5회로 재운항하고있다 가격적 탄력성 적극적인 현지... 변해야한다 제주항공이 제주를넘어 세계로 나가며 차별화하고있다 다른 저가항공과다르게 一포인트도 잘관리한다 그리고 앱도 상당히 잘...

제주항공 7C3106편 인천 - 괌 노선 탑승 후기 눈이 시릴 정도로 푸른 바다와 365일 따뜻한 날씨를 즐길 수 있는 괌은 다양한 액티비티는 물론 쇼핑과... 인천-괌 노선은 대한민국의 대표 LCC 항공사인 제주항공을 타고 떠났습니다. 괌이 아이들과 함께하는 가족 단위 여행객이 많은 만큼 기내 서비스는...

일본을 다녀올 때마다 착실하게 제주항공을 이용했고 그동안 연차 찬스로 다녀왔던 여행의 흔적들이 제주항공 일본 패스 스탬프로 하나 둘 모이기 시작했고 (제주항공으로 일본 노선 왕복 여행에 한해서 2016. 6월부터 2018. 5월까지 진행한 이벤트) 열심히도 돌아다닌 덕분에 일본 왕복 항공권이 1매...