유튜브 데이터를 활용한 조회수 예측 모델 개발

팀 5
팀원 1: 김보담, 2020122002, 응용통계학과, qhdamm@yonsei.ac.kr
팀원 2: 김서진, 2020122069, 응용통계학과, bwnebs1@yonsei.ac.kr

1. 서론 Introduction

1) 배경 Background

최근 스마트 기기의 보급이 활성화되고, 유튜브 플랫폼이 급성장하며 많은 사람들이 유튜브를 통해 콘텐츠를 소비하고 있다. 어떤 영상이 유익하고 재밌는지 판단하려면 우선 그 영상을 클릭하는 것이 우선이다. 이에 따라 어떤 요소가 사람들의 클릭을 만들어내는지 알아보고자 이번 프로젝트를 기획하게 되었다.

2) 목적 및 필요성 Purpose

해당 프로젝트를 통해 성공한 유튜브 콘텐츠의 특성을 보다 면밀히 파악하여, 대중과 창작자 모두를 만족시킬 수 있는 가이드라인을 제시하고자 한다.

3) 프로젝트 내용 및 의의 Significance

먼저 키워드(먹방, 브이로그, 운동, 지식, 뷰티)를 중심으로 데이터를 추출한다. 이후 추출 데이터를 바탕으로 classification 기법을 활용해 조회수를 예측하고자 한다. 또한 '키워드'를 제외한 데이터를 clustering 했을 때의 결과가 키워드로 분류한 것과 유사한지 알아볼 예정이다.

2. 데이터 Data

1) 데이터 수집 Data collection

Youtube API 를 이용하여 데이터를 직접 수집하였다. 검색어를 입력하면 보여지는 영상의 정보를 추출하였는데, 검색어로는 '먹방', '운동', '브이로그', '뷰티, '지식'으로 각각 진행하였고 영상 정보로는 '구독자 수', '제목', '영상게시일'을 수집하였다. 이렇게 진행했을 때 각 검색어 당 최대 500~600 개밖에 영상 정보가 수집되지 않아 게시 날짜를 2 달 단위로 바꿔가며 2020 년부터 최근까지의 영상을 검색하였다. 한 검색어의 한 날짜구간(2 달)에 해당하는 영상 300~500 개를 추출하여 수집한 모든 데이터를 합쳐 약 3 만개의 데이터를 얻을 수 있었다.

또한 구독자 수와 조회수는 API 를 이용할 때 쓸 수 있는 함수가 다른 요인들과 달라 한번에 추출하려면 에러가 떴다. 이를 해결하기 위하여 우선 'title', 'channel_name', 'video_id', 'channel_id', 'publish_time'을 각 영상마다 불러왔다. 그 다음 'video_id'를 읽어서 그에 해당하는 'views'를 뽑았고, 'channel_id'를 읽어서 그에 맞는 'subs'를 뽑아내었다.

2) 데이터 설명 Data description

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30122 entries, 0 to 6472
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   title         30122 non-null  object
 1   channel_name  30122 non-null  object
 2   views         29856 non-null  float64
 3   subs          26761 non-null  float64
 4   publish_time  30122 non-null  object
 5   channel_id    30122 non-null  object
 6   video_id      30122 non-null  object
dtypes: float64(2), object(5)
memory usage: 1.8+ MB
```

Figure 1 Data Type

데이터는 총 30,122 개의 행과 7 개의 열을 가진다. 'title'은 영상 제목, 'channel_name'은 채널 이름, 'view'는 조회수, 'subs'는 구독자 수, 'publish_time'은 영상 업로드일, 'channel_id'는 채널 도메인 주소의 뒷부분, 'video_id'는 영상 도메인 주소의 뒷부분을 의미한다. 하지만 영상에 따라 구독자 수와 조회수가 보이지 않는 경우가 있었다. 이를 위해 missing value 를 각 열의 평균으로 채우고자 하였으나, missing vlaue 가 포함된 행을 삭제하는 것이 모델 성능에 있어 더 낫다는 것을 확인했다. 이에 따라 후에 이 데이터들을 삭제하였다. 조회수와 구독자수는 실수형 데이터이고 나머지 열의 데이터는 문자열 데이터이다.

3) 데이터 전처리 Data preprocessing

해당 모델의 개발을 위해 API 를 통해 직접 수집한 데이터를 활용하여 전처리가 필수적이었다. 특히 제목 데이터의 경우 텍스트데이터로, 텍스트 전처리 기법을 활용했다. 검색 키워드가 한글이었으므로 대부분의 제목 역시 한글을 기본으로 작성되었을 것이라 판단, konlpy 를 사용해 전처리를 진행하였다.

가장 먼저, 문자표, 공백, 이모지 등 필요없는 정보를 삭제한 후 토큰화하였다. 이 때, 이모지의 경우 하나의 변수로 작용할 수 있을 것이라 판단하여 'emoji' 열을 새로이 추가하였다.

이후, konlpy 를 이용해 토큰화된 단어들에 대해 품사 태깅을 진행하여 명사 단어만 분석 대상으로 선정하였다. 다음으로 불용어 제거를 진행하였다. 한 글자로 이루어진 단어의 의미가 미미할 것으로 보고 한글자 단어를 제거하였으며, 채널명이 언급되는 경우 제목의 구성보다는 채널의 영향이 클 것으로 판단하여 채널명 역시 제거하였다. 해당 단어 토큰을 리스트화하여 pandas dataframe 형식으로 저장하였다.

index	label	title_new	emoji	views	subs	channel_name	publish_time	view_lv	time_lv	emoji_label	title_str
0	0	사람,번만,사람,레전드,꿀잼,드라마,뷰티,사이드,보기	false	804394.0	1960000.0	고뭉	202205	3	3	0	사람 번만 사람 레전드 꿀잼 드라마 뷰티 사이드 보기
1	0	할인,뷰마,박스,공개	true	199959.0	352000.0	뷰티마우스	202205	2	3	1	할인 뷰마 박스 공개
2	0	요즘,프로필,메이크업,연예인,프로필,메이크업,역대,메이크업	false	61394.0	159000.0	옥뷰티 OK Beauty	202205	2	3	0	요즘 프로필 메이크업 연예인 프로필 메이크업 역대 메이크업
3	0	배우,잠적,이유,뷰티,사이드,요약	false	7568.0	250000.0	DRAMA Voyage	202205	1	3	0	배우 잠적 이유 뷰티 사이드 요약
4	0	청담동,메이크업,추천,그대로,구매,지속,파운데이션,가닥,속눈썹,고급,하이,라이터,웨딩,조합	false	193907.0	133000.0	뷰티숨 BEAUTYSOOM	202205	2	3	0	청담동 메이크업 추천 그대로 구매 지속 파운데이션 가닥 속눈썹 고급 하이 라이터 웨딩 조합

Figure 2 Data Head

다음으로 제목 외 동영상 정보 전처리를 진행하였다. 먼저 게시일 정보 처리를 위해 게시일의 형태를 파악하였다. API 를 이용해 수집한 게시일 데이터는 yymmdd 형태를 띠고 있었으며, float 타입인

것을 확인했다. 즉, 작은 숫자일수록 이전에 게시된 게시물임을 이용하여 사분위수를 기준으로 4 개 등급으로 구분하였다. (0~3, 작을수록 더 오래된 게시물)

다음으로, 조회수 정보 자체를 regression 하는 대신 조회수 등급을 classification 하는 것을 프로젝트의 방향으로 선택하여, 조회수 등급 정보를 생성했다. 총 4 개 등급으로 구분하였으며, 수집한 데이터의 사분위수를 각각의 임계값으로 설정하여 등급을 나누었다. (0~3) 마지막으로, 'emoji' 정보를 0, 1 로 인코딩한 뒤 Standard Scaling 을 거치며 기본 전처리를 마무리 하였다.

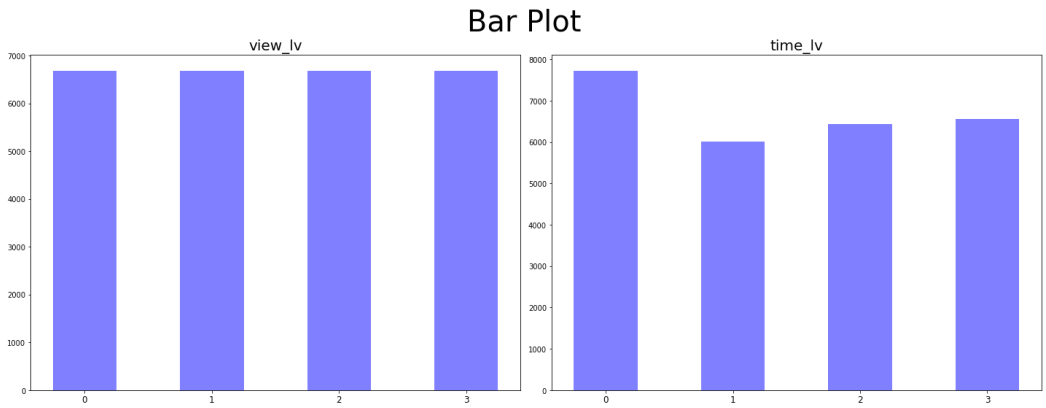

Figure 3 Discretization Result

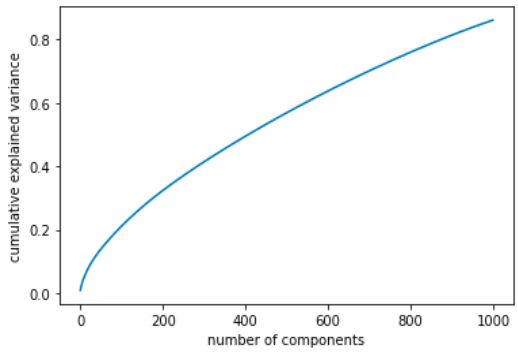

Figure 4 PCA

다음으로 벡터화를 위해 빈도정보와 tf-idf 정보를 각각 활용해 tf_matrix, tf-idf matrix 를 생성하였다. 이 때 sparse matrix 임을 감안하여 전체 토큰 26731 개 중 10 번 이상 등장한 단어를 값이 큰 순서대로 1500 개 선택하였다. 추가적으로, dimensionality reduction 을 위해 PCA 를 진행했으나, 각 PC 의 분산이 비슷한 수준이며 유의미한 차원 축소 결과를 얻을 수 없어 이는 폐기 조치하였다.

3. 방법 Methods

1) 주제 유형 Type of the analysis

주된 유형은 classification 이며, 부차적으로 clustering 을 수행하였다. Clustering 의 경우, 제목 텍스트데이터를 기반으로 하여 LDA 를 진행한 뒤, 이를 기반으로 한 K-means 분석방법과 Word2vec 기반 K-means 분석방법을 병행하였다.

2) 방법 설명 Methods description

먼저 제목 구성 정보, 이모지, 게시일 등의 변수를 통해 조회수를 예측하는 모델을 개발하고자 classification 방법론을 적용하고자 하였다. 가장 적절한 접근법 선정을 위해 pipelining 을 이용, 각 모델을 순차적으로 검증하였다. 이 결과 가장 좋은 성능을 보인 Decision Tree model 을 이용하여

모델성능을 향상시키기 위한 튜닝을 진행하였다. 앙상블 방법을 적용하여, 정확도 향상을 위해 노력하였다.

다음으로 LDA 를 진행하였다. LDA 는 토픽모델링의 일종으로, 토픽모델링은 텍스트문서 집합으로부터 주로 언급되는 핵심주제 즉, 토픽(Topic)들을 추출할 수 있도록 한다(Hofmann T., 1999). 단위문서(Document)들로 구성된 대량의 문서 집합(Corpus)에서 단어(word)들의 분포 등을 확률적으로 계산하면서 여러 개의 토픽들을 추출하고 각 문서 내에 포함된 토픽비율 등을 계산한다.(윤상훈, 김근영, 2021.03) LDA 는 디리클레 분포를 이용하여 단어들의 사후확률분포를 계산하여, 토픽 내에서 미치는 영향을 파악할 수 있다. 즉, 제목에서 어떠한 단어가 제목의 성향을 결정짓는지 파악할 수 있음을 시사한다. LDA 분석을 통해 조회수 모델에 사용된 데이터의 제목들엔 어떠한 특성이 존재하는지, 특히 조회수가 높았던 영상의 제목에는 주로 어떠한 단어가 사용되었는지 파악하여 제목 가이드라인을 제시하고자 한다.

마지막으로 Word2Vec 을 통해 제목에 사용된 각 단어간의 연관성을 파악하였다. 어떠한 단어가 사용되었을 때, 다른 단어가 함께 제목에 출현할 확률을 word2vec 모델을 통해 계산하여 함께 제목에 사용할만한 단어를 파악할 수 있도록 하였다.

4. 실험결과 Experimental Results

1) Classification

Classification 을 위해 다양한 방법론을 적용해 보았고, 가장 적절한 접근법을 선정 후 hyperparameter tuning 을 진행하였다.

앞서 'subs'와 'publish_time'을 카테고리화하여 'subs_lv'와 'time_lv' 열을 추가하였다. 최적의 모델을 찾기 위해 1) 'subs', 'publish_time', 'emoji_label', 'title_str' 활용, 2) 'subs_lv', 'time_lv', 'emoji_label', 'title_str' 활용, 3) 'subs'와 'publish_time'을 정규화한 데이터, 'emoji_label', 'title_str'을 활용한 모델, 즉 총 세 개의 모델을 비교해보았다. 이때 3)에서 정규화는 MinMaxScaler 를 이용하였다.

아래에서 선정한 Random Forest 모델에 피팅을 진행한 결과, 2)의 경우가 가장 낮은 정확도를 보였고, 1)과 3)은 유사하지만 1)이 약간 더 높은 정확도를 보였다. 따라서 1) 모델로 튜닝을 계속 진행하였다.

Case	Accuracy
1) 원본 데이터(float 형태)	0.7532710280373832
2) categorical	0.7349532710280374
3) normalization	0.7517757009345795

Figure 5 Model Tuning with different data types

Logistic regression, Gaussian Naïve Bayes, Decision Tree, K-nearest neighbor 방법을 순차적으로 적용해본 결과, Decision tree model 이 가장 좋은 성능을 보였다.

Model (if not mentioned, trained with default setting)	Accuracy Score
Gaussian NB	0.4302528334786399
K-nearest neighbors	0.5533744625163582
Logistic Regression	0.576369414843896
Logistic Regression- elastic net (l1_ratio = 0.3, C=0.1)	0.5726303982052721
Decision Tree	0.6941484389605533

Figure 6 Model Selection

이에 Decision Tree model 을 ensemble method 와 함께 사용해보고자 했다. Boosting method, 랜덤 포레스트 모델, Extra Tree Classifier 을 각각 적용해본 결과, 랜덤 포레스트 모델이 가장 높은 정확도를 보였다.

Model	Accuracy Score
Bagging(Knn, n_estimators = 1000)	0.30772106935875865
Random Forest(n_estimators = 50)	0.7264909328846514
Extra Trees Classifier(n_estimators=50)	0.7023742755655262
Ada Boost(n_estimators=50)	0.7476635514018691
Gradient Boost(n_estimators=50)	0.6512149532710281

Figure 7 Model Selection - Ensemble model

Bagging 모델의 경우, 연산량 과부하로 인해 오히려 낮은 성능을 보였으며, Extra Trees 의 경우 전체 데이터를 전수 사용하여 모델 피팅을 진행하므로 bootstrap 의 장점을 살리기 어려움을 파악할 수 있었다.

이에 추가적으로 Ada Boost 모델 튜닝과 랜덤 포레스트 모델 튜닝을 진행하였다. Random Forest 의 경우, n_estimator, max_depth, min_samples_split 을 변경하며 tuning 한 결과, 100 개의 estimator 과 빈도 정보를 통한 추정이 가장 합리적인 것으로 보였다.

Model	Tuning - Random Forest	Accuracy Score
1	n_estimators=50, max_features='sqrt', max_depth=None, min_samples_split=2, n_jobs=-1	0.7264909328846514
2	n_estimators=100, max_features='sqrt', max_depth=None, min_samples_split=2, n_jobs=-1	0.7581308411214953
3	n_estimators=1000, max_features='sqrt', max_depth=None, min_samples_split=2, n_jobs=-1	0.7221910637502337
4	X_normalized, n_estimators=100, max_features='sqrt', max_depth=None, min_samples_split=2, n_jobs=-1	0.6507758459525145

Figure 8 Random Forest parameter tuning

Ada Boost 의 경우, 아래와 같은 결과를 얻을 수 있었다.

Mode	Tuning - Ada Boost	Accuracy Score
1	DecisionTreeClassifier(), n_estimators=50, TF matrix	0.7375700934579439
2	DecisionTreeClassifier(), n_estimators=100, TF matrix	0.7476635514018691
3	DecisionTreeClassifier(), n_estimators=50, learning_rate=0.5, TF matrix	0.7465420560747663
4	DecisionTreeClassifier(), n_estimators=50, TF-IDF matrix	0.742803738317757

Figure 9 Ada Boost parameter tuning

따라서, 가장 좋은 모델은 2 번 랜덤 포레스트 모델로 선정하였다.

<Result>

Accuracy Score	r2_score
0.7581308411214953	0.7216533295422056

Figure 10 Result Model : Random Forest(n_estimators=100, max_features='sqrt', max_depth=None, min_samples_split=2, n_jobs=-1)

추가적인 성능 향상을 위해 간단한 딥러닝 모델로 적합 역시 진행해보았으나, 높은 training accuracy 에 비해 낮은 test accuracy 를 보여 효과가 미미하다고 파악, 폐기하였다. 차후 분석 진행 시, RNN 및 LSTM 모델을 개발하여 적용한다면, 보다 높은 정확도를 얻을 수 있을 것이라고 생각한다.

2) Topic Modeling - LDA

```
Topic ID: 0
        센터      0.05003736913204193
        산업      0.04329858720302582

Topic ID: 1
        카페      0.05301668494939804
        요리      0.016956185922026634

Topic ID: 2
        브이      0.18208636343479156
        로그      0.17263294756412506

Topic ID: 3
        지식      0.11508319526910782
        먹방      0.08553826063871384

Topic ID: 4
        운동      0.02961881458759308
        다이어트  0.0187144223600626
```

Figure 11 LDA - Total

동일 TF IDF 로 LDA 역시 진행하였다. gensim 모듈을 활용한 LDA 를 통해 주제별 단어 분포를 구하였고, 각 주제별로 단어들의 확률 분포를 구했다. 이후 단어들의 주제별 분포를 이용해 클러스터링을 진행하였다. 이를 이용하여, 키워드를 통해 수집한 영상정보들과 실제로 수집된 영상의 주제 정보가 유사한지 파악하고자 하였다. 또한, 각 주제에 자주 등장하는 단어와 해당 단어가 주제를 결정하는데 미치는 영향력을 파악할 수 있었다. 데이터 수집 시 키워드 수가 5 개였으므로, num_topics = 5 로 LDA 를 진행하였다. 그 결과, 5 개의 topic 에서 가장 영향력있는 단어 상위 2 개를 프린트 하였다.

미리 설정했던 주제들과는 달리, 도출된 주제들은 [산업관련, 요리 관련, 브이로그, 먹방과 지식, 그리고 운동] 이었다. '지식' 주제의 경우 검색했을 때 '다방면의 정보 전달 영상'이 뜰 것이라는 기대와 달리, '지식산업센터'와 관련된 영상이 많이 노출되어 결과가 저렇게 나온 것으로 추정된다. 추가적으로, '지식' 키워드로 추출한 데이터를 보면 '이렇게 비싼 치킨! 왜 남는 게 없다고 할까?' 등 '음식'과 관련된 주제를 다루는 영상이 많았다. 또한, '먹방' 키워드로 검색해 얻은 데이터 역시 '먹방 브이로그', '먹방 여행' 등 부차적인 주제가 추가된 경우가 많아, 예상과는 다른 결과가 도출되었다.

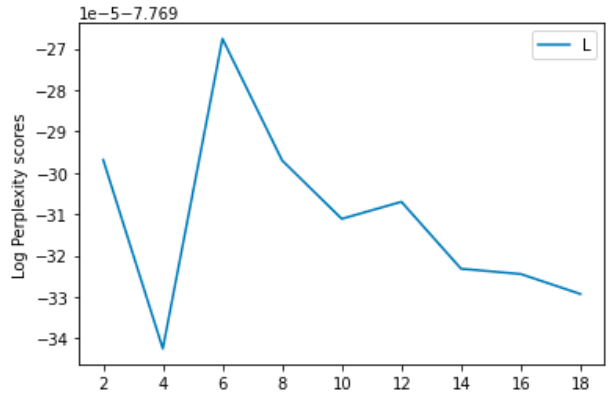

Figure 12 log perplexity score

다음으로 조회수 라벨이 3 인 데이터(조회수 상위 25%)를 대상으로 LDA 를 다시 진행하였다. 이 때, 상위 영상들의 가장 적합한 군집 수를 결정하기 위해 log perplexity score 을 파악했다. 일정한 분포를 따르지 않는 불규칙한 데이터로 인해 Perplexity score 도 불안정적인 그래프 형태를 보이다, 7 이후로 감소하는 것을 확인할 수 있었다. 앞선 값들 중, 군집수가 4 일 때 가장 score 가 낮기 때문에, 해당 군집수로 분석을 진행했다.

```
Topic ID: 0
        브이      0.10641777515411377
        로그      0.10344487428665161


Topic ID: 1
        먹방      0.06760582327842712
        브이      0.0469050258398056


Topic ID: 2
        브이      0.07204253226518631
        로그      0.06523484736680984


Topic ID: 3
        먹방      0.03509318083524704
        공부      0.018101409077644348
```

Figure 14 LDA- high views

이 때, 0 번 주제는 먹방과 브이로그, 1 번 주제는 일상과 브이로그, 2 번 주제는 운동과 브이로그, 3 번 주제는 브이로그로 나타났다. 결과는 '먹방', '브이로그' 가 가장 빈번하게 등장하며, 모든 주제에 가장 큰 영향을 주는 것으로 나타났다. 이는 상위 조회수를 보이는 영상에서 '먹방'과 '브이로그' 키워드 영상의 비율이 높기 때문인 것으로 판단하였다. 또한, 모든 주제에서 '브이로그'라는 키워드를 가지는 영상들이 많으며, 이는 최근 일상을 공유하고 친근하게 접근하는 영상들이 주를 이루는 유튜브 트렌드를 반영한 것으로 파악하였다.

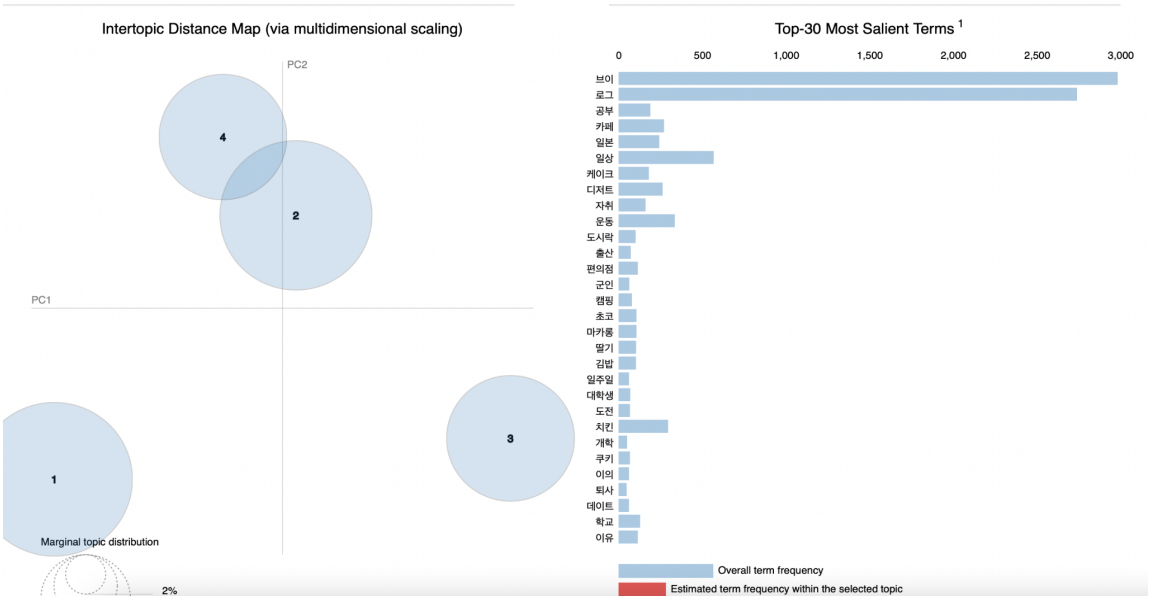

Figure 13 LDA visualization

pyLDAvis 모듈을 활용해 보다 자세하게 살펴보면, '먹방' 관련 주제의 경우 음식 이름을 특정지어 쓰는 경우가 다수였으며, '치킨', '치즈', '딸기', '편의점', '불닭', '뿌링클' 등 자극적이거나 많은 사람들이 선호하는 음식들이 상위 키워드로 자리하였다. '브이로그'의 경우 '자취', '일상', '집밥', '학교' 등 일상과 연관된 단어들이 상위 키워드로 나타났다.

마지막으로 word2vec 군집화 분석을 진행하였다. gensim 의 Word2Vec 모듈을 활용, 주변 3 단어와의 관계를 반영하여 100 차원 워드벡터로 단어 토큰들을 반환하였다. 이 때 생성된 모델에 각 키워드를 검색했을 시 나타나는 유사 단어들은 다음과 같았다.

index	먹방	운동	뷰티	지식	브이로그
0	떡볶이	산소	헤어	투자	일상
1	디저트	홈트	웨딩	분양	공부
2	치킨	루틴	트럭	센터	무채색
3	리얼	전신	리뷰	청라	카페
4	불닭	근력	청담동	타워	토론토
5	치즈	올인원	셀럽	산업	랩스
6	젤리	복근	차홍	정보	직장인
7	레시피	에센셜	얼굴	급량	급똥
8	라면	하체	진정	가산동	스터디
9	사운드	뱃살	머리	동서	마켓
10	드랍	햄스트링	살롱	유망	당근
11	김밥	허리	성용	고덕	코지
12	아이스크림	통증	알로	안양	제본
13	불닭볶음면	어깨	민낯	인덕원	일본
14	편의점	다이어트	눈썹	금호	신주쿠
15	직접	걷기	이시영	임대	하루
16	새우	가지	틴트	옥정	브이
17	김치	스트레칭	수분	드라이브인	비티
18	삼겹살	코어	변신	평택	내복
19	양념치킨	무릎	티켓	수원지	도쿄
20	집밥	최고	쿠션	타운	몸부림
21	까르보	식단	발렌티노	투기	시험
22	메뚜	상체	패션	공장	중학생
23	케이크	프로그램	천연	배곧	기간
24	튀김	층간소음	커버	과천	포장

Figure 15 Word2Vec Results

'먹방' 의 경우 음식과 관련된 단어들이, '운동'의 경우에는 상체, 하체 등 운동 부위 관련 단어들이, '뷰티'에서는 헤어, 웨딩, 민낯 등의 단어들이 눈에 띄었다. '지식'의 경우 지식산업센터 관련 단어들이 주를 이루었고, '브이로그'의 경우 일상과 공부가 상위 카테고리를 차지했다.

5. 결론 및 제언 Conclusions and Discussion

1) 프로젝트 진행 내용 개요 Summary

우선, Youtube API 를 이용하여 검색어를 기준으로 최근 2 년동안의 영상에 대한 '조회수', '제목', '채널 이름', '업로드 날짜', '구독자 수'를 수집하였다. 그 다음 konlpy 를 이용하여 전처리를 진행하였다. 'title'에서 이모티콘의 사용 여부를 보여주는 'emoji_label' 열을 추가하였고, 문자는 토큰화하여 명사 단어만 추출하였다. 'publish_time'의 경우 'yyyymm'의 형식으로 알아보기 쉽게 가공하였고, 추가로 시간이 빠른 순서대로 0, 1, 2, 3 의 숫자를 부여하여 실수형 데이터를 카테고리화한 'time_lv'열을 추가하였다. 'subs'와 'views'도 위와 마찬가지로 숫자가 작은 순서대로 0, 1, 2, 3 의 숫자를 부여하여 각각 'subs_lv'와 'views_lv'열을 만들어주었다.

'title' 데이터를 벡터화한 후 다양한 방법으로 classification 을 진행해 본 결과 Random Forest 와 Ada Boost 가 가장 높은 정확도를 보였다. 그러나 각각의 모델 튜닝을 통해 Random Forest 가 최선이라고 파악하였고 그 결과 약 0.76 의 정확도를 가진 모델을 얻을 수 있었다.
또한 LDA 의 결과 '먹방'과 '브이로그'가 상위 키워드로 떠올랐다. 이를 활용하여 pyLDAvis 를 통해 '먹방'과 '브이로그' 관련 영상의 제목을 살펴보니 '먹방'의 경우 음식을 명시하고, '브이로그'의 경우 일상을 공유하는 영상이 트렌드인 것을 확인하였다. 마지막으로, word2vec 을 통해 각 키워드별로 어떤 주제를 가진 영상이 높은 조회수를 불러오는지 알 수 있었다.

2) 프로젝트의 결과 및 한계점 Limitation

classification 을 위해 다양한 방면으로 성능을 높이고자 시도해보았지만 모델이 기대보다 낮은 성능을 보였다. '제목'과 '구독자 수'로 classification 을 했을 때 모델은 최대 0.73 의 정확도를 보였으나 '업로드 일' 변수를 추가한 후에는 0.75 이상의 정확도를 보였다. 이를 통해 유튜브 데이터의 다른 변수를 추가하면 성능이 높아질 것으로 예상하였으나 크롤링을 위한 API 의 일일할당량이 제한되어있어 '좋아요 수', '댓글 수'와 같은 다른 데이터를 수집하기 어려웠다.

또한 실제 크롤링 데이터가 예상했던 주제를 보이지 않는 경우가 존재했다. 예를 들어 '뷰티' 관련 영상에 예상치 못한 '과천' 키워드의 등장과 같이, 예측 가능 범위를 넘어선 데이터들의 존재가 분석에 걸림돌이 되었다. 추후 분석을 진행한다면, 키워드로 주제 자체를 검색하기 보다는 하나의 종류나 특성을 검색하는 것이 바람직할 것으로 보인다. 예를 들어, '음식' 대신 '치킨'을 검색하는 등의 방법으로 이를 해결할 수 있으리라 본다.

3) 프로젝트의 기대효과 및 활용방안 Contributions

이 프로젝트는 2020 년부터의 데이터를 활용하였기 때문에 최근 유튜브 플랫폼의 변화 동향과 트렌드 분석에 유용하게 작용할 수 있다. 또한 유튜브를 처음 시작하고자 하는 사람에게 방향성을 제시할 수 있고 이미 개인 채널을 운영하고 있는 사람이라면 자신의 채널이 최신 트렌드를 파악하고 있는지 확인할 수 있다. classification 모델에 본인의 영상 제목을 넣어봄으로써 기대하는 조회수를 얻을 수 있는지 확인하고, 그렇지 않다면 우리가 부수적으로 진행한 word2vec 등을 활용하여 영상을 보완할 수 있다. 또한 한국 시장에 진출하고자하는 기업의 마케터들도 이 프로젝트 활용하면 한국 소비자들의 니즈를 파악하는데 용이할 것이다.

6. 참고문헌 References

1) 김나경 (Na-gyeong Kim), 김정민 (Jeong-min Kim), 이혜원 (Hye-won Lee),and 국중진 (Joong-jin Kook). "머신러닝을 이용한 유튜브 악성 댓글 탐지 시스템." 한국정보처리학회 학술대회논문집 28.2 (2021): 775-778.
2) Taewon Yoo(유태원),and Hyunggu Jung(정형구). "Category Classification of Educational Videos on YouTube through Machine Learning Approaches." 한국정보과학회 학술발표논문집 2019.6 (2019): 1914-1916.
3) 김형욱,and 송진웅. "유튜브 과학 채널에 대한 이용실태 분석 및 채널 판별 예측 모형 평가 -소셜 빅데이터 분석 및 머신 러닝 활용을 중심으로-." 교육공학연구 36.2 (2020): 383-412.
4) Yoon, S. H., & Kim, K. H. (2021). Expansion of Topic Modeling with Word2Vec and Case Analysis. *The Journal of Information Systems*, *30*(1), 45–64. https://doi.org/10.5859/KAIS.2021.30.1.45