YouTube 조회수 예측 및 상관관계 분석

201820734 신우현

202021191 정윤미

201820703 양성욱

201820743 김준서

202020798 박준우

**1. 프로젝트 소개(Project Introduction)**

**1-1. Background**

유튜브는 구글이 서비스하는 동영상 공유 플랫폼으로, 전세계 최대 규모의 동영상 공유 및 호스팅 사이트로서 이용자가 영상을 시청, 업로드 및 공유할 수 있다. 2020년 10월 아이지 에이웍스에서 조사한 자료에 따르면 한 달 동안 대한민국 인구의 83%인 4319만 명이 유튜브 앱을 사용한 것으로 나타났다. 유튜브 사용자의 수가 증가함에 따라 유튜버를 시작하는 사람의 수도 함께 증가했는데, 이 프로젝트에서는 조회수를 예측하는 모델을 구현하고, 어떤 요소들이 조회수와 관련이 분석하여 유튜브 영상을 제작할 때 어떤 요소를 고려해야 하는지를 분석하고자 한다.

**1-2. Requirements/Assumptions/Risks/Constraints**

조회수 예측 모델을 설계하기 전, 본 프로젝트에는 몇 가지 제약 조건이 있다. 첫째, 유튜버 자체의 인기는 고려하지 않는다. 이미 많은 구독자를 가진 대형 유튜버는 어떤 영상을 올리든 일정 수준 이상의 조회수가 보장되어 있다. 이 경우에는 영상 제작 시 고려해야 할 요소라고 보기엔 어려우므로 특정 구독자 수를 가지는 유튜브 채널의 정보만 수집하여 구독자 수가 조회수에 미치는 영향을 최소화하고자 한다.

둘째, shorts 영상은 분석 대상에서 제외한다. Shorts 영상의 경우, 영상을 업로드한 유튜브 채널의 평균 조회수보다 휠씬 높은 경우가 많고, 영상 자체의 길이가 짧기 때문에 사람들이 영상을 소비하는데 거리낌이 적다. 따라서 shorts 영상은 영상의 길이를 제외하면 조회수에 영향을 주는 요소가 적다고 판단했고, 또한 일반 영상에 비해 높은 조회수가 outlier가 될 수 있다고 판단하여 제외했다.

셋째, 썸네일 요소는 고려하지 않는다. 썸네일이 조회수에 영향을 미친다는 것은 부정할 수 없는 사실이다. 하지만, 어떤 썸네일이 조회수에 많은 영향을 미치는지는 그 기준이 모호하다. 예를 들어, 인물의 얼굴이 썸네일에 나타났다고 하면 그 인물이 누구인지, 그 인물의 외모가 어떤 지, 그 인물이 어떤 행동을 하고 있는지에 따라 영상의 조회수에 영향을 주는 정도가 다르다. 따라서, 본 프로젝트에서는 영상의 썸네일이 조회수에 미치는 영향이 동일하다고 가정했다. 또한, 썸네일 에서는 주로 영상 내용과 관련된 내용이 담기는데 이 정보는 영상의 제목으로도 알 수 있으므로 썸네일은 고려하지 않는다.

**2. 데이터 준비(Data preparation)**

데이터 수집은 유튜브 채널 수집, 유튜브 영상 정보 수집, 수집한 정보 전처리 과정을 거쳐 이루어진다.

1. **유튜브 채널 수집**

유튜브 영상 정보를 수집하기 전에, 먼저 영상을 수집할 유튜브 채널을 수집했다. 수집할 유튜브 채널은 ‘블링(vling)’이란 사이트에서 수집했다. 블링은 광고주용 유튜브 채널 검색 사이트이며 여러 태그를 선택하여 원하는 조건의 유튜브 채널을 검색할 수 있다.

이 프로젝트에서 수집한 채널의 조건은 채널 카테고리가 패션, 뷰티, 푸드/먹방, 엔터테인먼트, Vlog/일상, 여행, ASMR, 게임, 펫/동물에 속하는 유튜브 채널, 채널의 구독자 수가 45000에서 50000 사이인 채널을 수집했다. 채널 수집 중 일일 조회수가 0이거나 영상 별 평균 조회수가 0인 경우, 정상적인 채널일 아니라고 간주하여 수집 대상에서 제외했다. 채널 수집 결과 최종적으로 175개의 채널을 수집할 수 있었다.

**2) 유튜브 영상 정보 수집**

유튜브 채널 수집 후, 유튜브 영상 정보를 수집하기 위해 YOUTUBE API를 사용했다. YouTube Data API는 유튜브와 관련된 기본적인 API로 검색결과, 재생목록, 채널 등을 기준으로 영상별로 영상의 길이, 조회수, 댓글 수 등의 정보를 불러올 수 있다. 이 프로젝트에서는 YouTube Data API v3를 이용해 영상 정보를 수집했고 최신 트랜드를 반영하기 위해 채널별로 최근에 업로드한 최대 50개의 영상을 수집했다. 최종적으로 7715개의 영상 정보를 수집할 수 있었다. 구체적인 수집 과정은 다음과 같다.

먼저, 1)에서 수집한 채널을 youtube.search()에 채널 명을 넣어 유튜브 채널의 id를 가져온다. 수집한 채널 id를 youtube.channels()의 검색 조건으로 넣어 채널 정보를 가져오고, ['contentDetails']['relatedPlaylists']['uploads']로 ‘업로드한 동영상’ playlist id를 가져온다. 이후, youtube.playlistItems()에 수집한 playlist id를 검색 조건으로 넣어 해당 playlist에 들어있는 영상의 id를 가져온다. 마지막으로 youtube.videos()에 영상 id를 검색 조건으로 넣어 해당 영상의 상세 정보를 가져온다. 수집한 영상 정보는 다음과 같다.

영상 제목: ['snippet']['title']를 통해 읽을 수 있다.

영상 업로드 시간 : ['snippet']['publishedAt']를 통해 읽을 수 있다.

영상 태그 : ['snippet']['tags']를 통해 읽을 수 있다.

좋아요 수 : ['statistics']['likeCount']를 통해 읽을 수 있다.

댓글 수 : ['statistics']['commentCount']를 통해 읽을 수 있다.

영상 길이 : ['contentDetails']['duration']를 통해 읽을 수 있다.

카테고리 id : ['snippet']['categoryId']를 통해 읽을 수 있다.

영상 조회수 : ['statistics']['viewCount']를 통해 읽을 수 있다.

**3) 수집한 데이터 전처리**

1. **데이터 결측 값 처리**

데이터 전처리의 첫 단계로 수집한 영상 데이터에서 결측 정보들을 살펴보았다.

테이블이(가) 표시된 사진

자동 생성된 설명

[데이터 결측치 정보]

먼저 543개 영상의 좋아요 수와 댓글 수 항목에서 결측치들을 확인할 수 있었다. 또 Tag항목의 경우에는 7700여개의 영상 중 무려 5606개의 영상에서 결측치가 있는 것을 확인할 수 있었다. 이 결측치들을 처리하기 위해 우선 tag feature는 데이터에서 제거하기로 결정했다. 그리고 좋아요 수와 댓글 수의 결측 값들은 API로 데이터를 수집할 때 좋아요 수와 댓글 수가 0인 경우에 수집을 하지 않은 경우였다. 좋아요 수와 댓글 수가 0인 경우도 조회수 예측에 의미가 있을 것으로 판단하여 제거하지 않고 0으로 대체하였다.

1. **데이터 이상값 처리**

수집한 데이터의 feature들 중에서 이상값 처리가 필요한 feature로 길이와 조회수를 선정하였다. 먼저 영상 길이의 이상치를 처리하였다. short영상을 제거하기 위해 1분 미만의 영상을 제거하였고, 유튜브 이용자들의 짧은 영상에 대한 수요와 너무 긴 영상이 조회수 예측에 주는 영향을 고려하여 1시간 이상의 영상은 제거하였다. 그 다음 조회수의 이상치를 처리하는 과정을 거쳤다. 전처리 전 수집한 데이터들의 조회수의 값들을 분석해보았다.

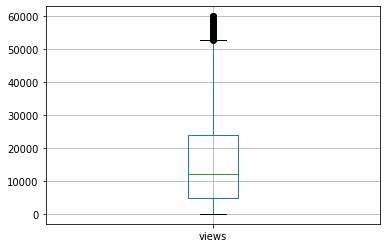
텍스트이(가) 표시된 사진

자동 생성된 설명

[조회수 수치]

위 그림에서 볼 수 있듯 최소와 최댓값이 중앙값에 비해 극단적인 것을 확인할 수 있었다. 그리고 중앙값이 15000 정도인 것에 비해 평균이 28만을 넘은 것을 확인할 수 있는데 이는 조회수 상위 40개 항목만 해도 조회수가 천만을 넘는 극단적인 값들을 가지고 있는 것이 가장 큰 요인으로 판단된다. 그런 영상들의 경우 우리가 수집한 feature들의 영향보다는 그 때의 유행이나 유튜브 알고리즘 등으로 인해 특별히 조회수가 상승한 것으로 생각할 수 있다. 그러므로 일정 수치 이상의 조회수 데이터에 대해 제거가 필요할 것으로 보인다.

테이블이(가) 표시된 사진

자동 생성된 설명

[이상치 제거 전 boxplot] [조회수 6만 이상 영상 제거 후 boxplot]

먼저 이상치 제거 기법으로 IQR 기법을 생각해볼 수 있는데, 조회수에 대한boxplot을 그려본 결과 박스의 크기가 너무 작아 outlier가 너무 많았고, 이를 적용해본 결과 대부분의 데이터가 제거되어 바로 적용하기는 어렵다고 판단하였다. 이후 데이터의 손실과 예측 성능 향상을 위한 효과적인 outlier제거를 고려하여, 조회수6만 이상의 영상들과 조회수 1천 미만의 영상들을 제거하였다.

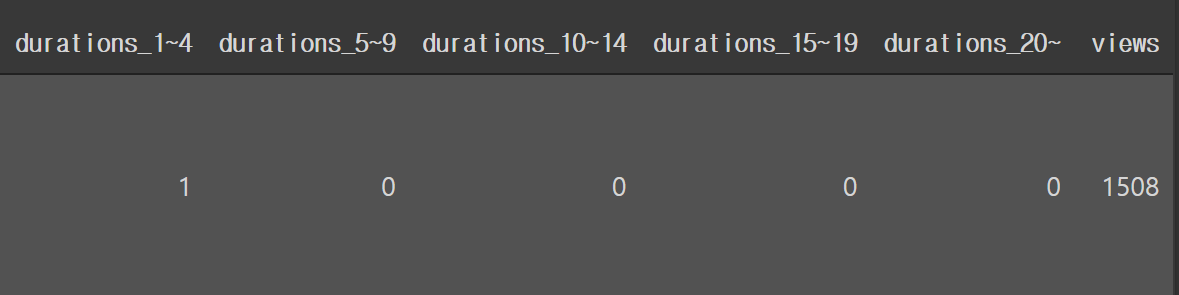
1. **데이터 전처리**

영상 업로드 시간과 조회수의 관계를 알아보기 위해서는 영상 업로드 시간 feature를 조회수 예측 모델에 용이하게 변환하는 과정이 필요했다. 우리는 조회수가 계절에 따라, 밤낮에 따라 연관이 있을 것으로 생각하여 object 타입이었던 publishedAt feature를 시계열 데이터로 변환한 뒤, ‘month’를 이용하여 3,4,5월은 봄으로, 6,7,8월을 여름으로, 9,10,11월은 가을, 12,1,2월은 겨울로 나누어 계절을 one-hot encoding 하였고, ‘hour’을 기준으로 오전 6시부터 오후 18시까지는 낮으로, 그 외의 시간은 밤으로 나누어 one-hot encoding 하였다.



[계절, 밤낮 one-hot encoding]

위 그림의 데이터의 경우 3월 24일 7시경에 업로드 된 영상이다. 3월이므로 봄에 1을 오전 7시이므로 낮의 값을 1로 값을 설정해 놓은 것을 볼 수 있다.

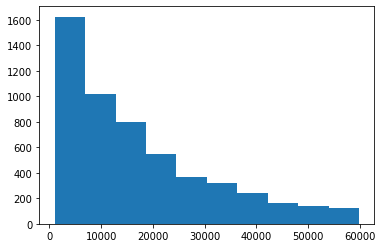


[영상 길이 one-hot encoding]

다음은 영상 길이 별로 조회수와 관계가 있을 것으로 생각해 5분 미만, 5분에서 10분, 10분에서 15분 미만, 15분에서 20분 미만, 20분 이상의 영상으로 나누어 one-hot encoding 하였다. 위 그림의 경우 5분 미만의 영상임을 확인할 수 있다.

또한 데이터 모델링을 진행하기 전, 데이터 값 평준화를 위해 정규화를 진행해주었다. Min-max 정규화를 이용하여 진행하였다. 정규화는 One Hot encoding을 거쳐 0과 1의 값 만을 가지는 feature를 제외하고 진행하였다.

1. **EDA**



**3. 모델링(Modeling)**

회귀 분석은 둘 이상의 변수 간의 관계를 보여주는 통계적 방법이다. 선형 회귀는 실제 값과 예측 값의 차이를 최소화하는 직선형 회귀선을 최적화하는 방식이다. 선형 회귀 모델은 규제(Regularization) 방법에 따라 다시 별도의 유형으로 나눌 수 있다. 규제는 일반적인 선형 회귀의 과적합 문제를 해결하기 위해 회귀 계수에 패널티 값을 적용한 것을 말한다. 조회수를 예측하기 위해서 우리는 다중 선형 회귀, Ridge 선형 회귀, Lasso 선형 회귀, Random forest regression, SVR을 이용하여 테스트를 해보았고 이들의 성능을 비교하였다.

**1) 다중 선형회귀**

다중 선형회귀는 단순선형회귀와 같이 독립변수 X의 변화에 따른 종속변수 y의 변화를 선으로서 예측하는 기법이다. 단순선형회귀와의 차이점은 독립변수 X가 여러 개인 분석기법이라는 점이다. 따라서 feature가 여러 개일 때 사용할 수 있기 때문에 일반 선형회귀보다 더 좋은 성능을 기대할 수 있다.

선형 회귀에서 어떻게 최적의 가중치와 편향을 찾아내기 위해서는 라벨값(y)과 예측값(y^) 사이의 평균제곱오차(Mean Squared Error, MSE)를 최소화하는 파라미터 w와 b를 찾아야한다.

그런데 다중 선형 회귀 모델은 과적합(overfitting)되는 경향이 있다. 이로 인해 일반화 능력(범용능력)이 떨어져 새로운 데이터를 제대로 예측하지 못한다. 이를 해결하기 위한 것이 바로 라쏘(LASSO)와 릿지(Ridge)이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다중 선형 회귀 모델을 사용한 결과, R2 score는 약 0.332가 나온 것을 Google Colab을 통해 확인할 수 있다.

1. **LASSO regression**

LASSO는 MSE가 최소가 되게 하는 가중치와 편향을 찾으면서 동시에, 가중치들의 절댓값의 합이 최소가 되게 한다. 즉, 가중치의 모든 원소가 0이 되거나 0에 가깝게 되도록 만든다. 따라서 어떠한 특징들은 모델을 만들 때 사용되지 않기도 한다. 어떤 벡터 요소의 절댓값의 합은 L1-norm이다. 즉, LASSO는 L1-norm 페널티를 가진 선형 회귀 방법이다.

MSE항이 작아질수록 라벨 값과 예측 값들의 차이가 작아지고, L1-norm이 작아질수록 많은 가중치들이 0이 되거나 0에 가까워진다. α가 크면 전체 값(MSE+penalty)이 L1-norm의 영향을 크게 받아 L1-norm을 작게 만드는 것이 우선시된다. 반면, α가 작으면 MSE가 작아지게 하는 것에 더 비중을 두게 된다. 따라서, α가 너무 작으면 overfitting이, 너무 크면 underfitting이 일어난다.

텍스트이(가) 표시된 사진

자동 생성된 설명

최적의 파라미터 α을 찾기 위해 5-fold Cross Validation 사용하였다. 그 결과 α = 0.01일 때 가장 높은 score를 보임을 알 수 있었고, α의 값을 0.01로 지정하여 Training과 Test를 진행하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

LASSO regression를 사용한 결과, R2 score가 약 0.031이 나온 것을 Google Colab을 통해 확인할 수 있다.

1. **Ridge regression**

Ridge regression은 L1-norm 대신에 L2-norm 페널티를 가진다. LASSO는 가중치들이 0이 될 수 있지만, LIDGE의 가중치들은 0에 가까워질 뿐 0이 되지는 않는다.  따라서, 특성이 많은데 그중 일부분만 중요하다면 LASSO가, 특성의 중요도가 전체적으로 비슷하다면 RIDGE가 좀 더 적합하다.

텍스트이(가) 표시된 사진

자동 생성된 설명

LASSO와 마찬가지로, 최적의 파라미터 α을 찾기 위해 5-fold Cross Validation 사용하였다. 그 결과 α = 1일 때 가장 높은 score를 보임을 알 수 있었고, α의 값을 0.01로 지정하여 Training과 Test를 진행하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Ridge regression를 사용한 결과, R2 score가 약 0.332이 나온 것을 Google Colab을 통해 확인할 수 있다.

1. **RandomForest Regressor**

랜덤포레스트는 의사결정트리를 이용한 학습 결과가 성능과 변동의 폭이 크다는 단점을 극복하기 위해 등장한 방식이다. 주어진 Training set에서 무작위로 중복을 허용하여 n개 선택하고 선택한 n개의 데이터 샘플에서 데이터 특성 값을 중복 허용 없이 d개 선택한다. 이를 이용해 의사결정 트리를 학습하고 생성하는 것을 k번 반복하고 생성된 k개의 의사결정트리를 이용해 예측하고, 예측된 결과의 평균으로 최종 예측 값을 결정한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

RandomForest 회귀의 하이퍼 파라미터 n\_estimators(생성할 트리의 개수), max\_features(최대 선택할 feature의 개수)를 찾기 위해 5-fold Cross Validation을 사용하였고, max\_features = 8, n\_estimators = 50일 때 가장 높은 Score가 나옴을 확인할 수 있었다. 따라서 이들을 하이퍼 파라미터로 지정하여 모델을 생성하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

RandomForest regression를 사용한 결과, R2 score가 약 0.588이 나온 것을 Google Colab을 통해 확인할 수 있다.

1. **SVR**

Support Vector Regression 알고리즘은 Linear Regression이 과적합되는 것을 피하기 위해 사용되는 Ridge, Lasso와 같은 정규화 방법 중 하나이다. 아래 두 식은 SVR의 수식과 이를 마진 값 ϵ를 사용하여 나타낸 식이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

ϵ는 회귀식 위 아래에 생성되는 마진의 크기, 즉 오차의 허용 범위이고, ξ, ξ\*는 마진의 위 아래에 생성된 추가범위이다. SVR은 데이터에 노이즈가 있다고 가정하며, 이러한 점을 고려하여 노이즈가 있는 실제 값을 완벽히 추정하지 않는다. 따라서 적정 범위(2ϵ) 내에서는 실제 값과 예측 값의 차이를 허용한다. 이때 사용자는 C를 조절함으로써 ξ, ξ\*값을 조정, 허용 범위를 설정할 수 있다. 따라서 C값이 작을수록 회귀식은 평평한 모양을 가지게 되고, 반대로 클수록 구불구불한 모습을 띄게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명



5-fold Cross Validation 사용하여 하이퍼 파라미터 kernel, C, gamma를 찾고자 하였다. kernel=rbf, C=30, gamma=0.1에서 가장 높은 score가 도출되었고 이들을 파라미터로 하여 모델링하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

SVR을 사용한 결과, R2 score가 약 0.466이 나온 것을 Google Colab을 통해 확인할 수 있다.

**4. 평가(Evaluation)**

**1) 모델 평가**

텍스트이(가) 표시된 사진

자동 생성된 설명

R2 score은 회귀 모델에서 독립 변수가 종속 변수를 얼마나 잘 설명해주는 지를 보여주는 지표이다. R2 score는 0과 1사이의 값을 가지며 1에 가까울수록 선형회귀 모델이 데이터에 대하여 높은 연관성을 가지고 있다고 해석할 수 있다. 각 모델에서의 R2 score은 다음과 같다.

*-다중 선형 회귀: R2 Score = 0.332*

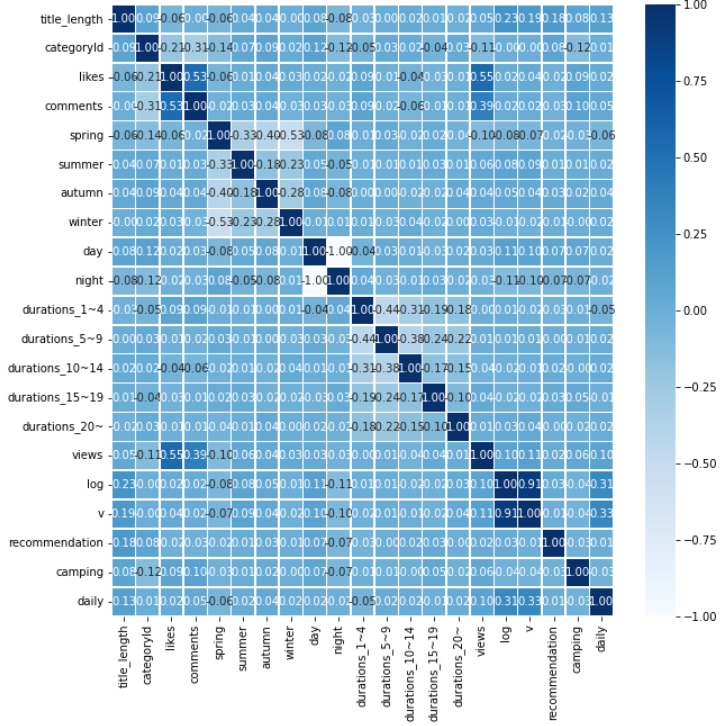
*-Lasso Regression: R2 Score = 0.031*

*-Ridge Regression: R2 Score = 0.332*

*-RandomForestRegressor: R2 Score = 0.588*

*-SVR: R2 Score = 0.466*

모델의 성능은 RandomForestRegressor > SVR > 다중선형회귀 Ridge회귀 > Lasso 회귀 순으로 우수하다는 것을 알 수 있다. 즉, RandomForestRegressor 모델로 조회수를 예측하였을 때 가장 높은 성능을 보인다는 것을 확인할 수 있었다. Lasso Regression는 R2 Score가 0에 가까우므로 조회수 예측 모델로 이용하기에 적합하지 않은 것으로 판단된다.

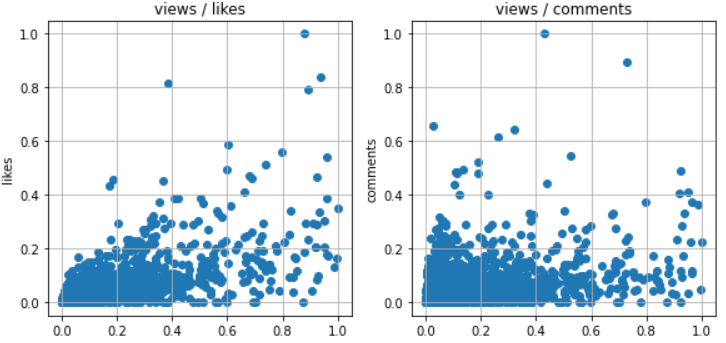


**2) 상관관계 분석**

위 그림은 각 feature들 간의 상관관계를 나타낸 heatmap이다. 먼저 각 feature들이 유튜브의 조회수와 상관관계를 가지고 있는지에 관해 알아보고자 하였으나 heatmap을 그려본 결과 조회수와 높은 상관관계를 가지는 feature는 찾을 수 없었다. 좋아요 수와 댓글 수가 각각 0.55, 0.39로 비교적 높은 상관관계를 보이고 있으나, 좋아요와 댓글을 남기기 위해서는 먼저 동영상을 조회해야 하기 때문에, 반대로 조회수가 좋아요 및 댓글 수에 영향을 준다고 해석해야 한다.

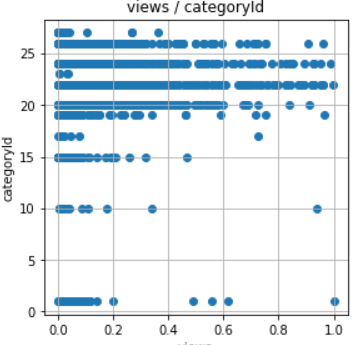
또한 각각의 feature와 조회수의 plot 그래프를 그려보았다. Plot 그래프에서 x축은 조회수, y축은 각각의 feature이다. 조회수가 높은 곳에서 밀도가 높은 특성이 조회수에 영향을 미친다고 판단할 수 있다. One hot encoding 과정을 거친 feature들의 경우 0과 1의 값 만을 가지기 때문에 각 그래프 아래에 0값의 그래프가 나타나는데, 이는 해당 feature을 제외한 다른 feature들의 값이 합쳐진 그래프로 해석되므로 무시한다.

**-Likes, Comments**



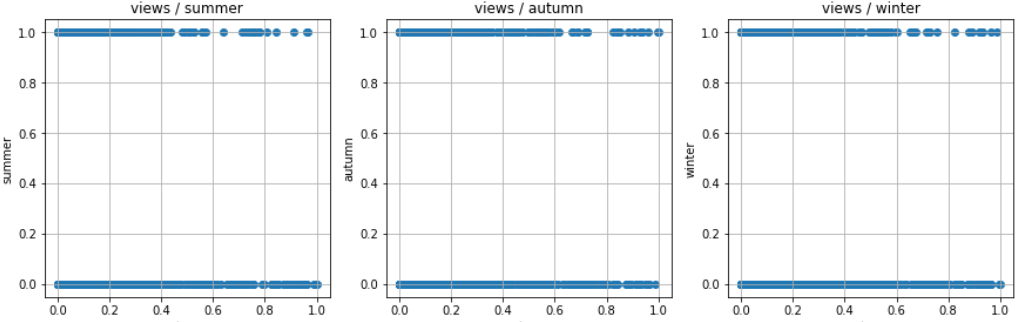
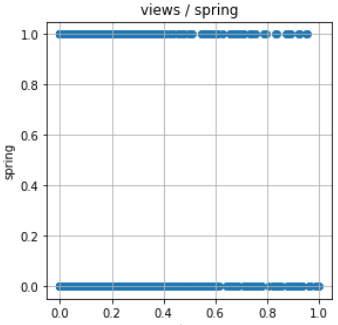
위 그림과 같이 좋아요 및 댓글 수는 조회수와 높은 상관관계를 보이지만 동영상을 먼저 시청해야만 좋아요와 댓글을 남길 수 있기 때문에 조회수에 미치는 영향이 아니라 조회수가 이 특성들에 미치는 영향이라고 할 수 있다.

**-Category Id**



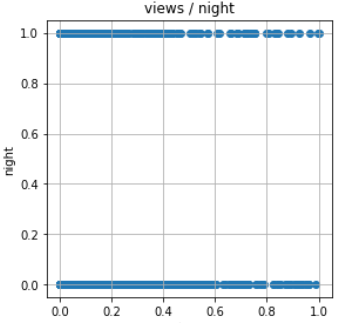
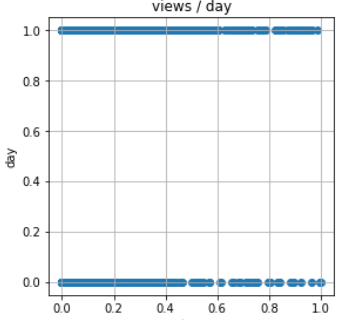
Category Id feature의 경우 다른 Id와 비교했을 때 22번(People & Blogs)과 24번(Entertainment)이 조회수가 많은 쪽에서 높은 밀도를 보였다. 즉, people & Blogs와 Entertainment 카테고리에 속한 동영상이 조회수가 높은 경향이 있다고 볼 수 있다.

**-계절**



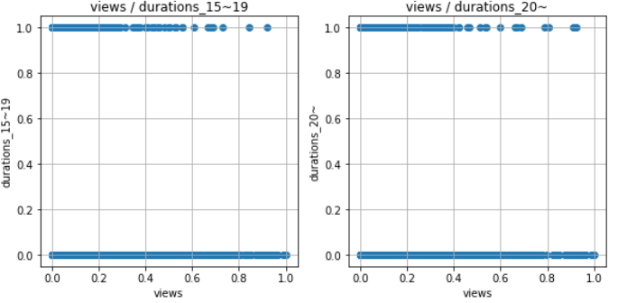
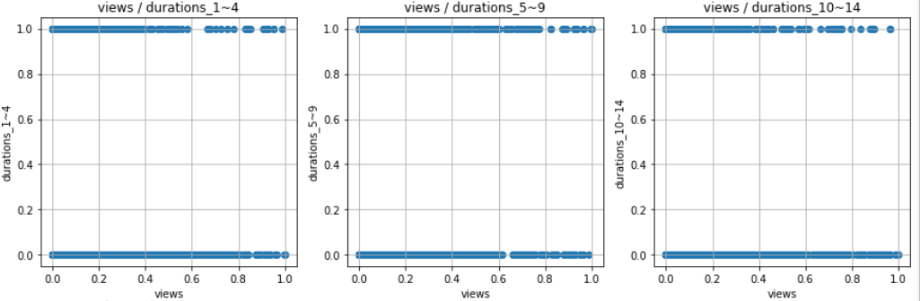
시기가 봄일 경우 다른 계절보다 높은 조회수에서 아주 약간의 높은 밀도를 보이나 그 차이가 미미하여 유의미한 결과를 가지지는 않는다고 보았다.

**-Day / Night**



Day feature의 경우 night과 비교했을 때 낮에 좀 더 높은 밀도를 보였다. 즉 밤에 영상을 올렸을 때보다 낮에 올렸을 때, 영상의 조회수가 높은 경향이 있다고 볼 수 있다.

**-Duration**



영상길이를 5분 단위로 나누어서 총 5개의 feature로 평가하였다. 영상 길이가 5~9분 일 때 조회수에서 가장 높은 밀도를 보이다가 이후 20분 이상의 feature로 가면서 밀도가 감소하는 모습을 보인다. 이를 통해 영상길이가 20분 이상으로 길어지면 조회수가 낮은 경향이 있고, 5분에서 9분의 길이의 영상이 높은 조회수를 얻는 경향이 있다고 볼 수 있다.

결론적으로 이를 정리해보면, **카테고리 번호는** **22번(People & Blogs), 24번(Entertainment)** **일 때**, **업로드 시간이 낮일 때, 영상 길이가 5~9분 일 때** 유튜브의 조회수가 높아지는 경향을 보인다. 반대로 업로드의 날짜(계절) 혹은 해당 영상의 좋아요 수와 댓글 수는 조회수와 연관성이 낮다고 해석할 수 있다.

**5. 결론(Conclusion)**

**­** **5-1. What you have done in this project**

이번 프로젝트에서는 유튜브 API에서 얻은 데이터를 통해 새로운 컨텐츠를 게시하였을 때 얻을 수 있는 조회수를 예측하는 모델을 만들고, 각 feature들간의 상관관계를 바탕으로 컨텐츠를 게시할 때 높은 조회수를 얻기 위해 작성자가 고려해야 하는 요소 및 특성들을 파악할 수 있었다. 유튜브 동영상을 게시하는 작성자는 높은 조회수를 얻기 위해서는 영상의 길이, 컨텐츠 장르, 업로드 시간 등과 같은 요소들을 신중하게 고려해야 하고, 기존의 영상 데이터들을 토대로 자신이 게시한 동영상의 조회수를 예측하기 위해서는 랜덤 포레스트 회귀모델이 가장 적합하다는 결론을 도출하였다.

**5-2. Main advantages**

프로젝트에 사용된 Dataset은 본래 구독자 수가 높았던 유튜버의 영상이나 단기간동안 유행한 컨텐츠와 같이 조회수 결정에 예상치 못한 변수를 가져오는 요소들은 전처리하는 과정에서 모두 제외하였기 때문에 단순히 유튜브 인기 동영상에 대한 분석이 아니라 새롭게 영상을 게시하거나 유튜브를 시작하려는 이들에게 충분히 유의미한 정보가 될 것이라고 예상한다.

**5-3. Limitation**

영상 제목 길이나 태그 개수와 같은 기본적인 요소들이 조회수에 큰 영향을 미칠 것이라고 예상했던 바와 달리 각 feature과 조회수와의 연관성을 찾기는 어려웠다. 이는 위에서 설명한 내용과 같이 유튜브 영상에 대한 기본적인 요소를 제외한 유튜브의 자체적인 영상 추천 알고리즘이나 구독자가 많은 유튜버의 영상을 찾게 되는 시청자들의 심리와 같은 요소들이 조회수 상승에 큰 영향을 미치고 있음을 의미한다.