Predict popularity of TED video

Eungchan Kim, JT Huang

1. Introduction

Problem

- What factors will drive the viewership of the TED YouTube videos?

Insight

- We can discover how the word of mouth promotion through social media affects the popularity of the video, in terms of viewership, using two different sources: social media (Twitter and Facebook) and YouTube platform.
- Our model can suggest the video owners how to make their videos popular.
- From the result model, we surprisingly find that:
  1) Facebook like count has a negative weight, which may mean that many Facebook users clicked the like button of the video, but did not watch.
  2) The weight of Twitter counts is way bigger than the weight of Facebook share counts, which may mean that social sharing through Twitter may be more powerful to attract audience rather than through Facebook.

Table 1. Independent variables and their coefficients in the linear regression

Attribute	FB_ like_counts	FB_ share_count	twitter_count	FB_ comment_count	duration	category
Coefficient	-41.397	27.850	152.822	41.022	1.684	248.872

Solution

- Using "linear regression" supervised learning to train and find a model and then try to predict the potential view counts according to those factors.

2. Problem

Motivation

- These TED and TEDx Talk videos are worth spread into different languages, not only their own native languages, so we want to build some tools to encourage translators to join the translation project.
- First we try to build a recommendation tool to recommend the potential popular videos for the translator to translate.
- Moreover, we are also curious about how the social media will help to promote the viewership of the videos.

Dataset

- YouTube data (28,952 videos) from 'tedxtalks' (27,581 videos) and 'TEDtalksDirector' (1,371 videos) using YouTube API
- Facebook graph data and Twitter data from each service's API according to YouTube link urls

Table 2. Collected attributes from each source

Source	Attributes
YouTube	'viewCount', 'commentCount', 'ratingCount', 'duration', 'favoriteCount', 'likeCount', ‘title’, ‘description’, ‘Category’, ‘tag’
Facebook	‘FB_like_count', 'FB_share_count', 'FB_click_count', ‘FB_comment_count'
Twitter	‘twitter_count’

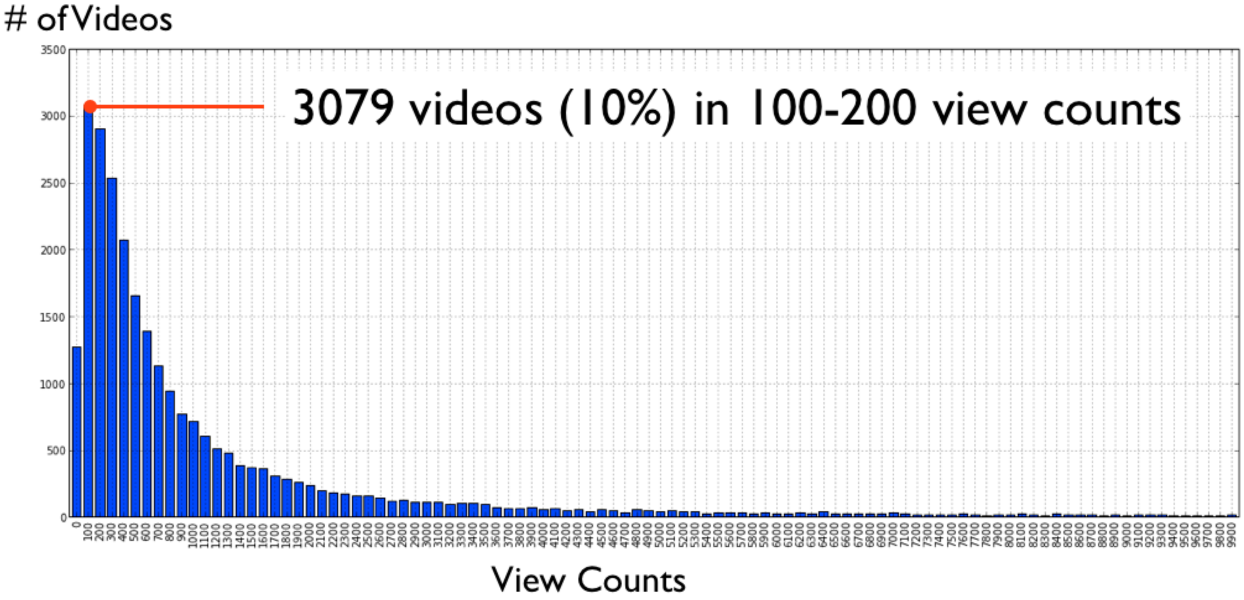

Figure 1. Distribution of view counts

Problems

- Collecting social media data from Twitter and Facebook: it takes more time than we thought to comply with the API policy (no more than 600 request in 600 seconds for Facebook APIs). Therefore, it took almost 10 hours to retrieve the all the social media data of those videos. (This is totally different from downloading the existing dataset)
- Converting nominal variable to numeric data: we converted “Category” values into numerical data.
- Data cleaning: there are some empty values from YouTube API data, we will just skip those data entries with empty values.

3. Solution

Using “linear regression” supervised learning with “scikit-learn” library.

- Adding factors one-by-one to observe the increased accuracy.

Table 3. Results from each iteration

Combination Name	Factors	Accuracy
A	**FB_like_count**	0.356182150223
B	A + **FB_share_count**	0.683559070773
C	B + **twitter_count**	0.715320465006
D	C + **FB_comment_count**	0.748929818155
E	D + **duration**	0.749118154053
F	E + **category**	0.749048360611

Failure:

- Originally we try to call the ***cross_validation.cross_val_score()*** function to automatically cross-validate through the dataset, but maybe because our dataset is not big enough, the result did not go well.
- Try to use "classification" supervised learning to classify videos as "popular" or not, but find problems to normalize future video data, and the results are not good enough.

4. Details

- After failing to use ***cross_validation.cross_val_score()***, we try to use ***cross_validation.train_test_split()*** to split the dataset into training set and testing set.
  1) **test_size**: maybe because our dataset is not big enough, we have to leave more data for training, and considering not to be "overfitting," we only use the value **0.1** (10% of the whole dataset as the testing set).
  2) **random_state**: because our dataset starts with TEDx Talk videos first and the TED Talk videos, so need the help of random sampling. Therefore we define the value as **1**.
- Try to use another "linear regression" function ***linear_model.Ridge()***, but the result is not so good as ***linear_model.LinearRegression()***, so we still use the later function.

5. Related work

- We had worked on the project of 'working with open data' (WWOD) class dealing with comparison between TED and TEDx. We just modified the codes of collecting YouTube data from the project (http://nbviewer.ipython.org/5439852).
- We reflected feedbacks from classmates after the presentation of the WWOD project: including social media data and predicting the popularity with various attributes of video.

6. Further work

- We need to increase the accuracy rate of our model, by adding other attributes like keywords from each video's title or description.
- We can expand the scope of our model to all YouTube videos.
- We can explore more data mining techniques such as decision trees, SVM, or Naive Bayes in order to classify videos.