YouTube Videos Prediction: Will this video be popular?

Kent Eng[1], Yuping Li[1], Liqian Zhang[1] | {kent3, yupingli, liqianz} @stanford.edu

[1]Civil and Environmental Engineering, Stanford

Motivation

Being a new type of job, YouTubers earn money through the advertisement and bonus from videos. Hence, the **popularity** of videos is the top priority for Youtuber.

In this project, we are trying to predict the performance of the videos that are going to be uploaded on YouTube. First, we set up an equation to manually classify all the videos into **four classes**: non-popular, overwhelming praises, overwhelming bad views, and neutral videos. **Title**, **time gap**, **category**, **tags**, **description**, and **video length** are the features of the algorithm.

SKHCYSS 10-11 3H Group3 Jet'amie

423 views • 8 years ago

2010-2011 SKHCYSS 3H-GROUP3 Jet'amie 宣傳片

0:53

Dataset

A set of trending YouTube video statistics from November 14, 2017 to June 14, 2018 was used in this project. The data in the **last trending date** was used for each video. The data set original contains trending date, title, category id, publish time, tags, views, likes, dislikes, comment count and description. We used the last trending date and publish time to get the time gap and utilized API to get video length. Since some videos became private or got deleted, it would have zero seconds for the video length. These videos were eliminated in our dataset. The rest of the videos were split into 3 sets: **70% were train set**, **20% were valid set**, **10% were test set**. Finally, all videos were label by the following equation:

$$\text{Score} = \frac{Comments}{Views} * (Likes - 1.5 * Dislikes)$$

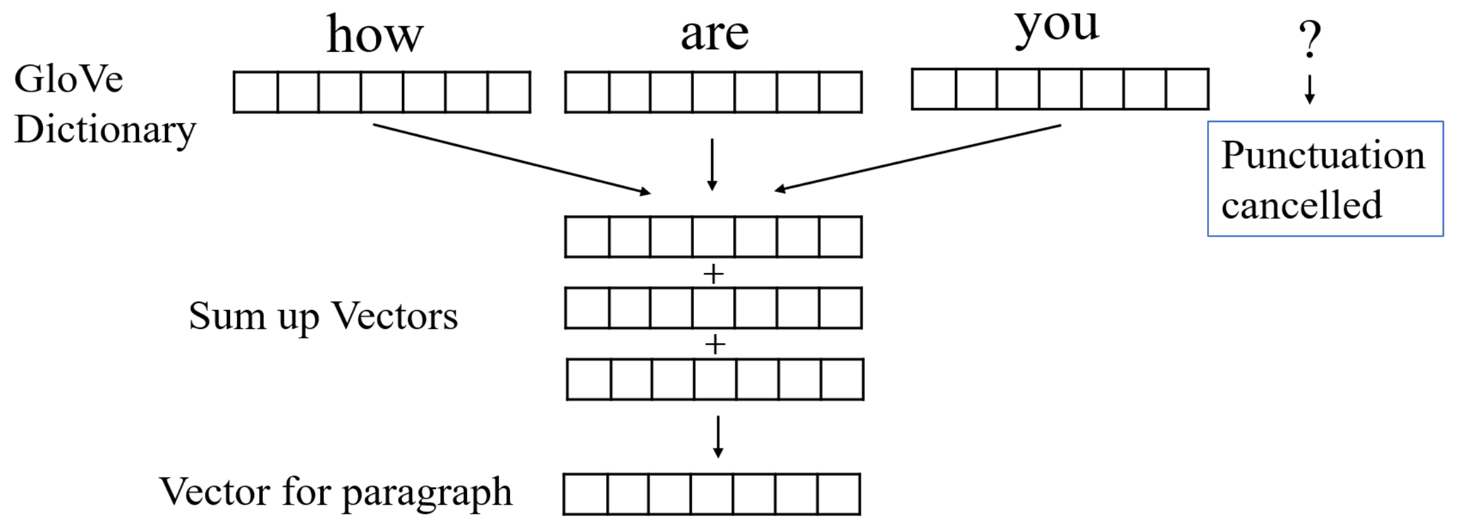
$$y = \begin{cases} 0, & Views < 100,000 \\ 1, & Views \geq 100,000 \quad Score < 0 \\ 2, & Views \geq 100,000 \quad 0 \leq Score < 300 \\ 3, & Views \geq 100,000 \quad Score \geq 300 \end{cases}$$

Model

Feature engineering

- Utilized the title, description, tags, category, time gap, and duration of each video as original features.
- Used GloVe for word embedding(vector size: 25)
- After embedding, combined all the digitalized data as features of each video.

GloVe Word Embedding:

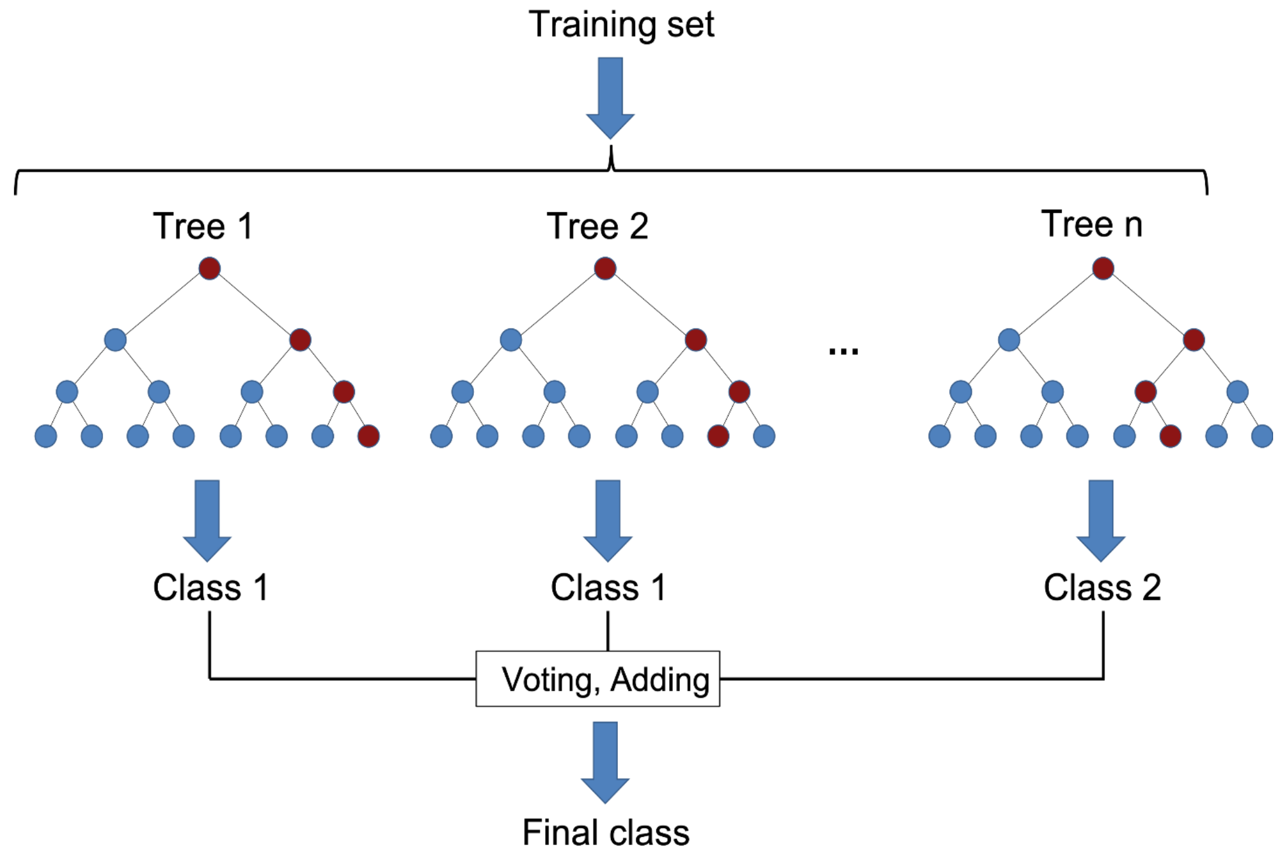


Main Multiclassifiers

- Softmax function for **XGBoost**

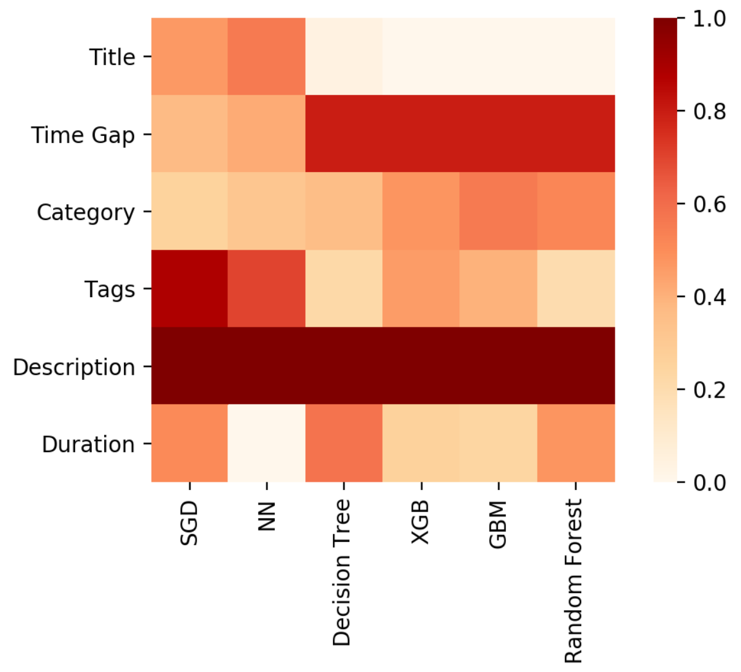
$$\ell(\theta) = \sum_{i=1}^{n} \log \prod_{l=1}^{k} \left(\frac{e^{\theta_l^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \right)^{1\{y^{(i)}=l\}}$$

- Framework of **Gradient Boosting Decision Tree** and **Random Forest**



Other Models

- **SGDClassifier** (loss function: modified huber):
- **Neuron Network** (solver: adam; activate function: logistic; hidden layer: 100, 2)
- **3-Level Binary Decision-Tree Framework**: One class is picked out at each layer, using multiple binary classifiers, such as random decision forests, gradient boosting method, neural network, etc.
- **Backward search** on features and drew the remained frequency matrix for each method to reduce features.

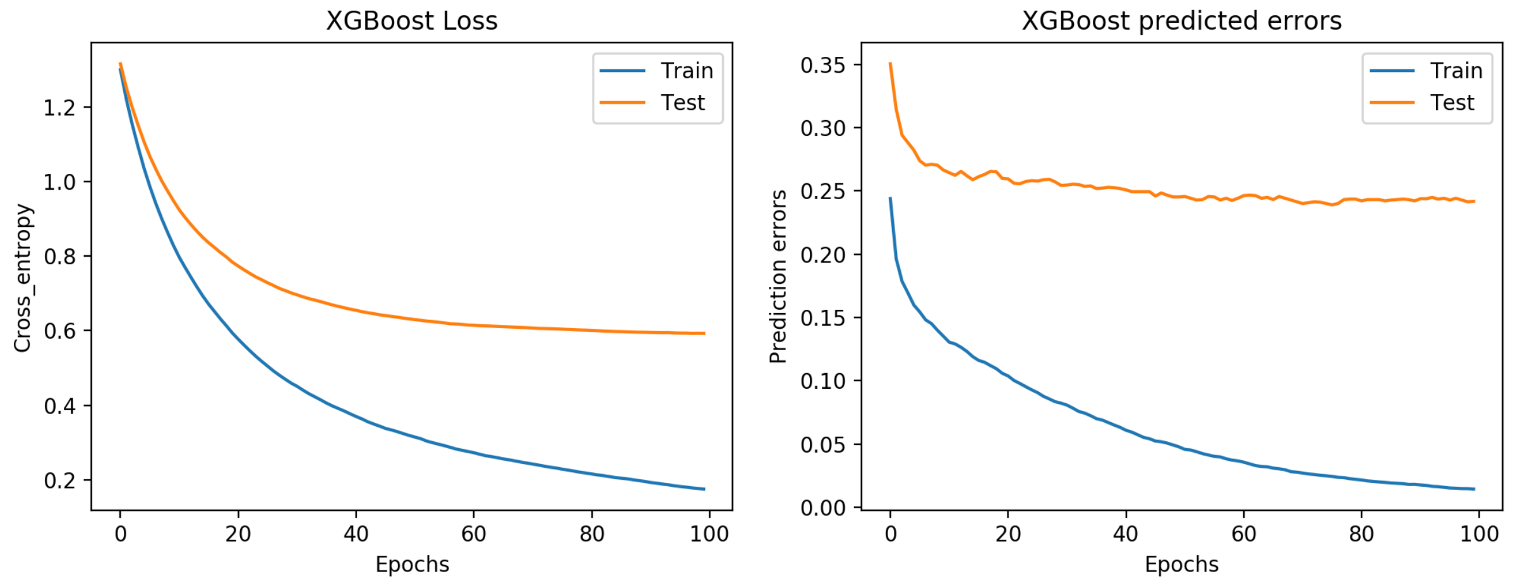


Results

- **F1 score of models**

Model	F1 score	F1 score (Binary framework)
Uniform prediction (Base line)	0.303	/
SGDC	0.552	0.557
SVM	Not Converge	Not Converge
Neuron Network (MLPC)	0.682	0.679
Decision Tree	0.646	/
Random Forest	0.698	0.723
XGBoost (6 features)	0.741	0.706
XGBoost (3 features)	0.736	/
GBM (6 features)	0.747	0.717
GBM (3 features)	0.727	/

- **Loss and error rate of XGBoost over epochs**



Discussion

After comparing the result (f1 score) of all the models, the boosting method is the best fit (GBM: 0.747; XGB: 0.741). By backward search on feature, description, category, and time gap are the most important features. The f1 scores of using these 3 features are 0.727 for GBM and 0.736 for XGB.

However, after considering the time cost, the extreme gradient boosting method with 3 selected features is chosen as the best model.

There are two major issues in our algorithm:

1. Over fitting.
2. Imbalanced data (Class 2 overwhelming).

These issues should be explored and resolved in future work.

Future work

For future work, the video itself (series of images) and subtitles can be used as additional features to consider the content of videos. Some features may also be replaced with other features to resolve the problem of overfitting. Besides that, the videos' data of a specific class can be collected from the YouTube Data API directly to increase the data size and resolve the problem of data imbalance. Furthermore, the convolutional neural network can also be used as a model to process the features that contain images.

Reference

[1] Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation.

[2] https://xgboost.readthedocs.io/en/latest/

[3].https://www.kaggle.com/datasnaek/youtube-new