Analysis of Trending YouTube Video Statistics

Jiong Liu
A53269256
jil027@eng.ucsd.edu

Sihan Wang
A53271463
siw003@eng.ucsd.edu

I. Introduction

This project is our final project of ECE225, which we will accomplish a predictive task based on the YouTube Video dataset on Kaggle. This dataset contains more than 200000 data points and around 20 features for each sample. YouTube (the world-famous video sharing website) maintains a list of the top trending videos on the platform. According to Variety magazine, “To determine the year’s top-trending videos, YouTube uses a combination of factors including measuring users interactions (number of views, shares, comments and likes).Note that they’re not the most-viewed videos overall for the calendar year”. YouTube Video is a really hot topic in predictive area and it’s also a really interesting topic for us so we choose it for our final project. The report will divided into four parts: dataset & analysis, predictive task, model, and finally, result & conclusion.

II. Dataset & Analysis

A. Dataset

In this project, we used “Trending YouTube Video Statistics” [1] from Kaggle. Details and contents of the dataset is shown in “Table. I”. This dataset is a daily record of the top trending YouTube videos. This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, and FR regions (USA, Great Britain, Germany, Canada, and France, respectively), with up to 200 listed trending videos per day.

TABLE I
Details & Contents of Dataset

Column Name	Description
video_id	the id of this video
title	the title of this video
channel_title	the title of this channel
category_id	the id of this category
publish_time	the time of the video published
tags	the tags belonging to this video
views	the number of views of this video
likes	the number of people like this video
dislikes	the number of people dislike this video
comment_count	the number of the comment
thumbnail_link	the link of this video’s thumbnail
comments_disabled	True or False
ratings_disabled	True or False
video_error_or_removed	True or False
description	the description of the video

B. Analysis

1) Basic Statistics: Basic statistics of YouTube Trending Videos in 10 countries is shown in “Fig. 1” and “Fig. 2”.

- Ratio of YouTube Trending Videos
  The statistics about ratio of YouTube Trending Videos in 10 countries is shown in “Fig. 1”. After combining statistics from multiple countries, it’s good to see the ratio of videos we have in different countries. In this plot we keep only last entry for duplicated videos. That’s why we can clearly observe that the country with the most long-trending video end up having lesser videos.

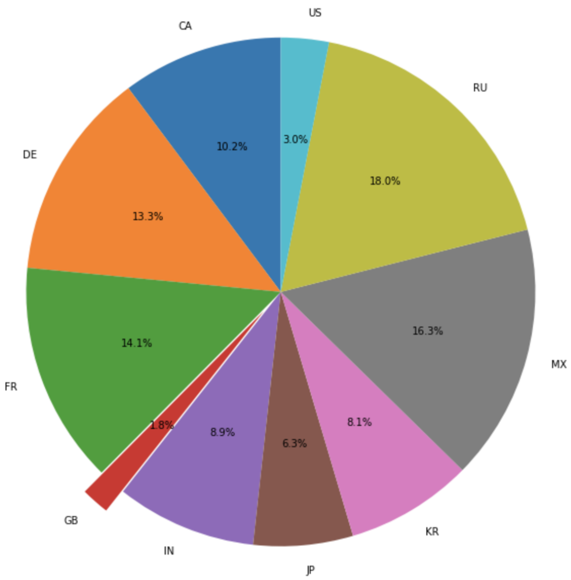

Fig. 1. Ratio of YouTube Trending Videos in 10 Countries

- likes, dislikes, views and comments
  The statistics about the number of likes, dislikes, views and comments got by different countries is shown in “Fig. 2”. Obviously, four of the graphs share the similar trend in shape and numbers. One possible reason we can think about and that may cause this situation is the video’s trending duration. Enduring trending videos have much more advantages in getting more views, likes, dislikes and comments.

2) Analysis around category: In this part, we want to analyze the effect of different category in these 10 countries. Which category is the most popular according to users from

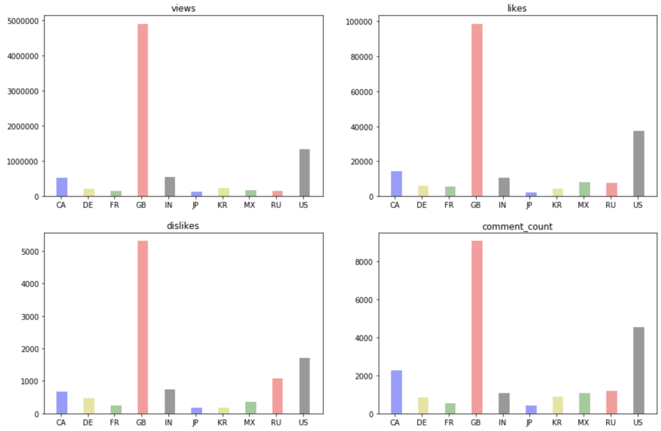

Fig. 2. Counts of Likes, Dislikes, Views and Comments Compared with Different Countries

different countries? We select four countries(Canada, Germany, US and UK) from the dataset to show the results.

From the figures shown below, we can see that the top category of these countries is Entertainment. Music's videos ranked insignificantly in Canada and Germany compare to US and UK. Top 5 categories in United Kingdom are entertainment-related, which is a quite interesting observation compared to the results in other countries and might not be a good sign.

3) Sentiment Analysis on Video's tags : There are so many categories in these countries. We use Entertainment and News and Politics category to show the result we got in "Fig. 7" and "Fig. 8".

We can see in different categories, the polarity of category is different shown in "Fig. 9". Some are negative(News and Politics), and others are positive(Entertainment), which may reflect to people's concerns in different areas.

III. Prediction Task

A. Task

After we have fully analyzed the data in the dataset we chose, we decided to make predicting the category based on a video's title, tags and description as our task, specifically for videos in USA. The reasons we only choose videos in USA are, in other languages, coding format may not be UTF-8, and they will need multiple corpora, which will cause conflicts. Our model dealing with datasets from different countries is the same.

This is a supervised multi-label classification problem. Instead of using features directly from data, for example an image classification problem, we have to firstly extract features from collections of words, then apply the new featured data into a machine learning model to solve the classification problem. Knowledge from text mining and machine learning will be helpful.

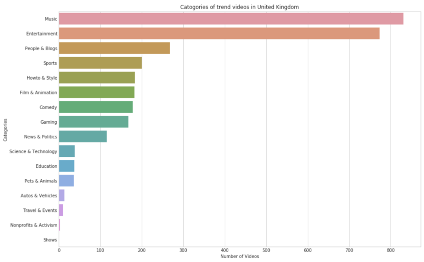

Fig. 3. Popular Categories of United Kingdom

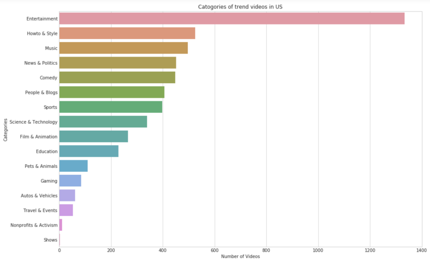

Fig. 4. Popular Categories of the Unite States

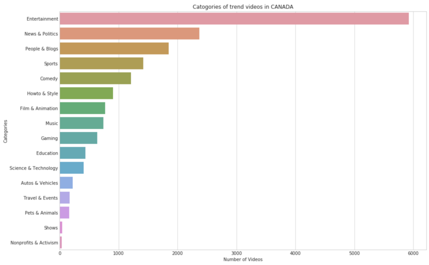

Fig. 5. Popular Categories of Canada

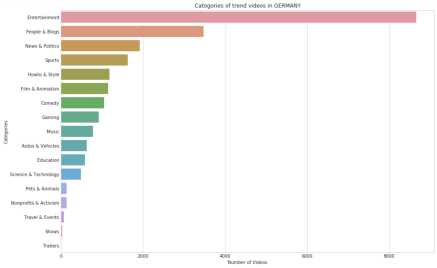

Fig. 6. Popular Categories of Germany

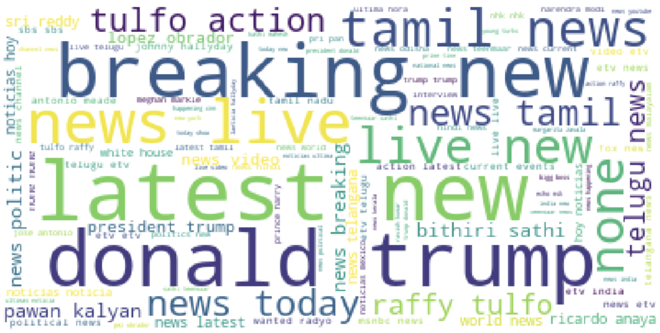

Fig. 7. Most Discussed Word in News and Politics Category

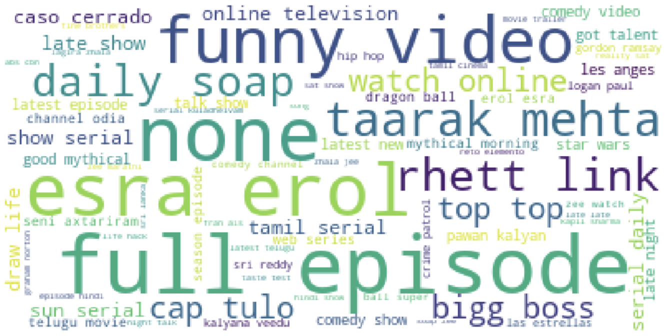

Fig. 8. Most Discussed Word in Entertainment Category

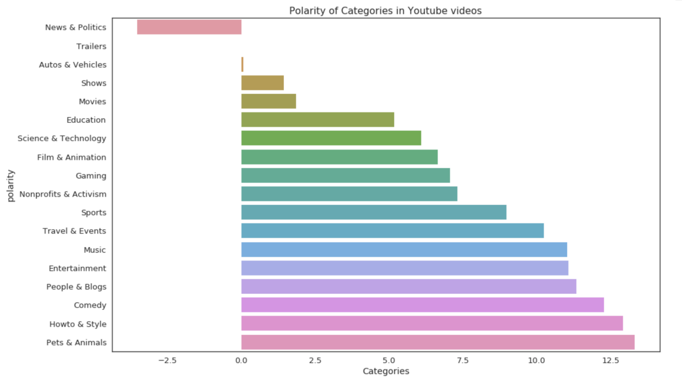

Fig. 9. Polarity of Each Category in YouTube Videos

B. *Prediction Assessment*

For the metric to assess our prediction on the dataset, we choose to use the precise accuracy and F1 score to evaluate our model. For a multi-label classification task, accuracy shows how many predicted labels are exactly the same as its assigned label.

The F1 score can be interpreted as a weighted average of the precision and recall:

$$F1 = \frac{2(precision * recall)}{precision + recall} \tag{1}$$

where:

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{2}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{3}$$

An F1 score reaches its best value at 1 and worst score at 0. In the multi-class and multi-label case, this is the average of the F1 score of each class with weighting depending on the average parameter.

IV. Our model

In order to do the predictive task we mentioned above, we split it into following two steps: data pre-processing and classifier training.

A. *Data Pre-processing*

1) Data Cleaning: To predict categories, some of the columns in our dataset is meaningless, for example “publish_time” and “comment_count”. We chose following columns:

- title: the title of this video.
- channel_title: the title of this channel.
- tags: the tags belonging to this video.
- description: the description of the video.

Our data records trending videos every week, therefore it is possible that a video exists in multiple records and we have to de-duplicate data. Since we are extracting text from records, it is necessary to fill null values with words like “missing”. Besides, to reduce confusion, transforming all words to lower case and replacing meaningless symbols are necessary.

2) Feature Extraction: We applied Latent Dirichlet allocation (LDA) model [2] to extract topics from texts. In natural language processing, latent Dirichlet allocation is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word’s presence is attributable to one of the document’s topics [3].

According to our analysis part, there are 16 categories among all trending videos in USA. Separately extracting features from title, channel title, tags and description brings us 64 new feature columns. This can be done by firstly extract vectors form texts by *sklearn.CountVectorizer*, then fit output matrices into *sklearn.LatentDirichletAllocation*.

Finally, we dropped the original four unencoded feature columns. We obtained a featured dataset with 65 features. The one with “category_id” is used as our assigned label to each video.

B. *Classifier Training*

We used Random Forest model as our classifier. Random forests [4] are an ensemble learning method for predictive tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is mean prediction of the individual trees [5]. Generally, random forests has good performance on kaggle competition.

In random forests, the input dataset D which consists of N d-dimension samples is firstly considered as the root of a classification tree. It does a loop to find the best features. Gini impurity is chosen as the metric to split the root which measures the probability of data:

$$Gini(f) = \sum_{i=1} J f_i(1 - f_i), i = 1, 2, ..J \tag{4}$$

where J is the number of classes and f_i is the portion of data belongs to it. While the Information gain is calculated based on the entropy:

$$Entropy = -\sum_{i=1} J p_i \log_2(p_i), i = 1, 2, ..J \tag{5}$$

where p_i is the probability of class i and the sum of p_i should be added up to 1. The information gain is the entropy difference between node and its children:

$$IG = Entropy(T) - Entropy(T|a) \tag{6}$$

The data set keeps splitting based on one of these two metrics until all nodes could not be separated.

We splited the whole dataset into 80% samples of training set and 20% samples of testing set. The best outcome comes with a total number of 2500 estimators.

C. Summary

Following chart shows our pipelines. Implementation details is attached in our Python notebook.

```
#data processing
from sklearn.model_selection import
    train_test_split
#evaluation metrics
from sklearn.metrics import accuracy_score,
    f1_score
#feature extraction
from sklearn.feature_extraction.text import
    CountVectorizer
from sklearn.decomposition import
    LatentDirichletAllocation
#classifier
from sklearn.ensemble import
    RandomForestClassifier
#stop words in vectorizer
from nltk.corpus import stopwords

#loading data
US_data = pd.read_csv('USvideos.csv')

#data cleaning
data = preProcessing(US_data)

#feature extraction
vectorizer = CountVectorizer()
token_counts = vectorizer.fit_transform(data)
lda = LatentDirichletAllocation()
lda.fit(token_counts)

#rebuild dataset
col_names = ["Desc. Topic {0}".format(x) for
    x in range(0, n_topics)]
topic_dist = lda.transform(token_counts)
topic_df = pd.DataFrame(topic_dist, columns =
    col_names)
data_description_featured = pd.concat([data,
    topic_df], axis=1)

#training
pipeline = make_pipeline(ce.OrdinalEncoder(),
    RandomForestClassifier())
pipeline.fit(X_train, y_train)

#predicting and evaluation
y_pred = pipeline.predict(X_test)
accuracy_score(y_test, y_pred)
f1_score(y_test, y_pred, average='weighted')
```

V. Result & Conclusion

A. Result

The best outcome comes with a total number of 2500 estimators. The accuracy of our predicts is 55.118%, and F1 score is 0.53520. Comparing with the accuracy that directly assign the most popular category to a video, which is 25.473%, our result is not bad. "Fig. 10" shows the confusion matrix of our best predicts.

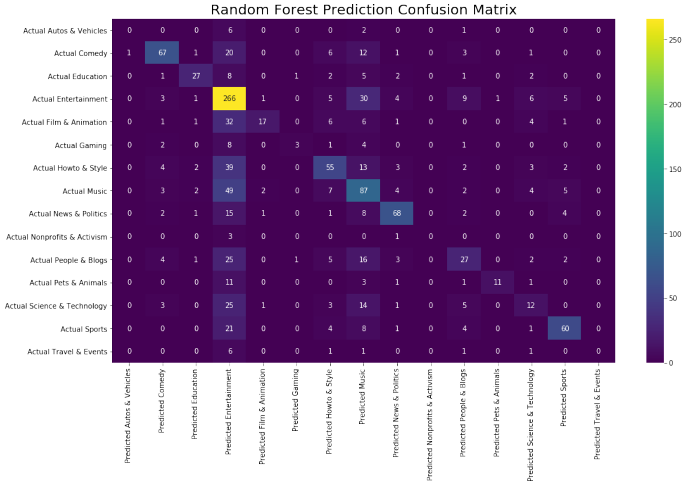

Fig. 10. Confusion matrix of our best prediction

B. Conclusion

We get our dataset from kaggle and trending Youtube videos is a really popular exploratory data analysis topic in kaggle. However, most of the tasks only focus on exploratory data analysis, rather than machine learning predicting problems. Predicting categories based on texts is a typical multi-label classification problem and there are many features which could be extracted from texts that will have effect on the prediction of category.

Though our best result is significantly better than naive approaches, the accuracy is still relatively low. It could due to multiple reasons, such as the dataset is not big enough, our feature extraction method could be better, and we could have tried different machine learning or deep learning models, which could be listed as our future work.

References

[1] Michael J. Trending youtube video statistics. https://www.kaggle.com/datasnaek/youtube-new.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] Wikipedia contributors. Latent dirichlet allocation — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Latent_Dirichlet_allocation&oldid=927159179, 2019. [Online; accessed 9-December-2019].

[4] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[5] Wikipedia contributors. Random forest — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Random_forest&oldid=925199071, 2019. [Online; accessed 9-December-2019].