ORIE 4741 Final Report: Classifying Youtube Videos

Kyle Wadell , Brandon Wang

December 2019

Contents

Abstract

In this report, we build a model to automatically identify the category of a Youtube video. We begin by training a set of models with general video statistics captured from the trending page over the past three months. We were not able to build an accurate classifier with just these features, so we used a natural language processor to create a second set of models which predicted the category id using the word embeddings generated from the videos title and description. These performed much better then the original set of classifiers. Finally we built a set of ensemble models which combined the predictions from both versions to create a prediction based off of both the videos performance statistics and the word embeddings. However, the ensemble models never significantly improved over the models focused purely on natural language processing, and we instead concluded that the best approach was a logistic classifier model using multinomial loss and trained on a subset of the possible word embedding.

1 Problem Description

On April 23, 2005, the first ever YouTube video was uploaded. Since then, YouTube has grown tremendously in popularity and new videos on all kinds of topics are being uploaded every day. YouTube has become the largest search engine for videos and each minute, more than 500 hours of video content is uploaded to YouTube. With over two billion people(about one-third of all people on the internet) searching for videos, it is extremely important to be able to categorize videos effectively for many different reasons.

As of today, YouTube currently has 31 different categories including categories such as gaming, comedy, people & blogs, and how to & style. YouTube viewers can use these categories to find videos relating to their interests and discover new videos. Content creators can also see what other kinds of videos similar YouTubers are creating in order to find opportunities to collaborate or research the competition. One of the most important reasons for YouTube to correctly identify categories for videos in to assist in selling ad space for effectively. For example, a credit card business would want its YouTube ads to target the right demographic of people who might want to apply for a new credit card. Putting their ads on YouTube videos about personal finance would be much more effective then having their ads on a YouTube prank channel's videos. Thus, finding the right category of videos to purchase ad space on could make a huge difference in a company's sales or conversion rates.

One thing to keep in mind is that video uploaders are actually the ones to choose which category their video belongs in on YouTube. Our goal was to create a better categorization of the videos on YouTube instead of relying on the uploader to pick a category in order to provide better ways to sell ad space on YouTube videos.

2 Data Set and Preprocessing

We used the Kaggle "Trending YouTube Video Statistics" data set for our report, which contains several months worth of data scraped from the trending page on YouTube. The original data set compiled the results of this scraper across several geographic regions: USA, Great Britain, Germany, Canada, and France. For our project we chose to focus specifically on the US data set which contains just over forty thousand unique videos, divided into 31 possible categories.

Metadata
Date Uploaded
Likes
Dislikes
Comment Count
Views

Textual Data
Title
Description
Tags
Channel Name

The original data set contains features which can be divided into three classes. First there is the category id, which serves as the output variable for our first set of models. Next there is a set of features comprising the video's "metadata," these are all the general summary statistics detailing a videos performance, such as amount of likes or comments. Finally, we have textual features which include strings describing a video's content.

3 Naive Models

Initially, we attempted to classify the videos into their respective categories using their performance metrics. This only included data from the number of views, likes/dislikes, and number of comments. Each video can belong in only one category which is a multiclassification problem. We started by having the data set split into a training and testing set. We fit two logistic models, one with a Multinomial Logit loss function and one with a one vs all loss function. Then we also fit a decision tree and a Gaussian Naive Bayes model.

3.1 Multinomial Logit

The Multinomial Logit loss is the most commonly used loss function for multiclassification problems which was why we chose this loss function to start with. The Multinomial Logit is a classification method we used that generalizes logistic regression for situations in which there may be more than two possible outcomes. The Multinomial Logit loss can be defined as:

$$MNL = \sum_{i}^{C} t_i log(s_i)$$

Where C is the number of classes, t_i is the ground truth and s_i inside the log is the actual predicted probability for the ground truth class. One thing to keep in mind is that the Multinomial Logit loss function has a heavy weight on the penalization for confident predicted probabilities that are wrong. The Multinomial Logit loss function resulted in approximately a 14.4% accuracy rate for predicting video categories.

3.2 One Vs All Hinge

Then we also used the one vs all hinge loss function to predict categories of YouTube videos. This method essentially trains a collection of binary classifiers using them as a subroutine. In a way, this almost ignores the multiclassification problem and reduces it down to binary classification problems. This hinge loss function can be defined as:

$$l(y) = \sum_{y \neq t} max(0, 1 + w_y * x - w_t * x)$$

Where t is the target label and w_t and w_y are the model parameters. The one vs all hinge loss function resulted in approximately a 14.4% accuracy rate for predicting video categories which was identical to the prediction from the Multinomial Logit loss function. This was because both models predicted that each video would fall into the music category based solely off the video's "metadata". This most likely happened because there was a larger number of music videos in the trending page on YouTube relative to all the other categories.

3.3 Decision Tree

Looking for accuracy improvements, we used a decision tree to predict video categories next. This allowed us to go about the multiclassification problem very systematically by partitioning our data according to different characteristics. In this method, we start by posing some questions to the data set based off its features(likes/dislikes, views, etc.) and then split into different groups that can then be described by different characteristics. We used Scikit-learn to train our decision tree and the predictions for YouTube video categories were around 60% accurate.

3.4 Gaussian Naive Bayes

The last method we tried on the "metadata" was Gaussian Naive Bayes classification. This method is Naive Bayes but assumes a Gaussian distribution so that it works with real-valued attributes. Unfortunately, this model's prediction results were the worst with an accuracy under 10%.

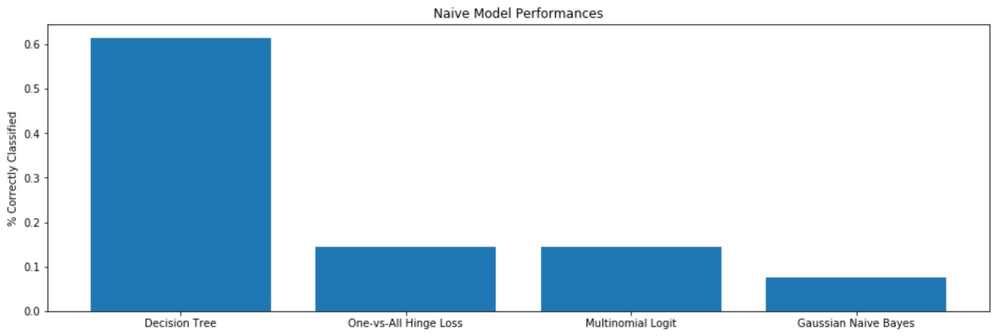

Figure 1: Cross-Validated Accuracey for each of the metadata focused models.

4 Encorporating Embeddings

4.1 Word Embeddings

As a result of the poor performance of our metadata models, we decided to incorporate the textual features our classification algorithm. For this, we used natural language processing to convert each of these features into our word embedding. This process aims to represent each segment of text as a high dimensional vector, such that two words with similar meanings will have a shorter euclidean distance between them then two words with opposite, or unrelated meanings. A demonstration of this process is shown below, on a two dimensional projection of the word embeddings assigned to each of the category titles. This figure was built using principal component analysis to reduce the 512 dimensions present in an embedding into a two dimensional plot. It only depicts an approximation of the actual embedding vectors but it does demonstrate how similar concepts, such as "Education" and "Science/Technology", are placed in similar euclidean positions.

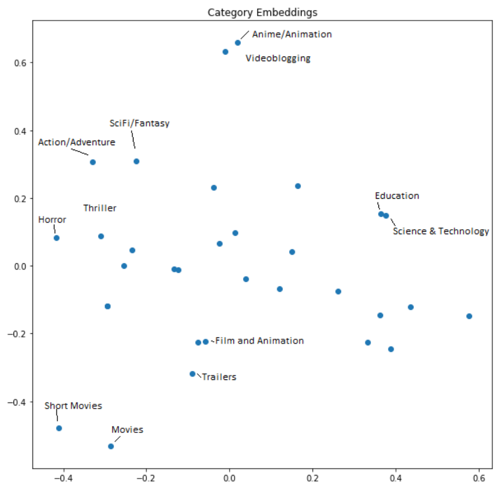

Figure 2: PCA plot of the word embeddings generated from category titles

After this process, our dataset now contained 1025 features for each video, one for the category id and then two sets of 512-length vectors containing the word embeddings for the title and the tags. The description feature was discarded because it contained too many entries which were either impossible or far to computationally expensive to create embeddings for, such as descriptions which contained extensive lists of hyperlinks and emojis.

4.2 Model Fitting/Validation

Although our data set is large, we also now have a large amount of a features for each observation. As a result, we chose to not seperate out any of the data into a test set and instead performed 5-fold cross validations on each of the models used with word embeddings vectors. We trained three model classes on the word embeddings data set, a logistic regression classifier with multinomial loss, a decision tree classifier, and a random forest classifier. It should be noted that k-fold cross validation tends to produce lower estimates of error then those achieved from splitting the data set into the test and training sets. As a result we also fit all of these models a second time using a test and training set, in order to compare them to the classifications built in the previous section. While many of the following models have low CV error estimates, they all performed higher then the best of the naive models when using a test and training data set.

5-Fold Cross Validation Results

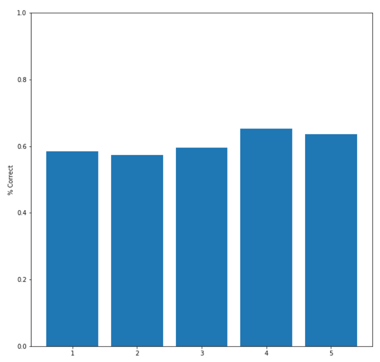

Figure 3: Multinomial Logistic Regression

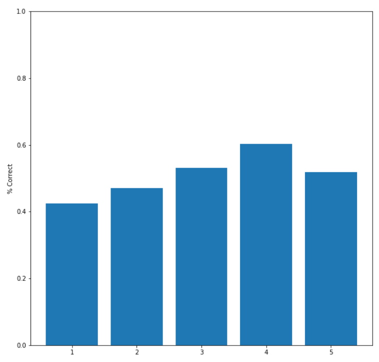

Figure 4: Decision Tree

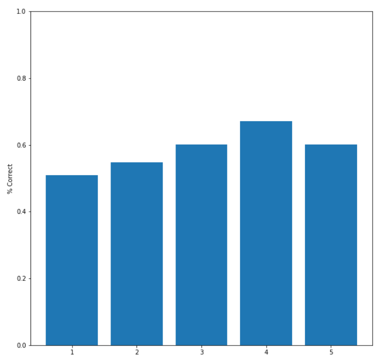

Figure 5: Random Forest

5 Ensemble Models

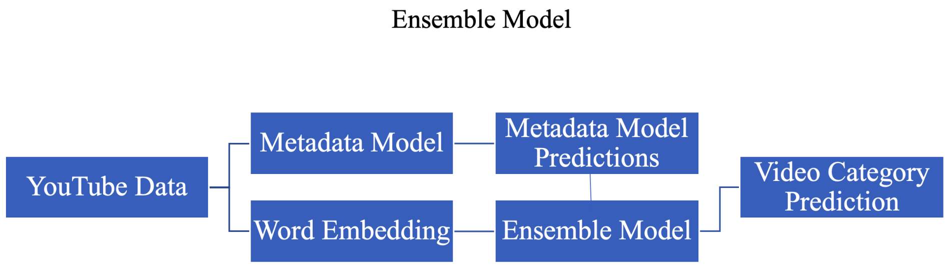

Figure 6: Ensemble Model Visualization

Finally, we built a selection of ensemble models which combined the results of the metadata models and the natural language processing models. For each of these models, we added the predictions from a multinomial logistic classification model, ran on the metadata, alongside the word embeddings built from the textual features. This additional feature was tested with the same three model classes used in the previous section: logistic regression with multinomial loss, decision trees, and random forests.

5-Fold Cross Validation Results, Ensemble Models

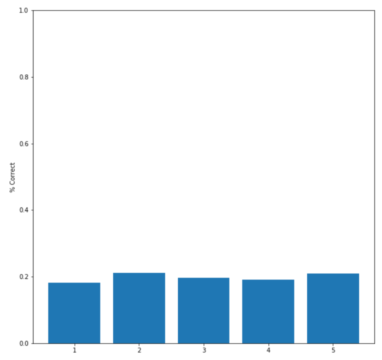

Figure 7: Multinomial Logistic Regression

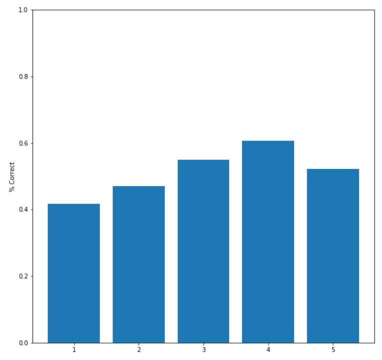

Figure 8: Decision Tree

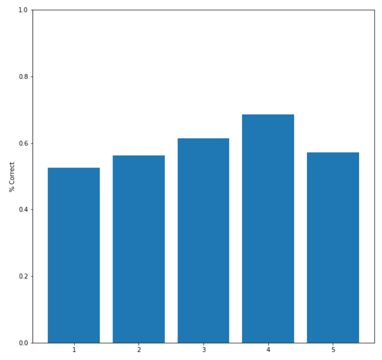

Figure 9: Random Forest

5.1 Final Model Selection

After building the ensemble models, we compared the average accuracy rates of each of these models across each of the five cross validation tests. With these results, we concluded that the metadata was not a signifigant factor in determining video category as its inclusion in the ensemble models tended to either worsen, or simply not change, the original textual models accuracey. Instead we found that a random forest model, trained on the embeddings generated from the tags and titles of videos was able to produce the most accurate results, with an average cross-validated error of 61%. Its worth mentioning that this accuracy is far more signifigant then it would have been in a binary classification scenario, where a random model could be expected to achieve 50% accuracy. When dealing with the 31 possible Youtube categories, a random model would only be expected to achieve 3% accuracy.

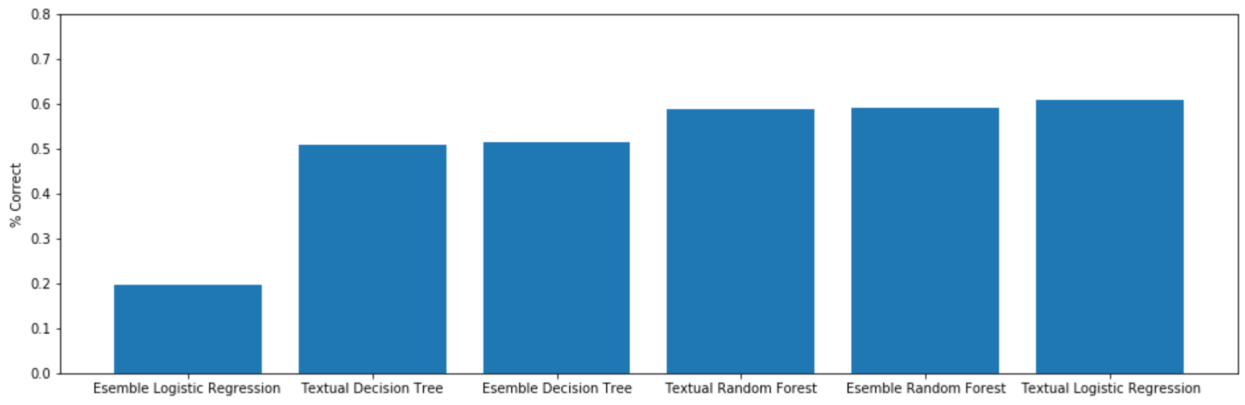

Figure 10: The average CV-Error for all six model classes. This compares three textual models, which only use word embeddings, and three ensemble models, which combine the word embeddings with the predictions of a logistic classifier trained with metadata variables.

6 Project Reflection

The way YouTube makes money is by keeping viewers on its site and watching videos. Their most important performance metric is total watch time. More time watching videos means more time watching advertisements and more money YouTube makes. This is where YouTube's recommendation algorithm comes into play. Automatically playing the next video in the recommendation list, viewers are drawn into more content similar to their interests. Google Brain, the group of engineers at Google responsible for the recommendation algorithm, started making recommendations three years ago. In that time, watch time on YouTube has grown twenty times over and seven out of ten videos people watch are videos suggested by the recommendation algorithm. Although recommending similar videos to watch is seemingly innocent, YouTube promotes controversial videos that fuel the spread of fake news and disinformation feeding directly to the large majority of the population. For example, YouTube's recommendation algorithm may have played a strong role in the spread of disinformation during the 2016 presidential election.

In regards to this, our project does not yield a Weapon of Math Destruction(WMD). Our classification of videos could potentially be a much safer way to group similar videos together and give viewers a better alternative to the recommendation algorithm to find similar videos to watch. The predictions of our model are easily measurable by watching the videos and do not have the negative consequences of the recommendation algorithm since they are based off the actual content of the videos and not the previous history in watching habits of a user. The training and testing sets for the model were also rigorously separated with results that agree with experimental measurements. Our classification model also does not discriminate against any group or individual based on race, sex, color, etc. It will fairly predict a video's category by its description and content data. This is completely independent of a user's personal information. In this way, testing for fairness is not very important for our project.

With these results, we may not see YouTube implement them in production to influence their recommendation algorithm since their current structure is very profitable. On the other hand, our classification algorithm could be put into production in marketing effective ad space to advertisement agencies. Companies would see more returns on investing into advertisements in YouTube videos with more accurate video classifications.

References

[1] ORIE 4741: Learning with Big Messy Data,
`https://people.orie.cornell.edu/mru8/orie4741/`.

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei. *Large-scale Video Classification with Convolutional Neural Networks.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725-1732

[3] Madden Amy , Ruthven Ian , McMenemy David. *A classification scheme for content analyses of YouTube video comments.* Journal of Documentation, Vol. 69 No. 5, pp. 693-714.

[4] Jose San Pedro, Stefan Siersdorfer, and Mark Sanderson. *Content redundancy in YouTube and its application to video tagging.* ACM Trans. Inf. Syst., New York, NY, July 2011

[5] Addair, T. G.. *Deep Learning YouTube Video Tags.* Stanford University

[6] Kaggle: Trending YouTube Video Statistics,
`https://www.kaggle.com/datasnaek/youtube-new`

[7] NYT: Algorithms Won't Fix What's Wrong With YouTube,
`https://www.nytimes.com/2019/06/14/opinion/youtube-algorithm.html`

[8] Towards Data Science: Common Loss Functions in Machine Learning,
`https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23`

[9] Medium: Loss Functions and Optimization Algorithms. Demystified.,
`https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c`

[10] Machine Learning Mastery: How to Choose Loss Functions When Training Deep Learning Neural Networks,
`https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/`

[11] The Guardian: Fiction is outperforming reality': how YouTube's algorithm distorts truth,
`https://www.theguardian.com/technology/2018/feb/02/how-youtubes-algorithm-distorts-truth`