COS 424 Final Project: YouTube Data

Jay Li
Computer Science, BSE
jiayangl@princeton.edu

Theodor Marcu
Computer Science, AB
tmarcu@princeton.edu

Katherine Xiao
ORFE, BSE
kx2@princeton.edu

Abstract

YouTube is the world's largest video sharing platform and one billion hours of video are watched every day on the website [1]. As a result, it is crucial for content creators to better understand textual and visual metadata factors that make trending content popular and engages viewers. Our project aims to predict user engagement, which we define as the ratio of likes and dislikes to number of views, of trending videos. We do so by creating a bag-of-words model for the descriptions, titles, tags, and thumbnail features for the videos and then running Elastic Net and LDA to find correlation between predicted and true engagement as well as inherent latent structures. In our analysis, we noticed an improvement in the performance of predicting user engagement using Elastic Net within categories and sub categories. For example, running Elastic Net on all categories yielded an R2 score of 0.455, which increased to 0.613 when running Elastic Net on subcategories identified with LDA.

1 Motivation

YouTube's core mission is to "give everyone a voice and show them the world." Many content creators, such as bloggers, entertainers, and video game commentators, have used their voices to create popular videos and built their careers on the platform. As creators earn their income through the number of clicks and views of ads before and during their videos, they are highly motivated to have more views and likes such that the videos will be well ranked by YouTube's algorithm, and in turn, gather more viewers. In our research, we take a Kaggle data set that records trending YouTube videos over a seven-month timeframe and aim to predict the user engagement for these trending videos based on the information that a video has at the time of publishing. We define engagement as the ratio of the total number of likes and dislikes to the number of views for a video. The results of such research can inform creators on what tags and textual descriptions to include in their videos when publishing, which would in turn increase their user engagement and help further their online careers.

2 Related Work

Our project builds on top of previous work focused on predicting the popularity of videos. For example, Kong et al. created HIPie, an "interactive visualization system" that allows users to easily analyze the virality and popularity of online videos [7]. Another example is the work done by Wang et al. on creating a model that can be used by influencers to "predict the number of views on their videos" [16]. This work uses supervised and unsupervised learning in order to draw conclusions about the data while developing viable tools for helping YouTube creators better tag their videos.

HIPie analyzes the metadata of past YouTube videos and employs the Hawkes Intensity Process (HIP) in order to explain and predict video popularity [7]. Kong et al. quantify the popularity of a video using the number of view counts. HIPie has three main features: allows users to "easily examine the popularity over time for online videos and forecast their future popularity," helps creators

"quantify virality and simulate video reaction to online promotions," and enables users to "compare and select content on the fly" [7]. The HIP process relies on both external inputs (social media shares) and internal inputs (YouTube platform) in order to model **popularity**.

The paper written by Kong et al. highlights how HIP is used within HIPie in order to create an **endo-exo map** used for explaining and predicting video popularity, comparing videos & channels, identify potentially viral videos, and simulate video responses to submissions. While HIPie is not a machine-learning model, the paper by Kong et al. highlights how machine learning models could be easily transformed into tools that allow content creators and users to predict the virality of YouTube videos.

In their article, Wang et al. explain how they aimed to build a model that could be used by content creators to predict the number of views for their upcoming videos [16]. Wang et al. used YouTube's 8M data set, and removed all the genres that are not related to the "Fitness and Gym" category. Using this data set, Wang et al. scraped a data set that includes the metadata for over 115,000 videos. In terms of feature selection, the authors chose to focus on the videos' *title* and **thumbnail image**, since they are the first thing a user sees on the platform.

Wang et al. used pre-trained models in order to extract meaningful features from the title and thumbnail. For the video title, they used a "clickbait scorer" that relies on deep learning methods [14]. For the thumbnail, they used Yahoo's "NSFW Scorer" that produced a NSFW score between 0.0 and 1.0 for the thumbnail of each video. In their article, Wang et al. found that the thumbnail and title had a marginal influence on the success of a video. Instead, their results showed that "future performance is based on past success," which essentially means that good content creators can expect more views for their future videos. Wang et al. used a **GradientBoostedRegressor**, which ranked the previous video view count, channel view count, and the number of subscribers, among other features, as the most important factors that determine a video's success.

While Wang et al.'s work showed how hard it is to predict future video success, their article highlighted two approaches that served to guide our own work. First, they inspired our analysis of YouTube thumbnails in order to extract meaningful features. Second, their results underscored the importance of feature selection: our work aimed to use only features that can be influenced by the creator when uploading the video, and not comments, likes/dislikes. We also avoided using features that the content creator does not have control over, like their number of subscribers or previous viewer count. Third, the work presented above also encouraged us to focus on seeking to predict user engagement rather than views, since the views might be harder to predict based only on features like thumbnail, title, description, and category/tags. In this paper we defined engagement as:

$$\text{Engagement} = \frac{\text{\# of Likes} + \text{\# of Dislikes}}{\text{\# of Views}}$$

3 Data Processing

The data was downloaded from Kaggle and contains 40,949 rows, each representing a trending video from a given day [2]. The data set contains 16 variables for each video, including trending date, title, channel title, category, views, comment count, thumbnail, and description. The videos included in this data set were trending over an eight month period between November 14, 2017 and June 14, 2018. However, the data was scraped on a day-to-day basis, indicating that since some videos were trending over a longer period of time and on multiple days, there were multiple videos that were duplicated in the data for a maximum of 30 times, representing that they had trended for upwards of 30 days. We removed all video duplicates except for the first occurrence, which corresponded to the first time the video became trending. Afterwards, we removed all videos that lacked a description, tags, or categories, which left us with a data set containing **5958 unique videos**.

We narrowed down the 16 variable to only those that were already available at the time of publishing: thumbnails, titles, descriptions, and tags. Furthermore, we decided to use a single **bag-of-words model** to represent the data set's **textual data**, which we define as the titles, descriptions and tags containing human-readable words, as well as **thumbnail data**. Because we are combining so many components of the data set into one, a bag-of-words model was the best fit as it simply counts the number of occurrences for each features without weighting the type of feature differently.

3.1 Textual Data

Upon our first review of the textual data, we found that in most of the descriptions, content creators included many special characters and URL links to catch viewer's attention or to promote other websites. As the special characters interfered with the textual data, and the URL links were unique and not repeated throughout all videos, we decided to remove such features from the data. This left us with only the words that the video descriptions had. We then tokenized and lemmatized all the words and used CountVectorizer to create a bag-of-words model, which resulted in a total of 37,351 different features. However, we found that the majority of these features occur fewer than five times each throughout all videos. Thus, we removed these low-occurrence features and extracted 9,333 for the descriptions instead. In order to preprocess this data we used the Python NLTK and gensim packages [9, 12].

Similarly, although the titles did not have URL links, we also processed this data by removing special characters, tokenizing, and then lemmatizing the words. This resulted in 9,316 total features in the bag-of-words model. However, as with the descriptions, we removed words that occurred fewer than five times in all titles, which left 1,738 features for the titles.

For the tags, upon initial analysis, we found that the videos had an average number of 20.09 tags each. Since about half of these tags were used to tag fewer than five videos in the entire data set, we removed those that occurred fewer times. This allowed the average number of tags per video to reduce down to 10.76 tags instead. Furthermore, as all tags were initially grouped together in the data set in a list format for each video, we separated them and then also created a bag-of-words model representation for all. This resulted in a total of 4,413 number of features in the BoW for tags.

3.2 Thumbnail Processing

In addition to analyzing textual data, we also aimed to extract features from the thumbnails of the videos. This is because thumbnails are one of the first things that a user might see when choosing to watch a video or not. As a result, we used the OpenCV open-source computer vision library in conjunction with the Yolo ("You Only Look Once") deep learning model in order to extract objects and their probability from the thumbnails [5, 11]. According to the Yolo paper, the model frames object detection in images as a regression problem with object bounding boxes and associated probabilities for each class of objects, using only one neural network that evaluates each image only once [11]. The Yolo system is simple to understand: all input images are resized to 448x448 pixels and then processed by only one convolutional network that outputs both image bounding boxes and class probabilities [11]. In their paper, Redmon et al. highlight how the Yolo system models detection as a regression problem that divides each image in a $S \times S$ grid, predicts B bounding boxes and their confidence together with C class probabilities, which are then encoded in a $S \times S \times (B*5+C)$ tensor. The end Yolo detection network architecture relies on 24 convolutional layers and 2 fully connected layers [11]. The Yolo model is featured in Figure 1, which is taken from the original paper [11].

For this project, we used a pre-trained version of Yolo using the COCO dataset [13, 8], which includes 80 object labels. The object labels included "person," "bottle," "cell phone," etc. An example thumbnail can be found in Figure 2. In order to extract features that could be used to predict engagement, we created a one-hot encoding array of length 80 for each video into a bag-of-words model. The array has a length of 80 because each spot represents one type of object that Yolo could detect.

As with the textual data, we removed certain features in the thumbnail data that occurred fewer than five times. This left us with a total of 38 different features that reoccurred across all thumbnails.

3.3 Combining Features

With the separate bag-of-words models for the titles, descriptions, tags, and thumbnails, we combined all four into one single BoW structure. As we also perform LDA on the features later on to create more cohesive latent structures, we needed to distinguish whether a feature was part of a title or part of another aspect of the video. Thus, we added prefixes to each feature: $d\_$ for *description*, $t\_$ for *title*, $ta\_$ for *tag*, and $p\_$ for *thumbnail photos*. In total, our bag-of-words contains 15,262 features for all videos.

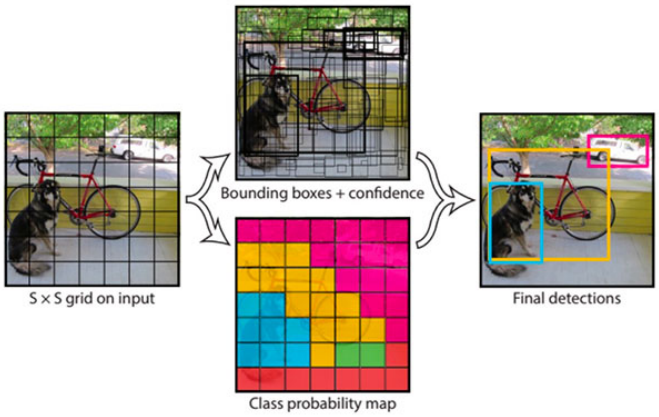


Figure 1: The Yolo model design [11].

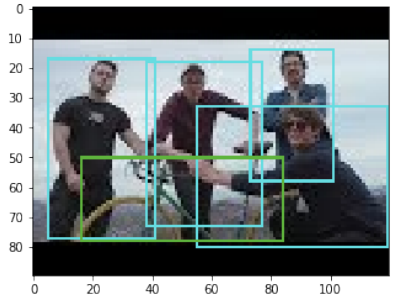


Figure 2: Example of analyzed YouTube video thumbnail. Blue bounding box refers to the **person** label, while the green bounding box represents a **bicycle** label.

4 Models

4.1 Elastic Net

In our initial analysis of predicting user engagement, we used Lasso and Ridge Regressions and Elastic Net. After preliminary analysis on mean squared error and R2 score, we decided to continue through with Elastic Net because of superior performance in both those categories. There were 3 non-default parameters that we used for Elastic Net:

1. Alpha: 0.002 because it returned the highest R2 score and lowest mean squared error, described further in the results section.

2. L1 ratio: 0.4 because it returned the highest R2 score and lowest mean squared error, described further in the results section

3. Max Iterations: 2000

4.2 LDA

Latent Dirichlet Allocation, or LDA, aims to uncover the hidden latent structures, especially within a large text document. It holds an underlying assumption that such documents includes multiple topics, each of which is distributed over a variety of words. As we had previously processed our thumbnail data into text and combined it with other textual data into one large bag-of-words model, LDA thus helps our research by revealing the underlying topics in one feature set [3, 4].

We use the following parameters to define LDA: $\beta_{1:K}$ are the topics and each is distributed over the entire vocabulary, $\theta_{d,k}$ is the proportion for the topic k in document d, $z_{d,n}$ is the topic assignment for word n in d, and $w_{d,n}$ is the word n in d. LDA can then be represented as the joint distribution

of observed and latent variables, as seen by the equation below:

$$p(\beta_{1:K}, \theta 1 : D, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \Big(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{d,n}, z_{d,n}) \Big)$$

Referring to Blei et al.'s [4] algorithm for performing LDA, we have:

1. Select $N \sim Poisson(\xi)$.
2. Select $\Theta \sim Dir(\alpha)$
3. For each word w_n of N words:
   (a) Select $z_n \sim Multinomial(\theta)$
   (b) Select $w_n \sim p(w_n|z_n, \beta)$, given that $p(w_n|z_n, \beta)$ is a multinomial probability conditioned on topic z_n

In our use of LDA, we defined the parameters as the documents are the YouTube videos and the words are the features in the bag-of-words model. We had initially varied α and η variables, but found that the default values were best for our implementation. We had also varied the number of topics and the number of components that we wanted in our model and found that using 10 components resulted in the best groupings for the features. This can be seen in Figure 8 and Figure 8 in the LDA results subsection.

5 Evaluation

5.1 Elastic Net Evaluation

We evaluated the performance of Elastic Net by using two different metrics described below.

1. Mean Squared Error: Mean squared error is the average squared distance of the predicted values to the true values, which in our case is GPA. The equation is as follows:

   $$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2$$

   We will compare mean squared error with the baseline mean squared error to see "how bad" the MSE could possible be. The baseline mean squared error will create the same predicted engagement for all videos, which is just the average of all the predictions. We decided to compare the MSE with the baseline MSE based on Prof. Barbara Engelhardt's suggestions during the poster session.
2. R2 Score: R2 Score is the normalized version of mean squared error meaning that it can be compared across data with different scales. The formula is as follows:

   $$R2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

This section mirrors the metrics used by Katherine in assignment 2.

5.2 LDA Evaluation

In order to evaluate topic models, Wallach et al. highlight two methods: "measuring performance on some secondary task, such as document classification or information retrieval, or by estimating the probability of unseen held-out documents given some training documents" [15]. In this paper, we evaluate LDA primarily by looking at the performance of Elastic Net on Subcategories. More specifically, this refers to evaluating LDA by considering how our prediction of engagement differs when running Elastic Net on all categories, the entertainment category, and the Ellen Degeneres Subcategory. An improvement or deterioration in the R2 and MSE scores can be attributed to the

performance of LDA on this data set. Additionally, we also evaluated LDA by analyzing the resulting subtopics manually, in the same way we did for Assignment 3. This helped us observe whether the resulting latent topics made sense.

We also evaluated LDA by calculating the **perplexity** and **log-likelihood** scores on varying numbers of latent components within the "Entertainment" category (from **5** to **50** components). This follows Theodor Marcu's Assignment 3 analysis on the OKCupid data set. Blei used the perplexity score to quantify the results of the algorithm when he introduced it in 2003 and it measures the generalization performance [4]. Hence, a lower score on a test data set indicates better model performance. This score is algebraically equivalent to to the inverse of the geometric mean per-word likelihood, as formally stated in the following equation (adapted from Blei and Theodor's Assignment 3):

$$perplexity(D_{test}) = exp\{-\frac{\sum_{d=1}^{M} log_p(w_d)}{\sum_{d=1}^{M} N_d}\}[4]$$

In addition to the perplexity score, we also calculated the average log-likelihood per data point. According to Morgan Giraud, this function is the log of the likelihood function that is equal to the probability of the dat aset given the model and parameters [6]. According to Giraud's online article, the likelihood function makes two important assumptions: data points are independent and similarly distributed. This applies to the latent topics within the "Entertainment" category. This measure can be calculated using the *score()* function in the SciKit Learn Python package and library [10].

6 Results

6.1 Elastic Net Main Results

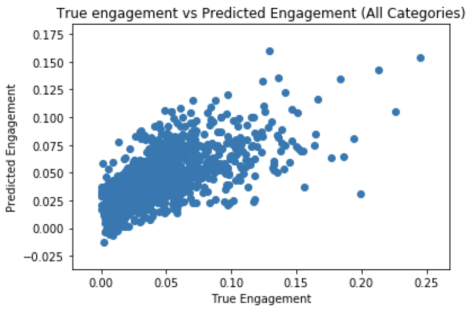


Figure 3: True Engagement vs. Predicted Engagement (All Categories).

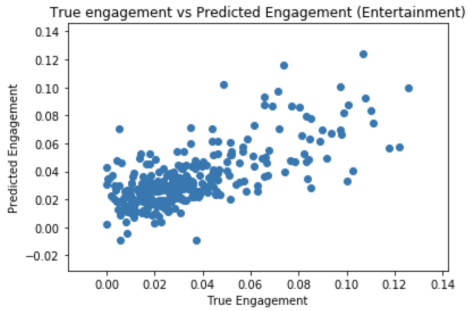


Figure 4: True Engagement vs. Predicted Engagement (Entertainment Category).

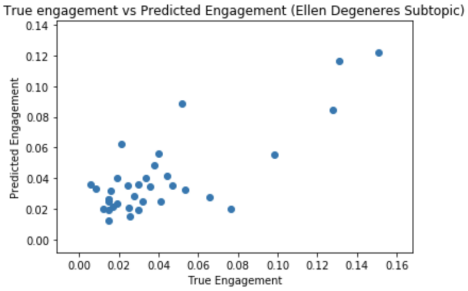
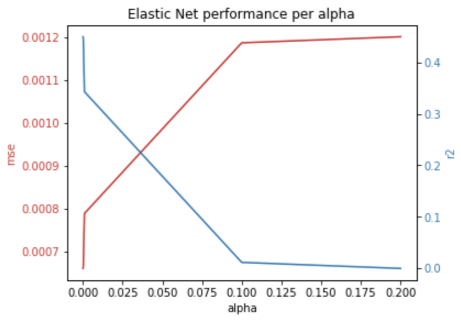
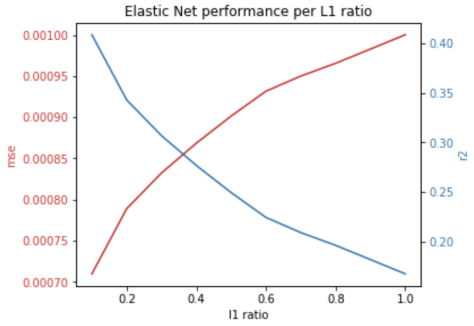


Figure 5: True Engagement vs. Predicted Engagement (Ellen Subtopic - Found by LDA).

We ran Elastic Net to predict the engagement of a video on varying scopes in the dataset. Our hypothesis was that the narrower the scope, the better we would be able to predict engagement. From the results table, we see that our hypothesis seems to hold true.





	Elastic Net R2	Elastic Net MSE	Baseline MSE
All Categories	0.455	0.00065	0.00120
Entertainment Category	0.479	0.00036	0.00068
Ellen Degeneres Subcategory	0.613	0.00048	0.00188

As we narrow down by category, we see that our model becomes more accurate in predicting engagement. First, the R2 score is steadily increasing as we narrow down the category, which aligns with our hypothesis. Secondly, it seems that MSE is extremely small, but it is also hard to tell without context, which is why we provided the baseline MSE. The baseline MSE demonstrates how bad the MSE could be if the prediction was the same across all videos. We took this prediction to be the average of the predictions from Elastic Net. We see that Elastic Net MSE is around 50% of the baseline MSE for all categories and the entertainment category, which shows that our model is note-able more effective in predicting engagement. What is interesting is that in the Ellen Degeneres subcategory, the Elastic Net MSE is around 25% of the baseline MSE which means it performs better in relation to the baseline MSE than the other two categories. Therefore, when looking at raw MSE values, the entertainment category seems to have the best performance in terms of MSE, but when we are looking relative to the baseline MSE, we see that the narrowest category again performs the best. This is because in the subcategory, the data points are more sparse, yet the Elastic Net model is able to overcome this difficulty. Overall, our two metrics seem to indicate that narrowing down by category can create a more accurate window for predicting engagement.

6.2 Elastic Net Parameter Results

First, we ran Elastic Net on a range of alpha values from 0 to 1 and found that for alpha values 0.1 and higher, mean squared error and R2 score stayed stagnant. Then within the range of 0 and 0.1, we found that an alpha value of 0.02 produced peak performance, as seen in the graph above.

With alpha fixed to 0.02, we then ran Elastic Net over a range of l1 ratios. From the graph above, we see that the model produces the best results when l1 ratio is around 0.4. This value makes sense because l1 is the mixing parameter and therefore 0.4 means our Elastic Net model uses both the l1 penalty and l2 penalty, allowing it to overcome the challenges that Ridge and Lasso face individually.

6.3 LDA Results

When looking at the resulting subtopics manually, LDA performed best on **10 components**. Fewer or more components resulted in categories that were not as easy to distinguish from one another. One of the best subcategories that we picked up on in the entertainment category was related to videos that referenced or included Ellen DeGeneres, which can be seen in Figure 6.

```
topic 9
                Component     Weight
102             p_person  78.636861
111              ta_ellen  75.099989
113  ta_ellen degeneres   74.099989
145   ta_the ellen show   72.099989
112   ta_ellen audience   65.099989
```

Figure 6: Top 5 features for the ”Ellen” subcategory within the ”Entertainment” category of videos. The ”p_person” feature refers to a person being detected in the thumbnail image for the video, while the features that start with ”ta” refer to textual features present in the title, description, or tags.

As mentioned above, we also evaluated LDA using the perplexity and log-likelihood scores. These scores are plotted in Figure 7 and Figure 8. These scores revealed that 10 was an optimal number of components, but that the LDA model still tended to over-optimize on the training set.

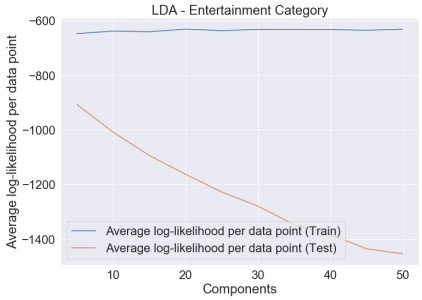


Figure 7: Average log-likelihood per data point.

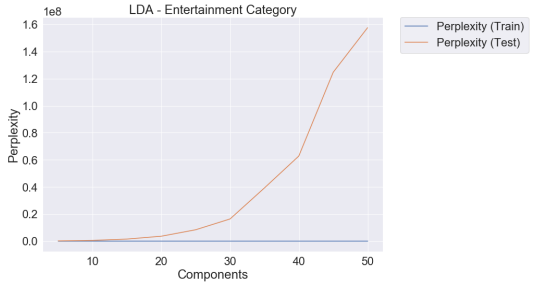


Figure 8: Perplexity scores for different components.

6.4 Influential Features

In order to better understand the results of our analysis, we also looked at specific videos within the Ellen subcategory that performed better or worse. This is because we wanted to see if there was a type of vide that Elastic Net had better results on. From our manual analysis, we found that our model performed better on videos with fewer likes, dislikes, and views, but a more comprehensive analysis is needed. An example of this can be seen in Figure 9 and Figure 10.

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count
241	vZzlS17HexE	17.15.11	Taylor Swift Instagram Story - Target 11/14/17	reputationswift	24	2017-11-15T03:11:17.000Z	Taylor swift Instagram Story|"Target"|"Taylors...	3006	75	4	10

Figure 9: Example of a video that performed well: Taylor Swift-related video with a few likes/dislikes/views.

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count
37157	rjh6CbETwn8	18.27.05	Giant Aluminum Ball Vs Oobleck from 250cm!	Guava Juice	24	2018-05-25T20:59:15.000Z	Oobleck|"how ridiculous"|"giant anvil vs"|"anv...	808842	24289	1816	4482

Figure 10: Example of a video that performed well: Giant Ball-related video with more likes/dislikes/views.

7 Discussion and Conclusion

For the feature set model, we do realize that bag-of-words and LDA have certain drawbacks. For both, they heavily rely on individual words and not phrases of words. Since the titles and descriptions

hold phrases that could repeat more generally than single words do, it would be interesting to create feature sets that have commonly reoccurring phrases instead. This would reduce the total number of features that we have in our model, improve the engagement predictions in our Elastic Net models, and could also possibly give us better results when running LDA and finding subcategories.

Another analysis that is interesting to explore would be to run Yolo on the first 30 seconds of each video. Yolo is well-suited for this, since it can run on videos up to 45fps [13]. Identifying the kinds of objects that are noticed by users in the beginning of videos might provide a better understanding on how the first 30 seconds of a video determine whether the user will view or not enough of it so it's counted as a view.

Acknowledgments

We appreciate the feedback we received during the poster session leading up to the paper deadline. Prof. Engelhardt's comments on establishing a baseline MSE proved really useful to our analysis. The comments made by Matt Myers, a COS 424 TA, also helped, since he suggested that we choose a few videos that did well/bad and see how their features differ.

Additionally, we would like to acknowledge the help of Jonathan Lu, a COS 424 TA, for help with understanding LDA and other latent variable models during Assignment 3. We also acknowledge the fact that our code was inspired from previous assignments where we worked with different partners and this is noted directly in our code. We would also like to thank the COS 424 Course Staff as a whole for their helpful precept notes and Piazza responses.

References

[1] Press - youtube. `https://www.youtube.com/yt/about/press/`. (Accessed on 04/22/2019).

[2] Trending youtube video statistics — kaggle. `https://www.kaggle.com/datasnaek/youtube-new#US_category_id.json`. (Accessed on 04/22/2019).

[3] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012.

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

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Morgan Giraud. Ml notes: Why the log-likelihood?, Jun 2017.

[7] Quyu Kong, Marian-Andrei Rizoiu, Siqi Wu, and Lexing Xie. Will this video go viral? explaining and predicting the popularity of youtube videos. *arXiv preprint arXiv:1801.04117*, 2018.

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[9] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[12] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[13] Adrian Rosebrock. Yolo object detection with opencv, Feb 2019.

[14] saurabhmathur96. saurabhmathur96/clickbait-detector, Oct 2017.

[15] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

[16] Allen Wang, Aravind Srinivasan, and Ryan O'Farrell. Youtube views predictor, Dec 2017.