

Introduction

The prediction problem I have selected is the classification of YouTube comments as spam or not-spam. YouTube is an incredibly popular video platform owned by Google, who reports that the site has 1 billion users, with 300 hours of content uploaded every minute and billions of views generated every day (YouTube). The popularity of the site, and the presence of a comments section on videos, has led to the prevalence of spam. Machine learning researchers define spam on YouTube as comments that have a promotional intent or are contextually irrelevant to the context of the video (Aiyar et. al). This definition includes comments that disseminate viruses and malwares (Alberto et. al). Spam comments negatively impact the credibility of a YouTube channel, as well as the user experience of comment-reading users (Aiyar et. al). Despite the prevalence of spam comments in YouTube channels, YouTube has only tackled the problem with limited methods (Aiyar et. al). Because of the limitations in YouTube's approach to spam detection, content creators have resorted to manually disabling comments on their videos (Alberto et. al), a tedious, incomplete, and user-unfriendly solution. It is clear that a more sophisticated solution is necessary to solve the problem. In this paper, I discuss a machine learning approach to classify comments as spam or non-spam.

Aiyar et. al divides spam comments into two buckets: link-based spam and channel promotional spam. I approached the problem with this definition in mind. For instance, when performing the error analysis I searched for ways that new features could address the existing problematic features by better classifying either link-based spam or promotional spam.

Setting Up The Data

The dataset came from the UCI Machine Learning Repository and is available at the link <https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection>. It came already in a tabular form, ready to be input into a machine learning algorithm. The data was collected using the YouTube Data API v3 and was donated to the UCI Machine Learning Repository in 2017. There are 1956 instances in the data. Features include artist, author, content, date the comment was published, and class. There are comments from 5 artists -- Psy (350 instances), Katy Perry (350 instances), LMFAO (438 instances), Eminem (448 instances), and Shakira (370 instances). The five videos were among the 10 most viewed in the collection period.

I split the dataset into three parts: a cross-validation set with 70% of the data, a development set with 20% of the data, and a test set with 10% of the data. When splitting the data, I made the proportions of the five artists relatively equal across the dataset, using the following breakdown:

Cv.csv → 1369 instances

- 1-246 (Psy)
- 352-597 (Katy Perry)
- 702-1007 (LMFAO)
- 1140-1453 (Eminem)

- 1588-1846 (Shakira)

Dev.csv → 391 instances

- 247-317 (Psy)
- 598-668 (Katy Perry)
- 1008-1095 (LMFAO)
- 1454-1543 (Eminem)
- 1847-1920 (Shakira)

Test.csv → 196 instances

- 318-351 (Psy)
- 668-701 (Katy Perry)
- 1096-1139 (LMFAO)
- 1544-1587 (Eminem)
- 1921-1957 (Shakira)





The initial feature space representation was set up to extract unigrams. No data cleanup was necessary. The class value that I tried to predict was whether or not a comment was spam, represented by a 1 for spam and an 0 for non-spam. The researchers who posted the dataset did not give explicit information about how the class value was obtained, although it was possibly collected by web scraping followed by crowdsourced surveys.

Error Analysis

After testing various algorithms on the CV set using a ten-fold cross-validation, including Naive Bayes, Logistic Regression with L2 Regularization, Support Vector Machines, Decision Trees, Logistic Regression with L1 Regularization, and Logistic Regression with L2 Regularization (Dual), I found that Logistic Regression had the best performance with respect to both percent accuracy and Kappa.

With Logistic Regression with L2 Regularization I got a baseline accuracy of 0.9423 and a baseline Kappa of 0.8839. I then examined the horizontal and vertical absolute differences of various unigram features in the error cells, in an attempt to identify problematic features. I evaluated two problematic features based on the horizontal absolute difference, and two problematic features based on the vertical absolute difference.

Horizontal Feature #1

Act \ Pred	0	1
0	 711	 5
1	 74	 580

Search:

Feature	Frequency	Horizont...	Vertic...	Feature W...
<input checked="" type="radio"/> this	6	0.3189	0.1805	0.319
<input type="radio"/> i	9	0.1267	0.1287	0.7133
<input type="radio"/> song	4	0.0097	0.1246	0.9523
<input type="radio"/> is	5	0.0117	0.1195	0.8435
<input type="radio"/> love	4	0.0041	0.088	0.9827
<input type="radio"/> the	8	0.0988	0.0804	0.2802
<input type="radio"/> and	3	0.256	0.0762	-0.1795
<input type="radio"/> that	0	0.0517	0.0675	0.7147
<input type="radio"/> in	2	0.0799	0.0658	0.4182
<input type="radio"/> to	3	0.2508	0.0607	-0.235
<input type="radio"/> it	3	0.1422	0.0607	0.6617
<input type="radio"/> views	0	0.0172	0.0605	1.5165
<input type="radio"/> so	2	0.0506	0.0531	0.912
<input type="radio"/> ôªø	26	0.0289	0.0476	-0.1262
<input type="radio"/> like	2	0.1333	0.0475	-0.6169
<input type="radio"/> you	2	0.2419	0.0447	-0.0836

As I was doing error analysis, I noticed that the term “this” had a high horizontal absolute difference and a high feature weight. While the frequency was not incredibly high, it had the solid tradeoff between feature weight, horizontal difference, and frequency. When looking at the instances, I noted that four out of six of these involved the term “this” in reference to a link. Examples include “<https://www.facebook.com/pages/Hiphop-Express/704682339621282> like this page yo” and “<http://psnboss.com/?ref=2tGgp3pV6L> this is the song”.

☒ Filter documents by selected feature

☐ Reverse document filter

☒ Documents from selected cell ...

Instance	Predicted	Actual	Text
<input checked="" type="checkbox"/> 154	0	1	reminds ...
<input checked="" type="checkbox"/> 248	0	1	http://ps...
<input checked="" type="checkbox"/> 269	0	1	I really lo...
<input checked="" type="checkbox"/> 422	0	1	https://...
<input checked="" type="checkbox"/> 439	0	1	this song...
<input checked="" type="checkbox"/> 1298	0	1	I love thi...

Instance 154 (Predicted 0, Actual 1)
Highlighting this feature hits

reminds me of this song <https://soundcloud.com/popaegis/wrenn-almond-eyes>

Instance 248 (Predicted 0, Actual 1)
Highlighting this feature hits

<http://psnboss.com/?ref=2tGgp3pV6L> this is the song

Instance 269 (Predicted 0, Actual 1)
Highlighting this feature hits

I really love this video.. <http://www.bubblews.com/account/389088-sheilcen>

When I looked for a unigram that included “http”, or “https” i noted these are not being included in the feature space. Picking up on the presence of “http” or “https” may help indicate whether or not a comment is spam.

Horizontal Feature #2

I next examined the feature “I”, which appeared to have a high horizontal absolute difference, a high feature weight, and a relatively high frequency, making it a good candidate for a problematic feature. When examining instances containing “I” that were correctly classified as spam, a lot of them had other unigrams within the sentence that would indicate that they are spam (bolded); examples include “Hey guys check out my new channel and our first vid THIS IS US THE MONKEYS!!! I'm the monkey in the white shirt,please leave a like comment and please **subscribe!!!!**” and “and u should.d **check** my channel and tell me what I should do next!”. On the other hand, instances containing “I” that were actually spam but incorrectly classified as not spam tended to not contain these unigrams. Examples include “I love katy fashions tiger, care to visit my blog sinar jahitan I also have the tiger collections tqvm” and “this song is so addicting. the hook is dope and catchy. love the video too. I'm getting popular fast because i rap real.. thumbs up if you piss next to the water in the toilet so its quiet.....”. It seems that, without key unigrams like “check” and “subscribe”, the algorithm may not be picking up on the fact that these comments are in fact spam.

Vertical Feature #1

Feature	Frequency	Horizont...	Vertic...	Feature W...
<input type="radio"/> will	0	0.062	0.0169	0.219
<input type="radio"/> still	1	0.0034	0.0172	0.3942
<input type="radio"/> shakira	1	0.0068	0.0187	0.4664
<input type="radio"/> watc...	0	0.0258	0.0197	0.516
<input type="radio"/> with	0	0.0568	0.0225	-0.1769
<input type="radio"/> billion	1	0.012	0.0229	1.0871
<input type="radio"/> emin...	0	0.0293	0.0253	0.6952
<input type="radio"/> video	3	0.1861	0.0264	-0.3442
<input type="radio"/> music	0	0.0671	0.0281	0.6106
<input type="radio"/> like	3	0.1173	0.0334	-0.5875
<input checked="" type="radio"/> it	4	0.1259	0.0465	0.6433
<input type="radio"/> views	1	0.0018	0.0468	1.5528
<input type="radio"/> in	2	0.0793	0.0654	0.437
<input type="radio"/> i	10	0.1091	0.1134	0.7248
<input type="radio"/> is	5	0.0107	0.1186	0.786
<input type="radio"/> this	7	0.3017	0.1657	0.3435

I next examined the feature “it”. It’s vertical absolute difference is moderately low, but not very low, and the feature weight is high. The frequency is a little low, but it was the best option to examine considering the trade-offs between vertical absolute difference, frequency, and feature weight. When comparing between instances classified as not-spam but actually spam and instances correctly classified as spam, I noted that in the correctly classified instances, the “it” tended to refer to another channel or account that the user was asking others to visit. Examples include “Hello! Do you like gaming, art videos, scientific experiments, tutorials, lyrics videos, and much, much more of that? If you do please check out our channel and subscribe to it, we’ve just started, but soon we hope we will be able to cover all of our expectations... You can also check out what we’ve got so far!” and “Hey guys! Im a 12 yr old music producer. I make chiptunes and 8bit music. It would be wonderful if you checked out some of my 8bit remixes! I even have a gangnamstyle 8bit remix if you would like to check that out ;) Thanks!!”. On the other hand, in the incorrectly classified instances in the error cell, the “it” tended to refer to something that was not the user’s channel or account. Examples include “IIIIIIIIII LOVE THIS SHAKE IT SONG OH SORRY EVERY SHAKE IT SONG I LIKE WATCH SUBSCRIBE AND

NEVER UNLIKE BROOOOO!!!!!!!!!!!! SHAKE IT UP” and “Fuck it was the best ever 0687119038 nummber of patrik kluivert his son share !” . This may make instances in this error cell look more similar to those correctly identified as spam, which do not include solicitations to check out one’s own content. Examples of instances in this cell include “i turned it on mute as soon is i came on i just wanted to check the views...” and “I dont even watch it anymore i just come here to check on 2 Billion or not”. This may reflect that the algorithm is interpreting spam as just solicitations to view one’s own channel or website, although spam may also simply be comments that are contextually irrelevant to the video (see Introduction). The algorithm may not be picking up on this distinction.


Vertical Feature #2

The final vertical feature that I examined was the unigram “in.” The vertical absolute difference was not as low a would be desired, and the frequency was not as high as desired, but this feature had the next best trade-off of low vertical absolute difference, high feature weight, and high frequency, after Vertical Feature #1. When examining instances containing “in” that were correctly classified spam instances, many examples included “in my channel” as they asked other users to engage in some action “in my channel.” Examples include “EHI GUYS CAN YOU SUBSCRIBE IN MY CHANNEL? I AM A NEW YOUTUBER AND I PLAY MINECRAFT THANKS GUYS!... SUBSCRIBE!” and “Check out pivot animations in my channel”. Alternatively, correctly classified non-spam instances did not contain these trigrams. Examples include “Behold the most viewed youtube video in the history of ever” and “The Guy in the yellow suit kinda looks like Jae-suk”. Neither of the two instances in the error cell used “in” in this way either: the instances in the error cell were “I #votekatyperry for the 2014 MTV #VMA Best Lyric Video! See who’s in the lead and vote: <http://on.mtv.com/Ut15kX>” and “this song is so addicting. the hook is dope and catchy. love the video too. I’m getting popular fast because i rap real.. thumbs up if you piss next to the water in the toilet so its quiet.....”. This may be causing these examples to look more similar to the non-spam instances when they are in fact spam instances. Like Vertical Feature #1 and Hoizontal Feature #2, it seems that the model is relying on features that indicate that a comment is soliciting something (like checking something out “in my channel”), and not enough on other features that may indicate that a comment may create a negative experience for other users.

Addressing Horizontal Feature #1

To address the problematic feature “this”, I decided to count character n-grams, namely 4-grams, to pick up on the http. I did not extract across whitespace or include punctuation.

Configure Character N-Grams

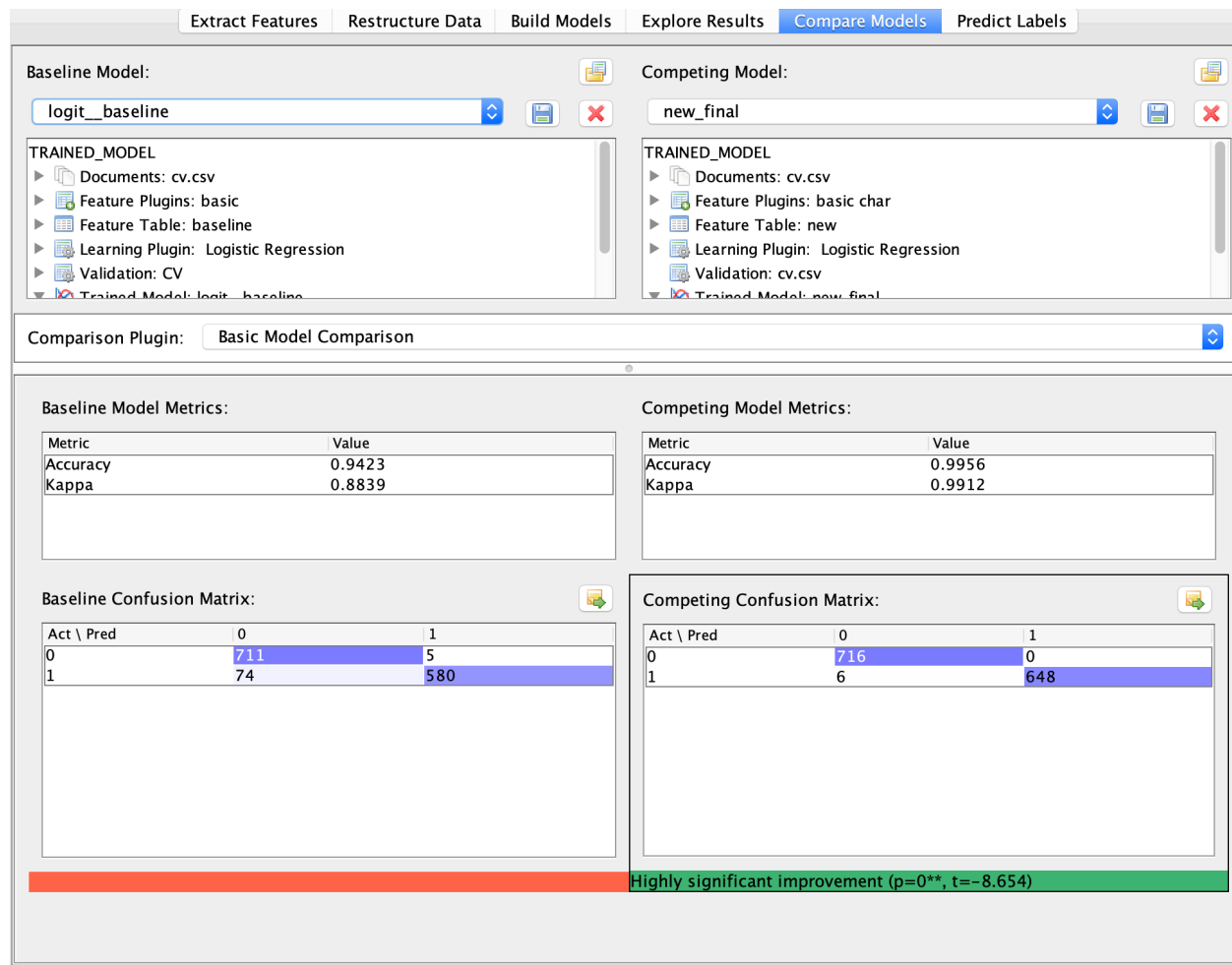
N= 

☐ Extract Across Whitespace

☐ Include Punctuation

☐ Track Hit Locations

The new accuracy was 0.9956 and the new Kappa was 0.9912. Clearly, both the Kappa and the accuracy increased significantly (see below).



Parameter Tuning

To perform parameter tuning, I began by dividing my CV set into 5 train-test pairs, yielding ten sets total. I used the stratifiedRemoveFolds filter in Weka, as there was no reason to maintain the original order of the data. I used logistic regression on the dataset and used the following parameter settings:

Setting **A** (default): maxIts=-1

Setting **B**: maxIts=5

Setting **C**: maxIts=10

From this point onwards, each setting will be referred to by its respective letter for simplicity.

When run on the cross-validation set with the extracted feature set as determined in error analysis (i.e. with all unigrams and 4-grams), the experiment took too long. As a result, I performed attribute selection using the ranker algorithm to select the top 12 attributes. After running an experiment with the three parameter settings using 5-fold cross-validation, I obtained

a Kappa of 0.62 as my baseline performance. The Stage 1 result showed a tie in Kappa for the Settings A and C, both with Kappa = 0.62. Setting B had Kappa = 0.61. As a result, I would choose Setting A to build the Stage 2 model, since it is the simplest setting.

Below is a table detailing the Stage 3 estimate of the model's performance on new data.

A	B	C	Optimal Setting	Test set performance
0.61	0.62	0.62	B	0.53 (0.57)
0.61	0.61	0.61	A	0.57
0.60	0.61	0.60	B	0.63 (0.65)
0.61	0.60	0.61	A	0.62
0.61	0.60	0.60	A	0.72

The average Stage 3 performance was 0.614

After running a t-test, I determined that the baseline performance was significantly greater than the tuned performance, with $p=0.208$. As a result, it is not worthwhile to do the tuning here. The baseline version of logistic regression will be used to evaluate against the final test set.

Final Evaluation

For the final evaluation, I used the updated feature space that included 4-grams, as determined from the error analysis, and I kept with the baseline parameter settings, as determined from the parameter tuning. I used the test set as a supplied test set and got accuracy = 0.98 and kappa = 0.96 as my final performance results. This was a significant improvement from the baseline performance, where accuracy = 0.94 and kappa = 0.88.

Discussion

This project, and this class in general, taught me a lot about how to think creatively to solve problems using the tools at hand. Mastering error analysis was a prime example of this. When I first began error analysis, I struggled to understand how the problems I was seeing could possibly have solutions. After working through examples with a peer for Assignment 3 and reviewing how the instructor worked through Assignment 3, I realized that my mental approach to solving these problems was too inflexible. I had been coming up with grand solutions that couldn't be implemented feasibly. With practice, I learned how to approach the problem with a creative mindset, and then figure out how to use the tools at hand to best address the problem. I

learned through this process that coming up with an imperfect solution that has an effect is better than coming up a perfect solution that can't be implemented.

Further, this process showed me how much more important the journey is, rather than the destination. A mindset that was emphasized in this project was to focus on the process of coming up with solutions, not on whether the solutions resulted in a significant improvement. Of course, improving performance is an important goal in the field of machine learning in general. However, I've discovered that you can learn much more when you have the "journey" mindset rather than the "destination," a mindset that I sometimes feel forced to put on in other classes and projects due to the nature of how they are structured and evaluated. Mastering the journey may result in advancements in the destination, years down the road after I've had experience with many datasets and many applications of machine learning.

Works Cited

Aiyar, Shreyas, and Nisha P Shetty. "N-Gram Assisted Youtube Spam Comment Detection." *Procedia Computer Science*, vol. 132, 2018, pp. 174–182., doi:10.1016/j.procs.2018.05.181.

Alberto, Tulio C., et al. "TubeSpam: Comment Spam Filtering on YouTube." *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, doi:10.1109/icmla.2015.37.

YouTube – Statistics (2015). <https://goo.gl/ozUXMB>