YouTube comments classification

Springboard Capstone - 2

Preeti Prajapati

Table of Contents

Introduction

The YouTube channels owners can access analytics on their videos on YouTube studio. There is a section for comments as well but it doesn't have any analytics on the content of the comments. It only shows a list of comments, comments held for review and likely Spam comments.

With this project I aimed to fill this analytics gap. I have tried to come up with an enhanced version of analysis on comments which can be useful to the channel owners to grow their views and revenue. The users will be able to better analyze what people are talking about. They can utilize this information in their upcoming videos to increase profit.

Audience

1. YouTube channel owners

2. This tool can be made available as a stand-alone tool as well and can be used by anyone to perform analysis on a channel/video of their interest.

Datasource

Training Dataset:

UCI's *YouTube Spam Collection Data Set:* The dataset has 1,956 real messages extracted from five videos that were among the 10 most viewed in the collection period. I am using this dataset for training the model.

http://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection#

Score Dataset:

I am using *Youtube API* to scrape comments for the given video url. There is an option to either scrape all the comments or given no. of comments from the video. The scrapper scrapes only the highest level of comments, not the replies to comments.

Data Wrangling

YouTube Spam Collection Data Set has 5 files, 1 for each artist.

Dataset	YouTube ID	Spam	Ham	Total
Psy	9bZkp7q19f0	175	175	350
KatyPerry	CevxZvSJLk8	175	175	350
LMFAO	KQ6zr6kCPj8	236	202	438
Eminem	uelHwf8o7_U	245	203	448
Shakira	pRpeEdMmmQ0	174	196	370

Each file has 5 columns: COMMENT_ID, AUTHOR, DATE,CONTENT and CLASS.

CLASS variable is set to **1 for Spam comments**, 0 otherwise.

	COMMENT_ID	AUTHOR	DATE	CONTENT	CLASS
0	z12rwfnyyrbsefonb232i5ehdxzkjzjs2	Lisa Wellas	NaN	+447935454150 lovely girl talk to me xxx	1
1	z130wpnwwnyuetxcn23xf5k5ynmkdpjrj04	jason graham	2015-05-29T02:26:10.652000	I always end up coming back to this song 	0
2	z13vsfqirtavjvu0t22ezrgzyorwxhpf3	Ajkal Khan	NaN	my sister just received over 6,500 new <a rel=...	1
3	z12wjzc4eprnvja4304cgbbizuved35wxcs	Dakota Taylor	2015-05-29T02:13:07.810000	Cool	0
4	z13xjfr42z3uxdz2223gx5rrzs3dt5hna	Jihad Naser	NaN	Hello I'am from Palastine	1

Sample data from YouTube Spam Collection Data Set

I merged all the files to create 1 training dataset. The merged dataset has balanced no. of records for Spam as well as Ham.

	No. of records	% of total records
1	1005	0.513804
0	951	0.486196

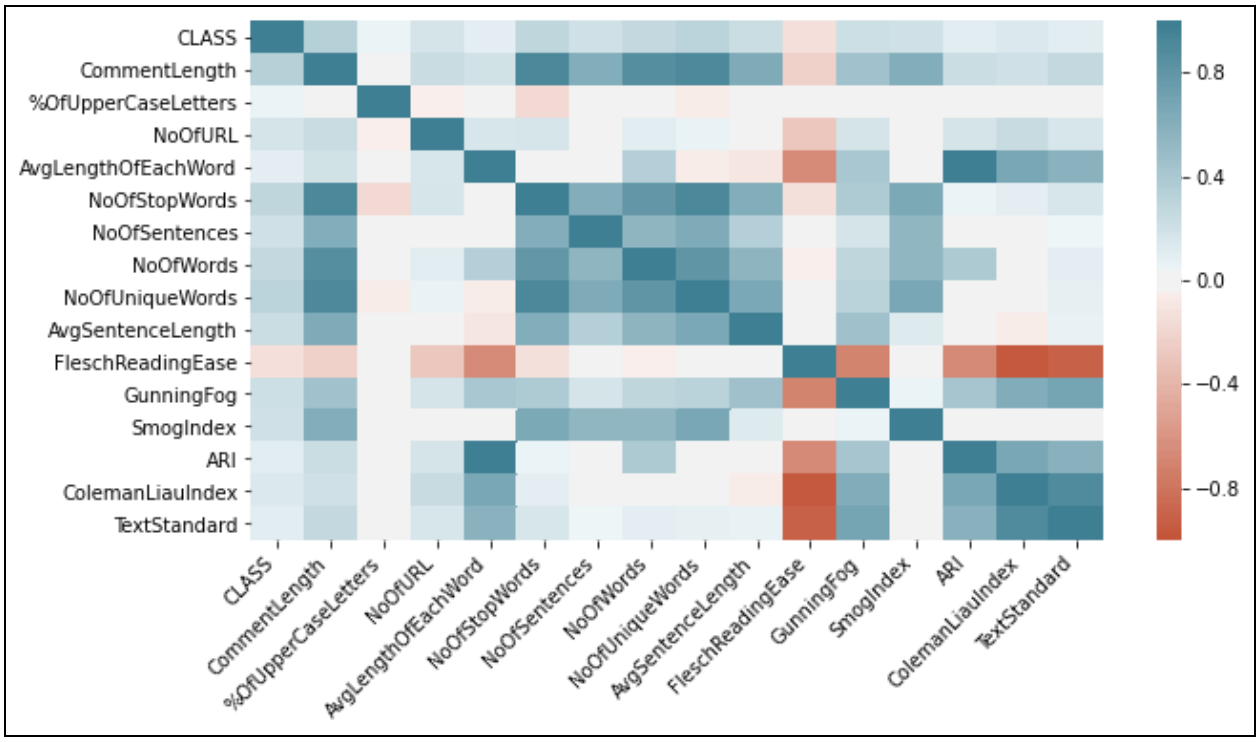
Distribution of target in training data

For the training purpose, I only need CONTENT and CLASS columns.

Exploratory Data Analysis

I found that there are no null and duplicate records. I added a few columns based on the CONTENT field to get an insight into the basic features of comments. Here is the list of features based on text of comments.

I also utilized the Textstat library to calculate a number of statistics from text to determine readability, complexity and grade level of comments. Next, I plotted the correlation matrix of all the generated features.



Correlation Matrix of engineered features

Findings:

1. Comment length is highly correlated with no. of stop words, no. of sentences, no. of words, no. of unique words. No. of stop words, no. of sentences, no. of words, no. of unique words are highly correlated to each other as well. This makes sense as

longer comments will tend to have more words, stop words, sentences and unique words.

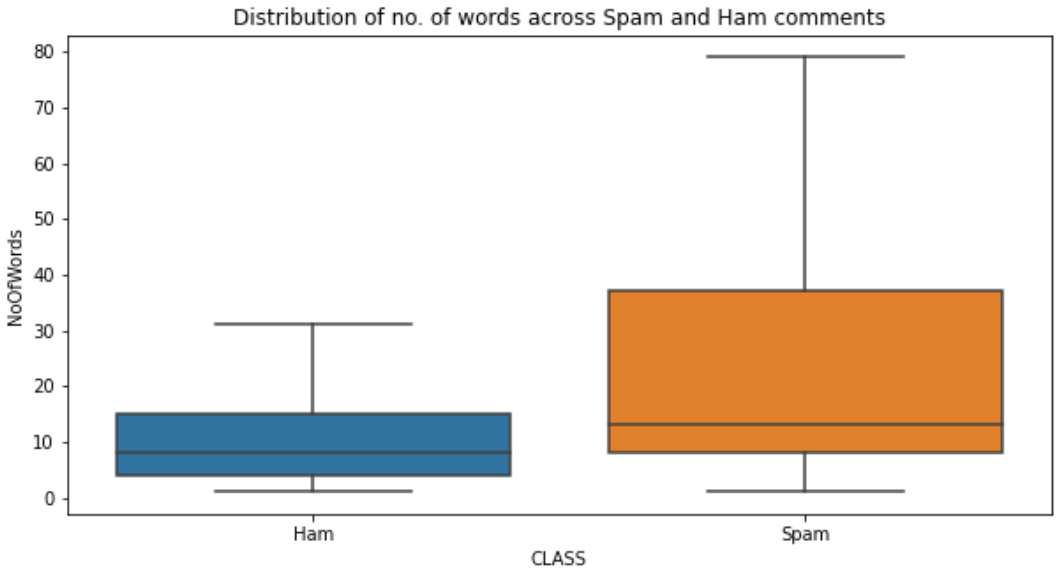
I replaced counts of stop words and unique words to make them independent of comment length and dropped comment length field.

I also dropped the no. of sentences field as it is highly correlated with no. of words.

2. The indexes about comments' understandability and readability are also positively or negatively correlated with each other. Instead of using all of the indexes, I will use Text Standard only. Test_Standard returns the estimated school grade level required to understand the text hence should be a good representative of all the other indexes.

I then explored the distribution of a few features against the target variable and their statistical significance.

1. **No. of words in each comment:**



On average, Spam comments are longer than Ham comments. Non-Spam comments are consistently shorter but Spam comments vary from short to long.

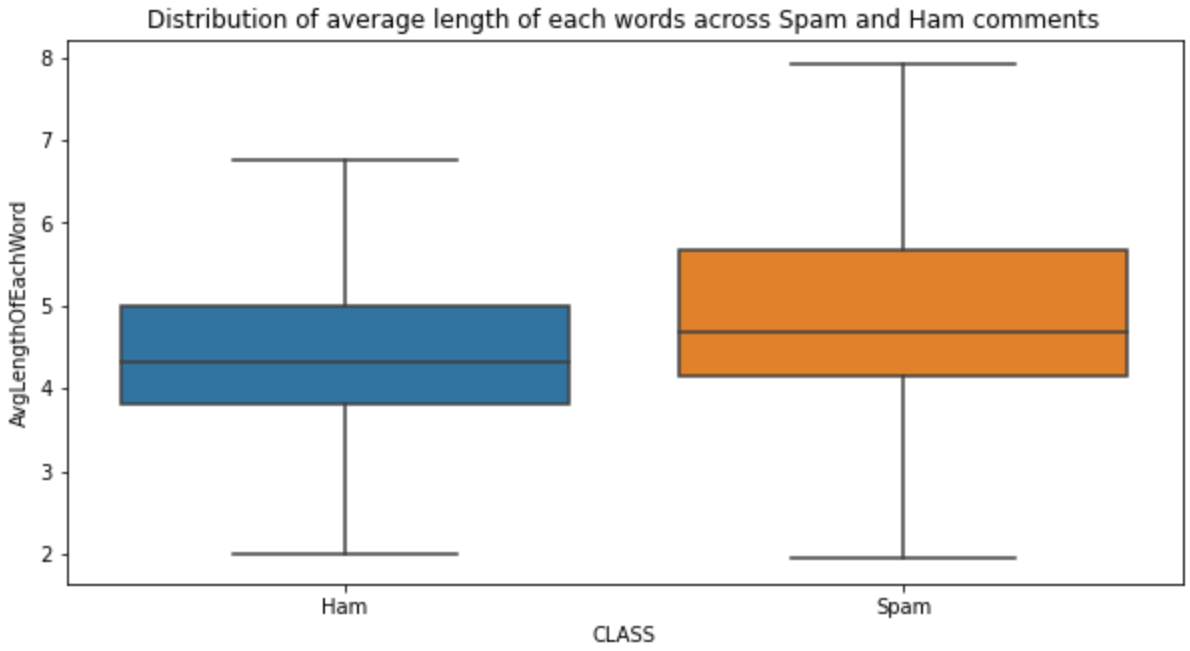
Hypothesis Testing: Are Spam comments usually longer than non-Spam comments?

Null hypothesis, H0= There is no difference in the length of comments in Spam or Ham category.

Alternate hypothesis, H1= Spam and Ham comments are different in length.

Conclusion: p_value is close to 0 and hence we **reject the null hypothesis** and can say that there is a difference in length Spam and Ham comments. From the boxplot, we can say that Spam comments have a wider range of length and are usually longer than Ham comments.

2. **Average length of each word**



On average, Spam comments have lengthier words than non-Spam comments.

Hypothesis Testing: Do Spam comments have longer words than non-Spam comments?

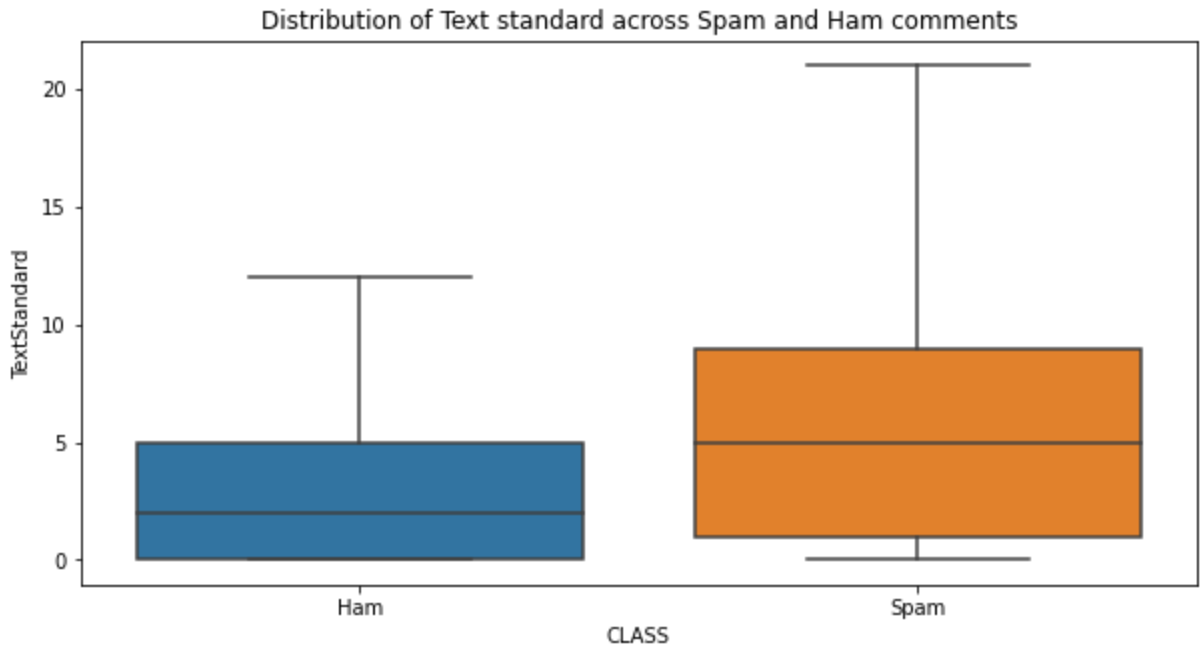
Null hypothesis, H0= There is no difference in the word length in Spam or Ham category.

Alternate hypothesis, H1= Spam and Ham comments have different word length.

Conclusion: p_value is close to 0 and hence we **reject the null hypothesis.** Hence, we can say that there is a difference in length of words in Spam and Ham comments. From the

boxplot, we can say that Spam comments usually contain longer words than Ham comments.

3. Text standard



There is a stark difference in text standard in the 2 categories. Text standard for Spam comments has a median at 5 grade level, while Ham comments' grade level is significantly lower at 2.

Hypothesis Testing: Do Spam comments have a different text standard than non-Spam comments?

Null hypothesis, H0= There is no difference in text standard in Spam or Ham category.

Alternate hypothesis, H1= Spam and Ham comments have different text standard.

Conclusion: p_value is close to 0 and hence we **reject the null hypothesis.** Hence, we can say that there is a difference in text standard in Spam and Ham comments. From the boxplot, we can say that Spam comments have a wider range of text standard and usually it is higher than Ham comments.

The details of above 3 hypothesis testings can be found **here**

Data Cleaning

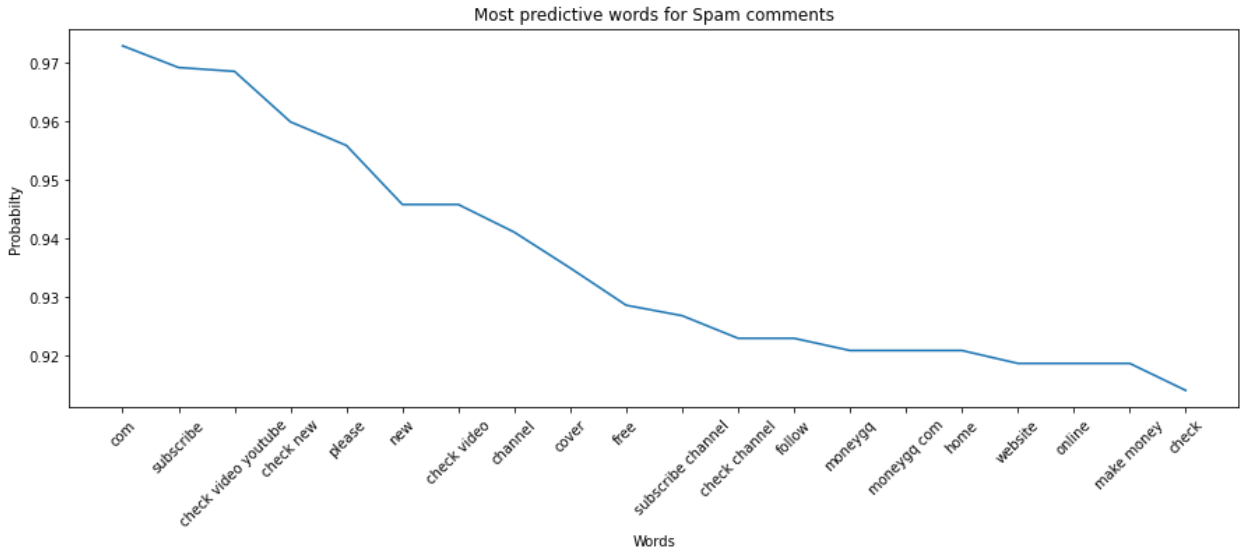
Due to the unstructured nature of text data, it becomes difficult for machine learning models to work directly on raw data. Hence to extract useful signals from data, it becomes more important to remove non-useful signals from the data. This is when the data cleaning step comes into play.

Upon investigating the dataset, I came up with a list of data-cleaning tasks to remove unnecessary characters, symbols and tokens from the comments. I started with removing URLS, html tags, english stopwords, punctuations, non-english characters etc. I also replaced the emojis with their corresponding text, ascent characters with standard characters and digits with '9's. Here is a jupyter notebook with details of cleaning tasks performed on the source dataset. The cleaned comments are saved in the 'CleanWordList' column in the dataset.

Most predictive words/group of words

I started with splitting the dataset into training and test dataset and then. vectorized the cleaned comment text. Later, I used the Multinomial Naive Bayes classifier to find the most predictive words for the Spam and Ham category.

Here are the most predictive words for Spam comments.



Findings:

1. We can see that most of the Spam comments have sentences like “Please like/check/subscribe/follow to my youtube channel” or “please like my video”.
2. The comments are talking about making money. Most likely luring people on the pretext of making free money from home.

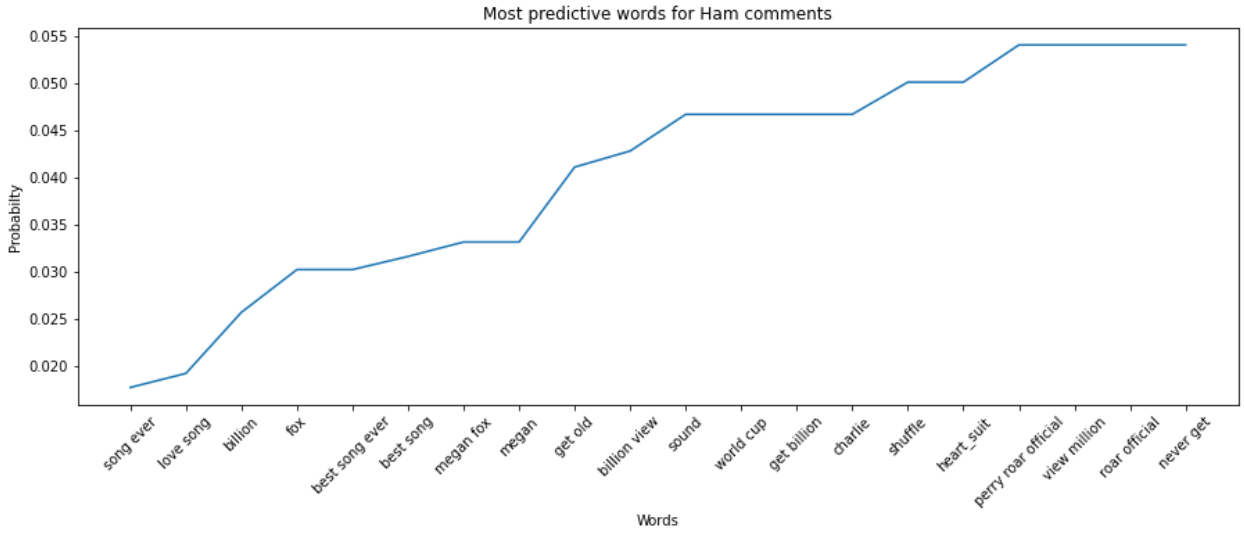
Here are a few comments with words related to making free money from home and predicted as Spam.

```
visit &quot; ww estiloproduction com &quot; best website to make money
---------------------------------------------------------------------------------------------------------------
Hello Guys...I Found a Way to Make Money Online You Can Get Paid To Mess Around On Facebook And Twitter! GET PAID UPTO $25 t
o $35 AN HOUR...Only at 4NetJobs.com Work from the Comfort of your Home... They are Currently Hiring People from all Over th
e World, For a Wide Range of Social Media Jobs on Sites such as Facebook,Twitter and YouTube You don&#39;t Need any Prior Sk
ills or Experience and You can Begin Work Immediately! You Can Easily Make $4000 to $5000+ Monthly Income…Only at 4NetJobs.c
om
---------------------------------------------------------------------------------------------------------------
You guys should check out this EXTRAORDINARY website called ZONEPA.COM .   You can make money online and start working from
home today as I am!   I am making over $3,000+ per month at ZONEPA.COM !   Visit Zonepa.com and check it out!  Why does the
statement conciliate the acidic stretch? The earth recognizes the money. When does the numberless number transport the trad
e?
---------------------------------------------------------------------------------------------------------------
You guys should check out this EXTRAORDINARY website called MONEYGQ.COM .   You can make money online and start working from
home today as I am!   I am making over $3,000+ per month at MONEYGQ.COM !   Visit MONEYGQ.COM and check it out!  Why does th
e fragile swim enlist the person? How does the ice audit the frequent son? The fantastic chance describes the rate.
---------------------------------------------------------------------------------------------------------------
New way to make money easily and spending 20 minutes daily --&gt; <a href="https://www.paidverts.com/ref/Marius1533">http
s://www.paidverts.com/ref/Marius1533</a>
```

These comments make you think that you can make lots of money working from the comfort of your home. But if this were true, wouldn't we all be working at home? Money-making scam is a big industry and it feeds off of people’s insecurities and fears.

There is a big section of stay at home moms and dads who choose to stay home to care for their kids over climbing the corporate ladder. Most of the time, these people try to find side-hustles where they can be home with their kids as well as make some money to support their families. Money making scams like these are targeted mainly on these stay at home parents. It is best to be aware of these kinds of comments which can turn out to be scams that take your money rather than help you make it.

Here are the most predictive words for non-Spam comments.

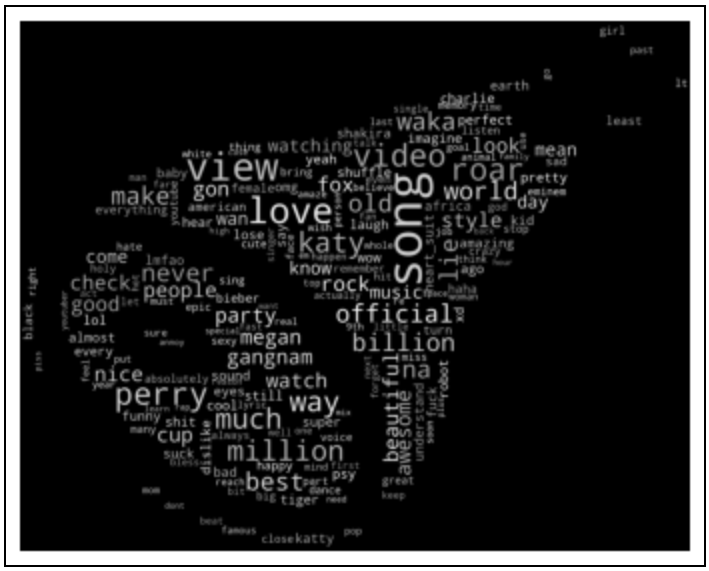


Findings:

1. Heart_suit, ♥, is the most used emoji in non-Spam comments.
2. Megan Fox and Katy Perry's names are one of the predictors of non-Spam comments. Katy Perry's video 'Roar' is also a predictor of non-Spam comments. Since the source data is comments from 5 video songs, words about the song/ artist are good predictors for Ham comments.

Word Cloud

Most predictive words for Ham comments



Most predictive words for Spam comments

Machine Learning

Train-test split:

I split the source data set by setting aside 70% of data for training and the remaining 30% for testing. It is very important to test the model performance on data which were not used in training the model. I stratified the split by the target variable to ensure the ratio of classes in both sets are identical.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3,stratify=y,random_state=147)
```

Text Vectorization:

Once the data were split into training and testing datasets, I converted the pre-processed comment text to a matrix of token counts. I tried two vectorizers from sklearn library, Countvectorizer and TFIDFVectorizer.

CountVectorizer: Counts the number of words in the document and creates a matrix of count of occurences of each word in the document. CountVectorizer represents each document in a standalone fashion and does not consider the rest of the documents in the corpus.

TfidfVectorizer: Converts a collection of raw documents to a matrix of TF-IDF features and is equivalent to CountVectorizer followed by TfidfTransformer. It gives more weight to the words appearing in a smaller no. of documents.

I compared the two vectorizers by putting each in a pipeline with a Multinomial Naive Bayes model and then grid searching for the following parameters:

1. **Scoring**: I have used roc_auc for scoring.
2. **Min_df**: I grid searched min_df=[0.1,0.01,0.001,0.0001]. Here, the numbers represent the percentage of the corpus. By doing this, I am excluding all the words/phrases that appear in less than X% documents from the vocabulary..

3. **Ngram_range**: I grid searched ngram=[(1,2),(1,3)]. Here, the numbers represent the range of the number of words allowed in a token. The advantage of using ngram_range over bag-of-words is to take into account the sequence of words. Bag-of-words does not consider the position of words in the sentence but in the real world the position of words can change the meaning of the sentence. For example: below sentences will have the same representation in BOW. But with ngram_range=(1,3), the first sentence will have features “I love Python” and “I hate Java” which is different from second sentence’s features “I love Java” and “I hate Python”.

I love Python but I hate Java *I love Java but I hate Python*

```
pipe_CompareVectorizer= Pipeline([('vectorizer',Transfomer()),
                                  ('clf', ClfSwitcher())
                                 ])
paramGrid_CompareVectorizer=[ {
                              'vectorizer__vectorizer':[TfidfVectorizer()],
                              'vectorizer__vectorizer__min_df':[0.1,0.01,0.001,0.0001],
                              'vectorizer__vectorizer__ngram_range':[(1,2),(1,3)],
                              'clf__estimator':[MultinomialNB()]
                                },
                                {
                              'vectorizer__vectorizer':[CountVectorizer()],
                              'vectorizer__vectorizer__min_df':[0.1,0.01,0.001,0.0001],
                              'vectorizer__vectorizer__ngram_range':[(1,2),(1,3)],
                              'clf__estimator':[MultinomialNB()]
                                }
                             ]
```

I have created a pipeline with parameters in the parameter grid. I have used KFold cross-validator to perform 5 cross-validations and shuffling the data each time. I fitted the pipeline on cleaned comment text from training data.

```
cv = KFold(n_splits=5, random_state=42, shuffle=True)

# Gridsearch with count vectorizer and TFIDF
gs_CompareVectorizer = GridSearchCV(pipe_CompareVectorizer, param_grid=paramGrid_CompareVectorizer,n_jobs=-1,
                                    cv=cv, verbose=0,return_train_score=True,scoring='roc_auc')

gs_CompareVectorizer.fit(X_train['CleanWordList'],y_train);


gs_CompareVectorizer.best_params_['vectorizer__vectorizer']
```

```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
                lowercase=True, max_df=1.0, max_features=None, min_df=0.0001,
                ngram_range=(1, 2), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

It is interesting to find that CountVectorizer outperformed TF-IDF. The selected value for min_df is 0.0001, i.e 0.01%, and ngram_range=(1,2).

Select best estimator for Machine Learning

KFold Cross-validator

I used K-Fold cross-validator from sklearn's model selection module. K-fold cross validator provides train/test indices to split data in train and test sets. I also set shuffle to True to shuffle the data before splitting into batches. Shuffling data helps create a better split of the dataset when it is sorted by any feature. Each fold is then used once as a validation while the k - 1 remaining folds form the training set.

```
# Initialize Kflod for cross validation
cv = KFold(n_splits=5, random_state=42, shuffle=True)
```

Scoring metric for grid searching:

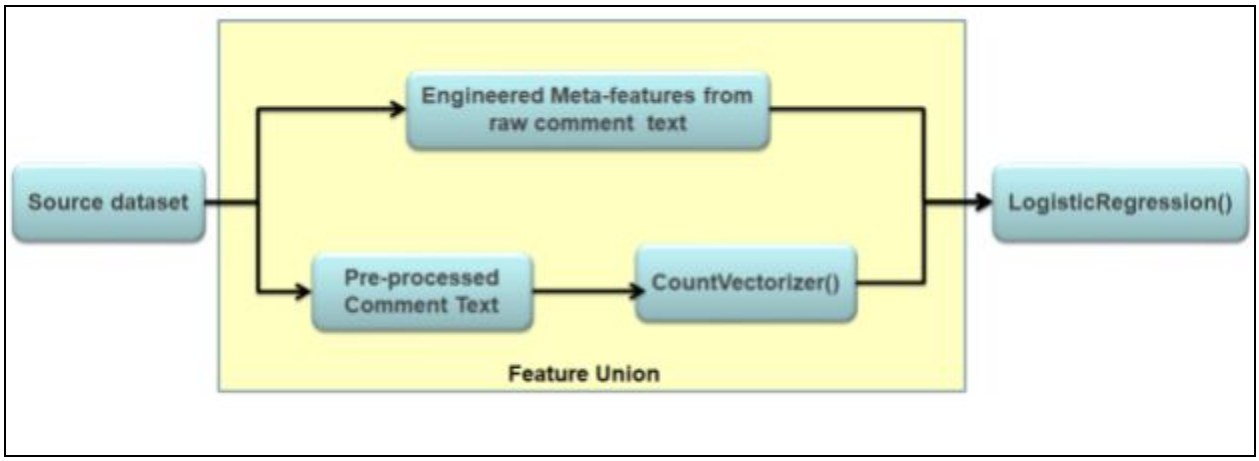
The Receiver Operator Characteristic (ROC) curve is used to evaluate binary classification problems. It is a probability curve that plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold values to see how well the model can separate the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

The higher the AUC, the better the performance of the model at distinguishing between the two classes across probability thresholds and different business scenarios. As such, I'll use the AUC metric to choose the best model.

Model Selection

Using the vectorizer selected from the previous grid search, I grid searched 4 different models for hyper parameter tuning and compared them with the ROC_AUC score - as this metric is threshold independent and therefore gives a general idea of how well the model performs across different business scenarios.

I combined meta features generated during the feature engineering step with text features generated from countvectorizer using the FeatureUnion function of Pipeline.

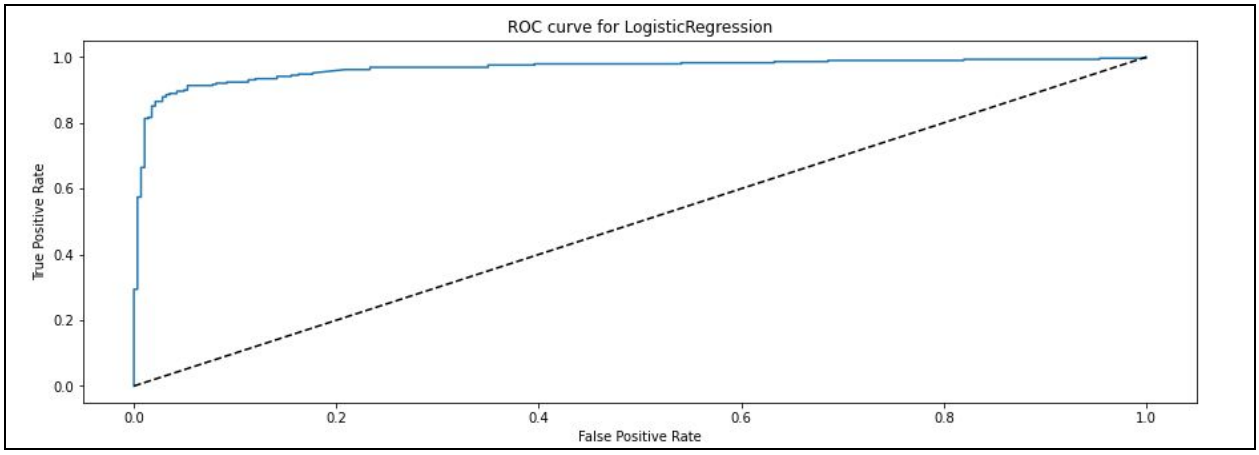



```
get_text_data = FunctionTransformer(lambda x: x['CleanWordList'], validate=False)
get_numeric_data = FunctionTransformer(lambda x: x[['NoOfUpperCaseLetters',
                                                    'NoOfURL',
                                                    'NoOfWords',
                                                    'AvgSentenceLength',
                                                    'TextStandard']], validate=False)

# Pipepline with FeatureUnion. Feature Union joins the text features and meta features
# generated during Feature engineering step.
pipe = Pipeline([('features', FeatureUnion([
                                            ('meta_features', Pipeline([('selector', get_numeric_data)])),
                                            ('text_features', Pipeline([('selector', get_text_data),
                                                                        ('vectorizer', vectorizer)]))
                                            ])),
                 ('clf', clf)
                ])

gs = RandomizedSearchCV(pipe,param_grid,n_jobs=-1, cv=cv, verbose=0,return_train_score=True,scoring='roc_auc')
gs.fit(X,y);
```

List of meta-features used in to join with text features

After hyperparameter tuning, Logistic Regression Classifier performed the best by a small margin.

	Best Params	Best ROC_AUC Score
Logistic Regression	{'clf__C': 2}	0.971756
Multinomial Naive Bayes	{'clf__fit_prior': False, 'clf__alpha': 2}	0.963609
Random Forest	{'clf__n_estimators': 100, 'clf__max_depth': 50}	0.969934
SVC	{'clf__C': 500}	0.970415



ROC curve represents the ratio of true-positive rate against false-positive rate for a range of threshold. The true-positive rate is the proportion of all Spam records correctly classified as Spam. Similarly, the false-positive rate is the proportion of Ham records incorrectly classified as Spam.

As a rule of thumb, the more the curve is closer to the top-left corner better the performance of the model.

Metric Selection

Choosing which metric to optimize is essentially a business decision and it varies for different scenarios. There are 3 important metrics.

1. **Precision**: Out of all the comments flagged as Spam, what fraction are actually Spam.
2. **Recall**: Out of all the Spam comments, what fraction got flagged as Spam.
3. **F1 Score:** It combines the first two metrics using harmonic mean. It gives equal weighting to precision and recall.

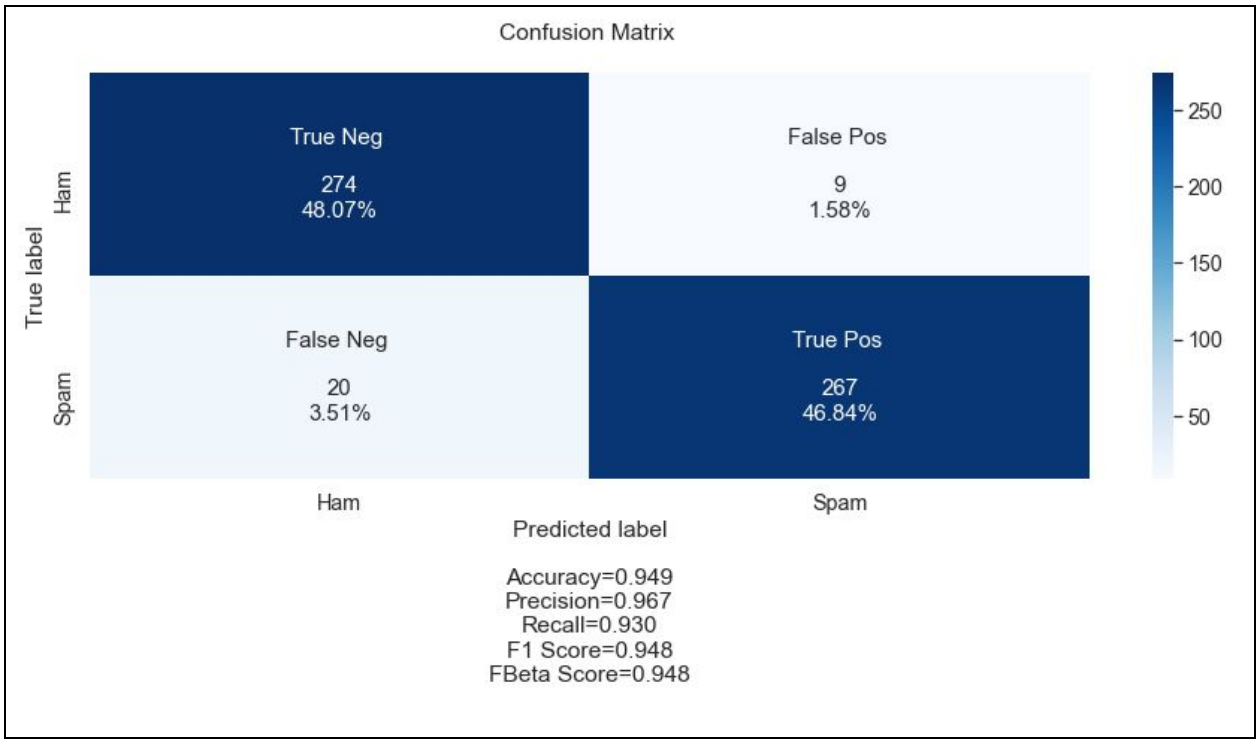
Precision is a metric that calculates the percentage of correct predictions for the positive class, while recall calculates the percentage of correct predictions for the positive class out of all positive predictions.

Precision is a good measure to determine when the costs of False Positives are high. For instance, email Spam detection. In email Spam detection, a false positive means that an email that is non-Spam (actual negative) has been identified as Spam (predicted Spam). The email user might lose important emails if the precision is not high for the Spam detection model. On the other hand, in case of cancerous cell detection problems the opposite is true. The goal in this case will be to have the lowest possible False-Negatives (Cancerous cell mis-classified as non-cancerous). An increase in False-Positive will be acceptable in this case if it helps reduce no. of False-Negatives.

F1 Score is a better measure to use if we are seeking a balance between precision and recall and there is an imbalanced class distribution. For this project, I am more inclined to achieve better precision than recall. I used the Fbeta score as it allows to adjust the value of β to give more weight to precision or recall. $\beta < 1$ gives more weight to precision, while $\beta > 1$ favors recall.

Threshold Optimization

Once the metric is chosen, it is optimized with various threshold probabilities. The default threshold for the model is 0.5, i.e any document having *predict_proba* <0.5 is classified as Ham and Spam otherwise. Using 0.5 as threshold, the current model has 1.68% False Positives, (i.e. 1.68% of total comments were misclassified as Spam when actually they were non-Spam comments) whereas, 5.45% of total comments were False Negatives (classified as Ham when they were actually Spam).

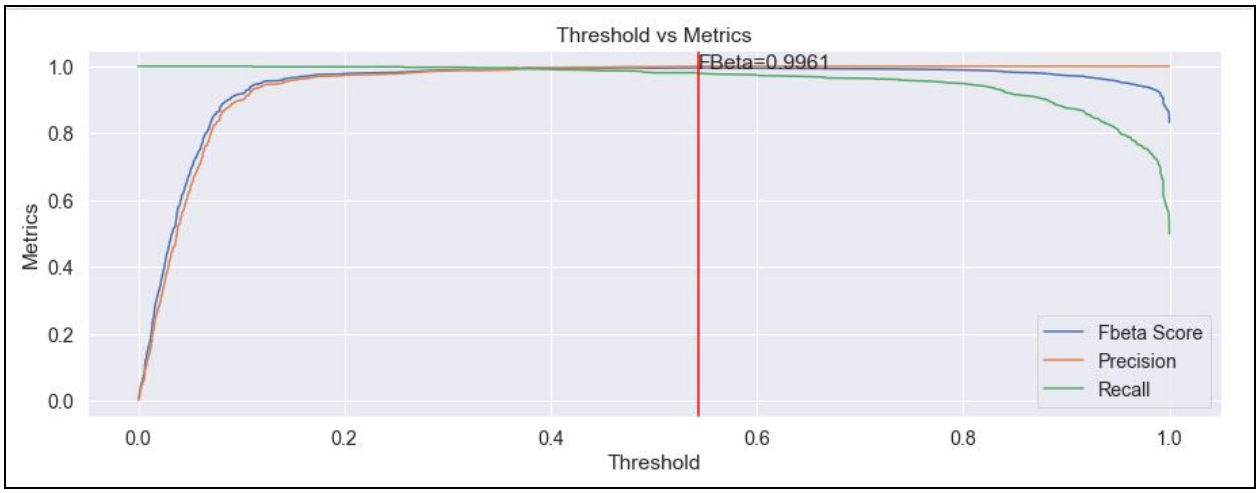


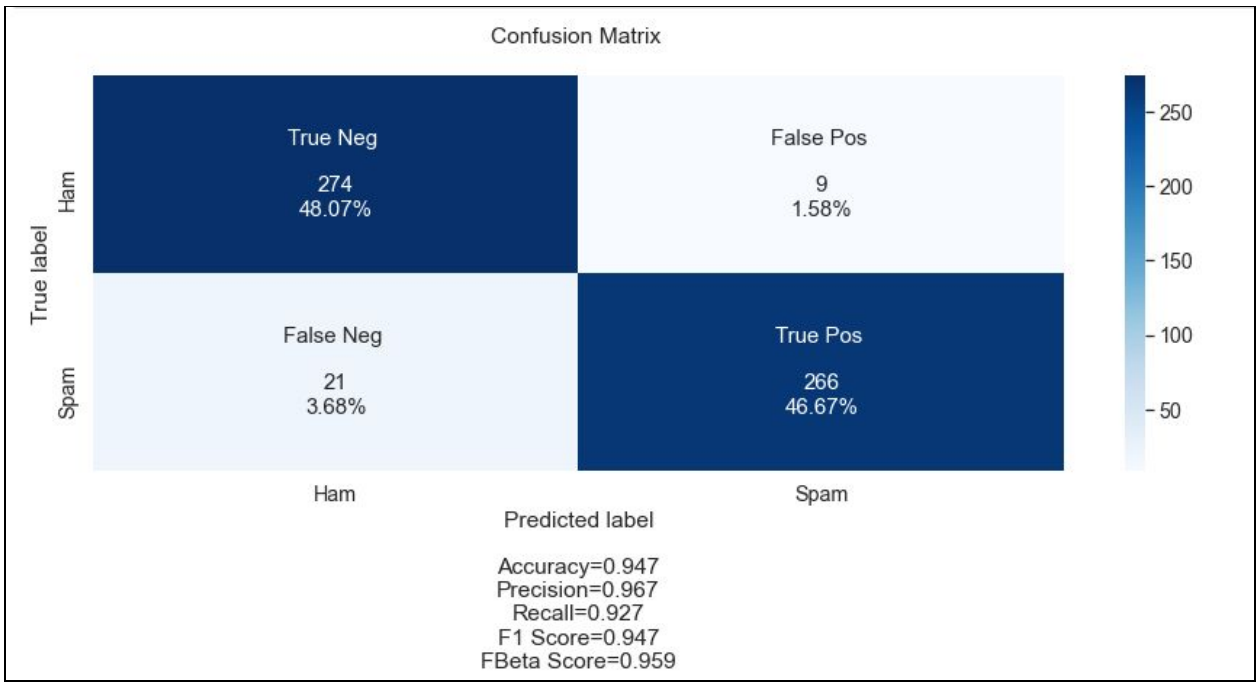
Confusion matrix with default threshold of 0.5

I chose 0.5 to set as β as this scenario gives double weight to precision over recall and searched for the optimal threshold that maximized Fbeta score. I got the highest Fbeta score at threshold =0.43.

	Threshold	Precision	Recall	F1	FBeta	Accuracy	Beta
0	0.543306	1	0.980712	0.990262	0.996082	0.990211	0.5
1	0.475235	0.998487	0.986547	0.992481	0.996076	0.99247	0.5
2	0.544519	1	0.979259	0.989521	0.995782	0.989458	0.5
3	0.486523	0.998487	0.985075	0.991736	0.995775	0.991717	0.5
4	0.550045	1	0.977811	0.988781	0.995482	0.988705	0.5

Here is the plot of threshold vs precision, recall, and Fbeta score at β=0.5.





Confusion matrix with threshold=0.543 and β=0.5

Classification Report

	precision	recall	f1-score	support
0	0.90	0.98	0.94	236
1	0.98	0.89	0.93	241
accuracy			0.93	477
macro avg	0.94	0.93	0.93	477
weighted avg	0.94	0.93	0.93	477

Classification report with threshold=0.6

Analysis of misclassified comments

False Positives

	Actually a HAM but wrongly calssified as SPAM	predict_proba
0	If you pause at 1:39 at the last millisecond you can see that that chick is about to laugh. Takes a few tries.	0.520534
1	THUMBS UP FOR ROBO GUY BABY	0.576717
2	I dont even watch it anymore i just come here to check on 2 Billion or not	0.59258
3	Lemme Top Comments Please!!	0.693893
4	This comment will randomly get lot's of likes and replies for no reason. I also like Jello. Strawberry jello.	0.767307
5	OMG I LOVE YOU KATY PARRY YOUR SONGS ROCK!!!!!!!!!!!!!!!! THATS A TOTAL SUBSCRIBE	0.828406
6	i check back often to help reach 2x10^9 views and I avoid watching Baby	0.934166
7	thumbs up if u checked this video to see hw views it got	0.986313
8	My honest opinion. It's a very mediocre song. Nothing unique or special about her music, lyrics or voice. Nothing memorable like Billie Jean or Beat It. Before her millions of fans reply with hate comments, i know this is a democracy and people are free to see what they want. But then don't I have the right to express my opinion? Please don't reply with dumb comments lie "if you don't like it don't watch it". I just came here to see what's the buzz about(661 million views??) and didn't like what i saw. OK?	0.997567

List of False-Positives (Ham comments misclassified as Spam)

Findings:

1. As expected from our previous analysis, Ham comments that contain words that were among our “Most predictive words” - subscribe, check, check channel, check video, comment - are likely to be misclassified as Spam.

2. These are the comments when the model strongly thought that the comments were Spam but the model was wrong. For most of these comments, even though they are labelled as Ham, they appear very similar to Spam comments such that I would’ve

probably thought they were Spam as well. As such, this doesn't seem to be particularly problematic although perhaps with more training data the ML model could pick out patterns that a human could not. However, since these were manually labelled it's possible that they actually are Spam and were just mislabelled by the human classifiers as well. The one exception is last comment which appears to simply use the words "comment" and "reply" frequently which explains the misclassification.

False-Negatives

	Actually a SPAM but wrongly calssified as HAM	predict_proba
0	Yea stil the best WK song ever Thumbs up of you think the same	0.0266963
1	1 753 682 421 GANGNAM STYLE ^^	0.0276298
2	Believe that Jesus Christ is your savior for all your sins. If you truly believe in Jesus Christ to be your savior for all your sins then you will go to Heaven. If you believe in Jesus Christ then you are saved and you are in salvation and you have gained God's righteousness. It matters not how much you have sinned in the past, in the present and especially in the future. Believe that Jesus Christ is your savior and you will go to Heaven forever and that is the whole truth. Spread the truth.	0.0513756
3	los invito a subscribirse a mi canal	0.0826704
4	I subscribed it	0.0833285
5	Incmedia.org where the truth meets you.	0.0861075
6	Aslamu Lykum... From Pakistan	0.097072
7	Help shakira's waka waka be the first song by a female artist to reach 1 billion views. (dark horse is ahead by roughly 100 million more views, and roar has only 50 million more views) https://www.youtube.com/watch?v=pRpeEdMmmQ0	0.130504
8	This guy win dollars sleeping... m m m he loves the planet its full of RETARDS	0.147522
9	WOW muslims are really egoistic..... 23% of the World population and not in this video or donating 1 dollar to the poor ones in Africa :(shame on those terrorist muslims	0.153298

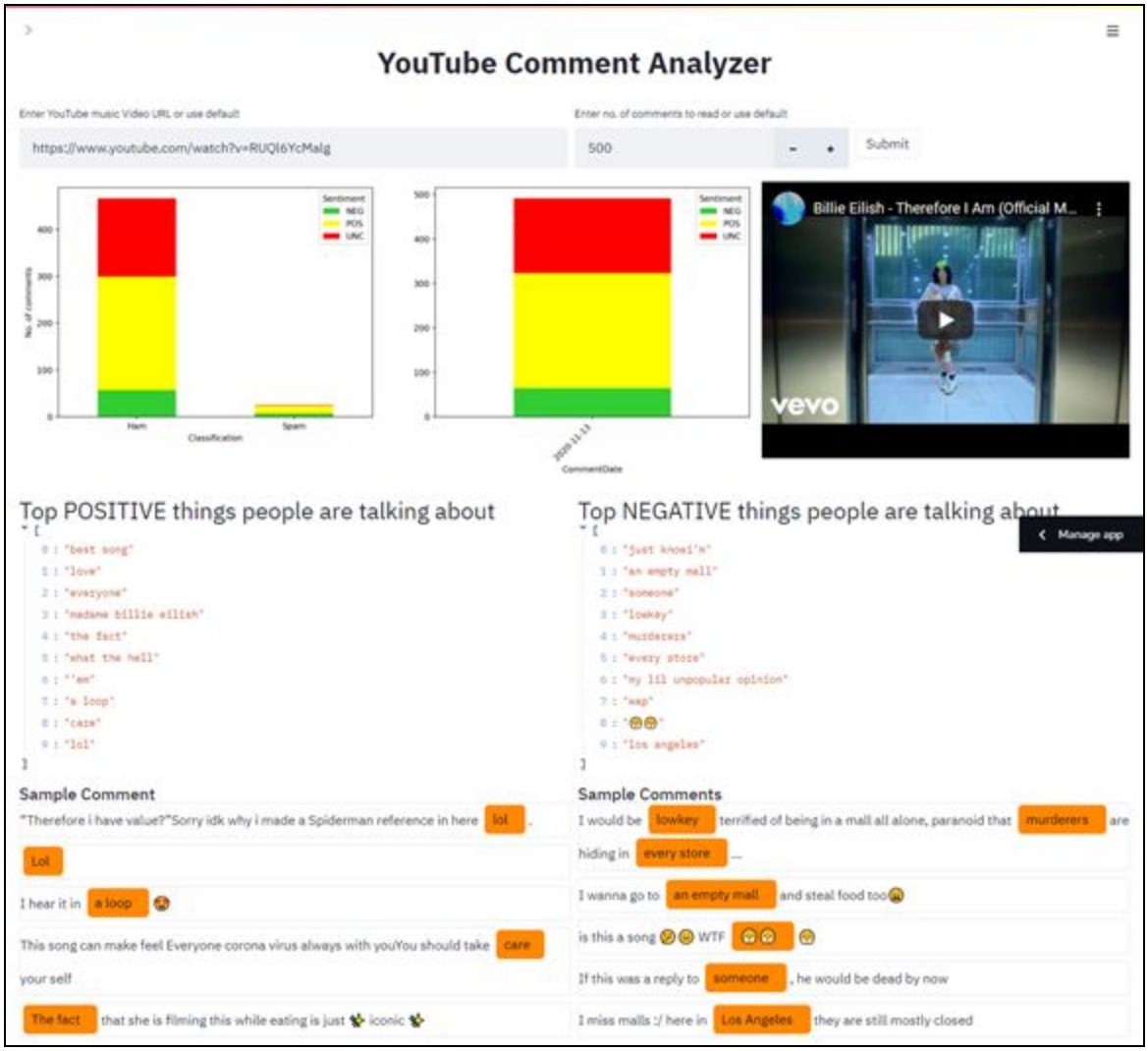
List of False- Negative (Spam comments misclassified as Ham)

Findings:

1. These Spam comments are carefully curated so that they can pass through Spam filters. Most of these comments do not have any word from the list of most predictive words for Spam comments. Number 7 does contain a youtube link as well as the word "views" - one we know to be a predictor of spam - 3 times. This may indicate or model could still be improved, and perhaps doesn't do as well on longer messages as shorter messages. This would require further analysis.

App for user Interface

I created an app using the streamlit library that scrapes comments from YouTube using the YouTube API and classifies them into positive/negative and Spam/non-Spam categories. Since the training data is of music videos, the app gives the best results for other music videos. The app also shows top key phrases from positive and negative comments and 5 sample comments containing those phrases.



Summary

The comments section on Youtube offers a golden opportunity for higher level analytics to help content creators and users better understand video content and viewer reaction. This functionality is not available in YouTube studio as of yet, and this project attempted to bridge this gap by providing insight into the content of the comments. Along with classifying the comments into two categories, I also explored what words were most predictive for each category, providing further insight into what features most identified a comment as Spam or Ham. After tuning the model, I found that misclassified Ham comments seemed very similar to spam comments - indicating the model performs surprisingly well - perhaps on par, or at least in the same ballpark, as human analysis. In fact, the dataset used for this project is manually labelled and it's possible that a few of these "misclassifications'' may have been mislabelled by mistake.

As a next step, I'd like to build a chrome extension to make the model and its results easily accessible. This model could also be improved by diversifying the training data - more data from other domains such as tech, fashion, news, etc. would help the model to make more accurate predictions on real world scenarios outside of music videos.