YouTube comments classification

Springboard Capstone - 2

Preeti Prajapati

Mentor: Benjamin Bell

Table of Content

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 comments. It only shows a list of comments, comments held for review and likely spam comments.

In this project I have tried to come up with an enhanced version of analysis on comments which can be useful to the channel owner 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

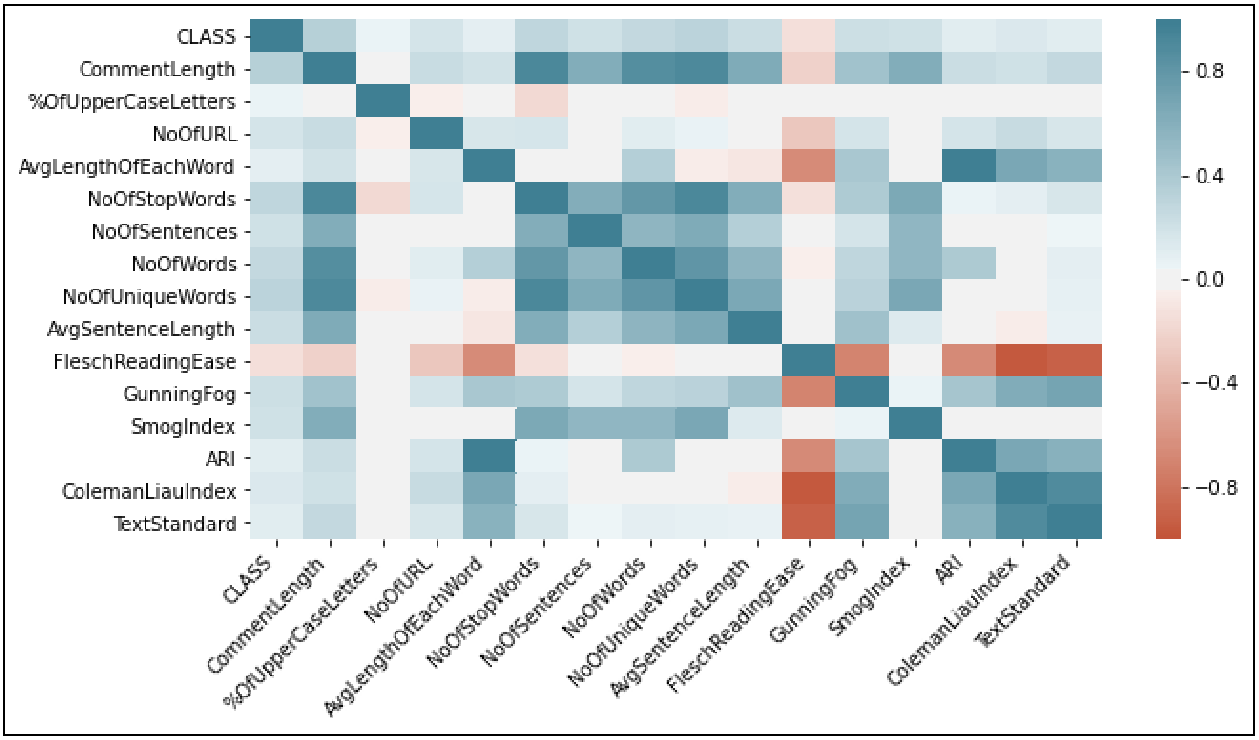
Distribution of target in training data

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.

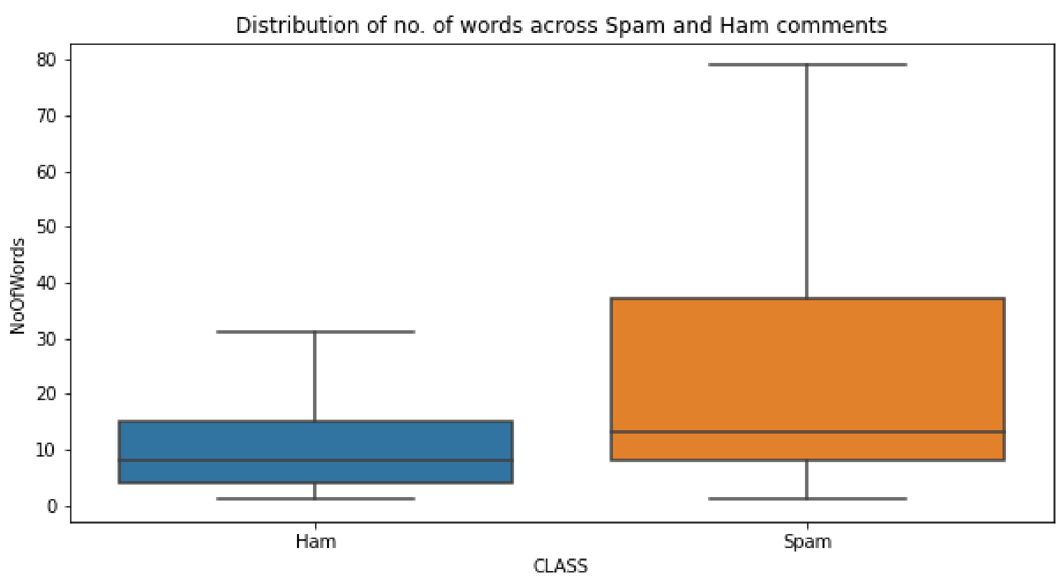
I replaced counts of stop words and unique words to make them independent of comment length and dropped comment length field.

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.

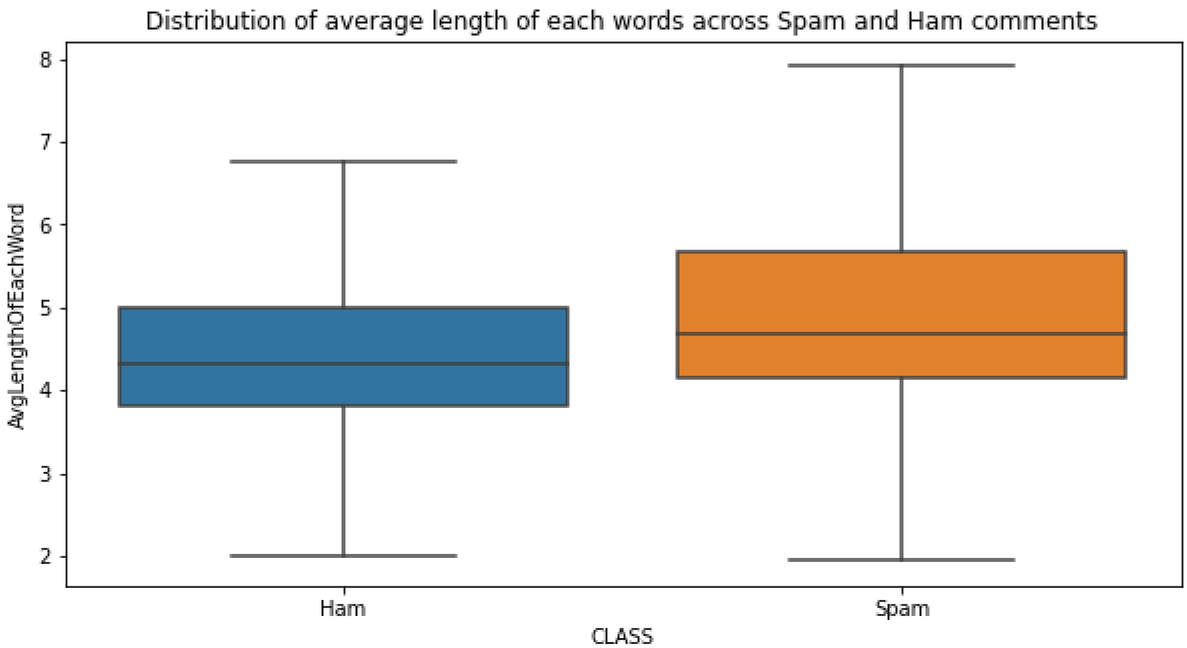
Hypothesis Testing: Are spam comments usually longer than non-spam comments?

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.

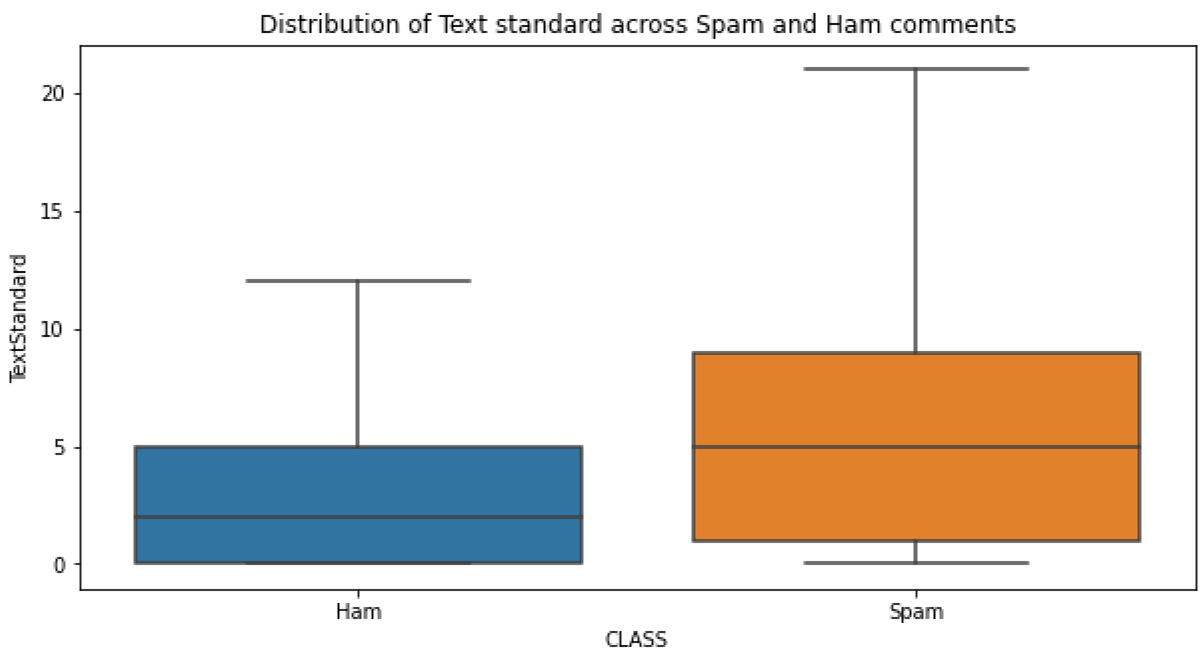
Hypothesis Testing: Do spam comments have longer words than non-spam comments?

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**

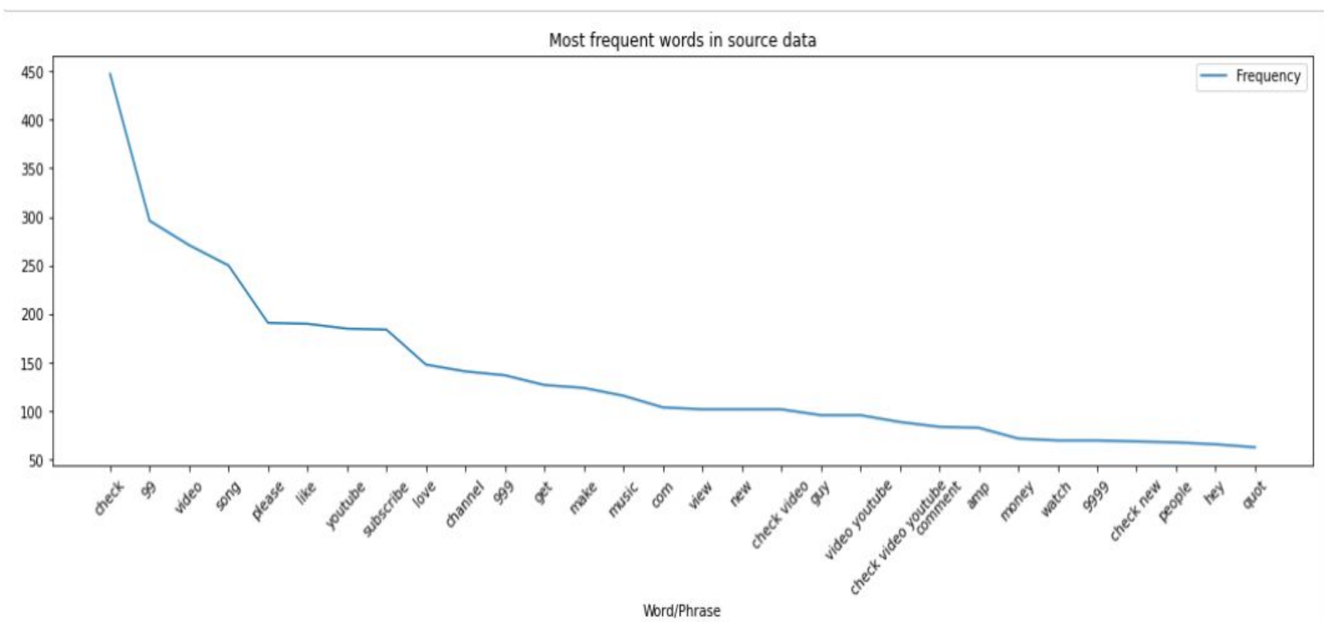
Data Cleaning

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 frequent words

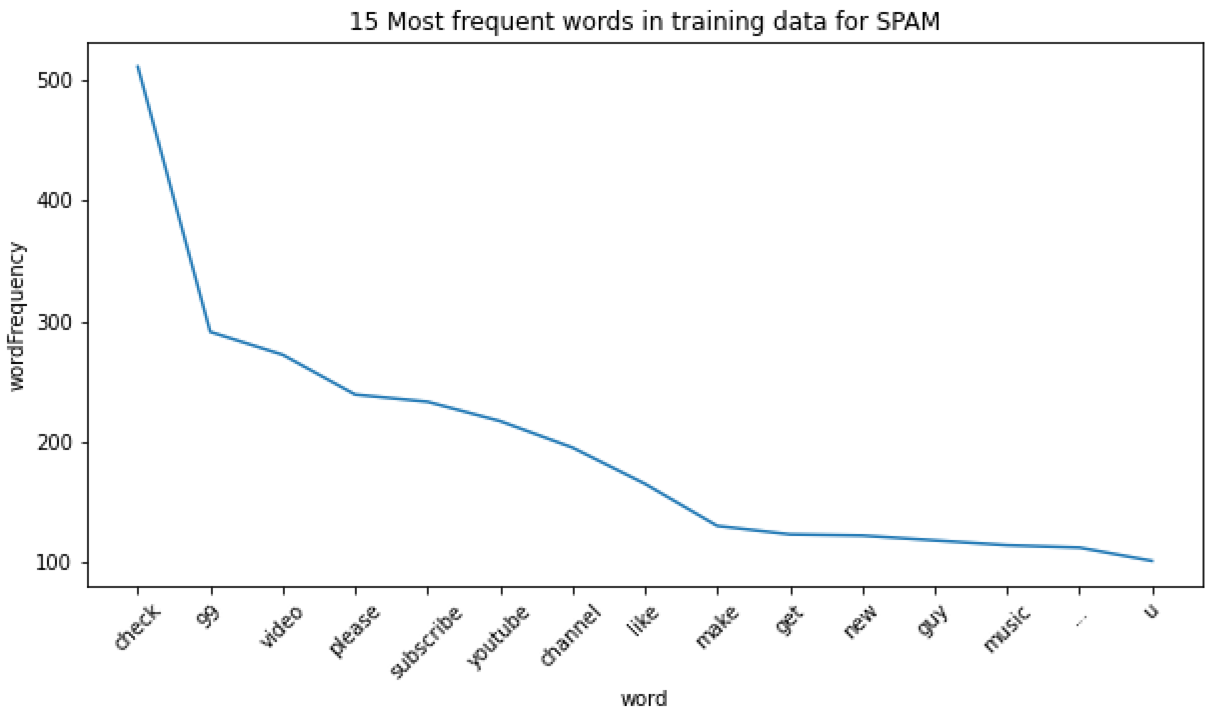
1. **Most frequent words in all of the training data:**



Findings:

1. Here 9 and 99 represent 1 and 2 digit numbers respectively.
2. From the list of most frequent words, it looks like most of the comments have sentences like “Please subscribe to my youtube channel” or “Please like my song/video”.

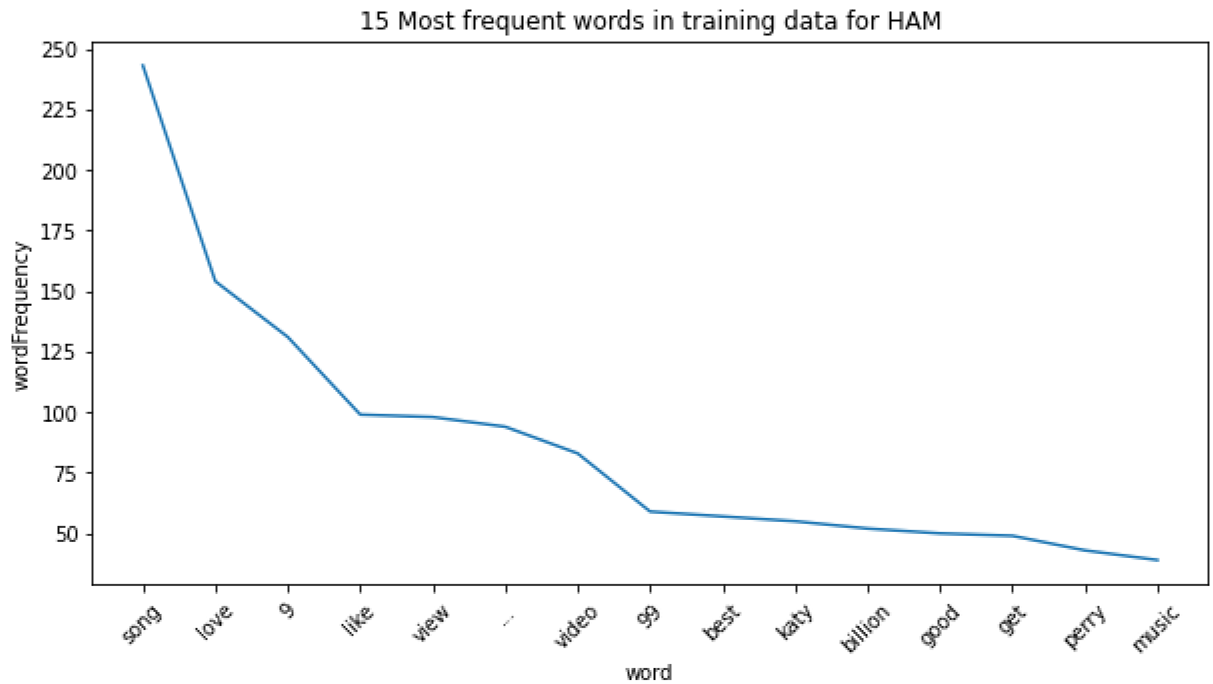
2. Most frequent words in training data for SPAM :



Findings:

1. It appears that most of the Spam comments have sentences like “Please like/check/subscribe to my youtube channel” or “please like my video”.
2. 2 digit numbers are the second most frequent word. This can be investigated further to know what this number represents.

3. **Most frequent words in training data for NON_SPAM comments:**



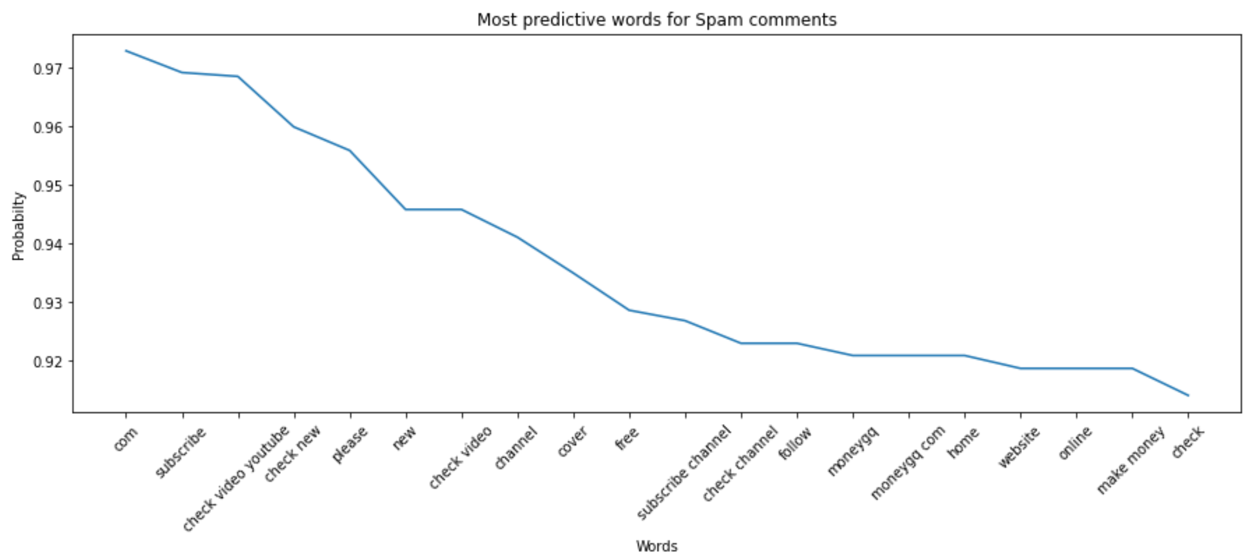
Findings:

1. There are no sentences that stand-out from the list of most frequent words in non-spam comments.
2. Like spam comments, non-spam comments also have 2 digit numbers in their most frequent words list.
3. This training data is about video songs from 5 artists, so it makes sense why the word "song" is the most frequent word.
4. Apart from Katy Perry, we don't see any other artist's name in this list.

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.

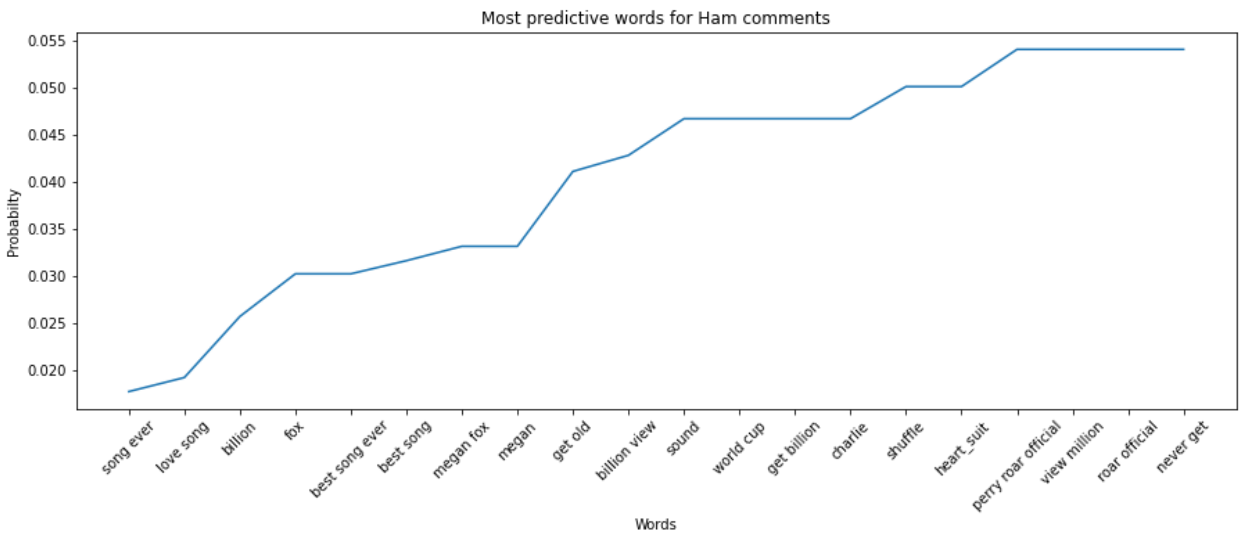
Here are a few comments with words related to making free money from home and predicted as SPAM.

```
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.
3. 'World Cup' is also one of the predictors of non-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.

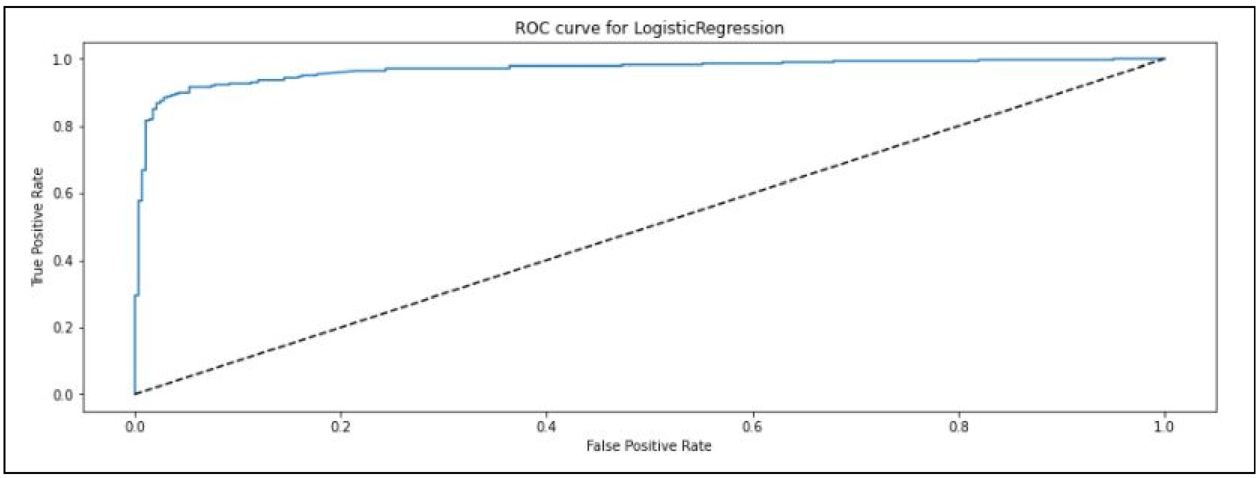
The higher the AUC, the better the performance of the model at distinguishing between the two classes across probability thresholds and different business scenarios. As such, I'll use the AUC metric to choose the best model.

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.

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 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 weightage to precision and recall.

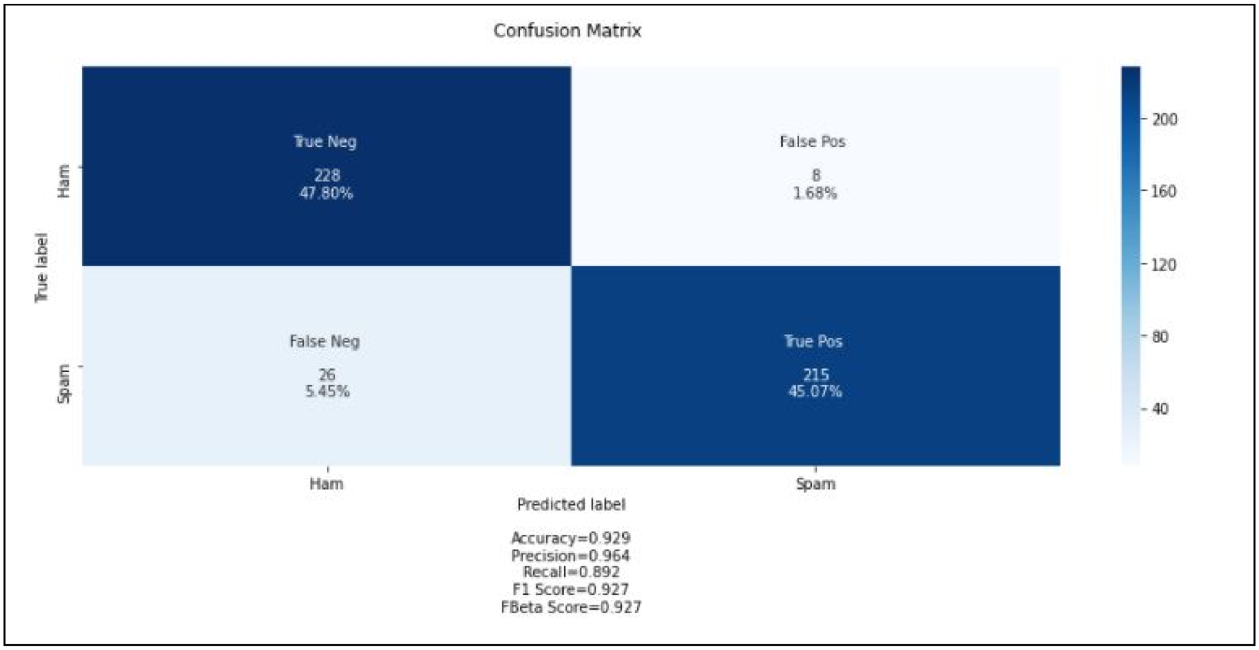
Precision is a metric that calculates the percentage of correct predictions for the positive class, while recall calculates the percentage of correct predictions for the positive class out of all positive predictions.

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.

Adjust Threshold

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 Positive,i.e. 1.68% of total comments were misclassified as Spam when actually they were non-spam comments. Whereas, 5.45% of total comments are classified as Ham when they were actually spam.

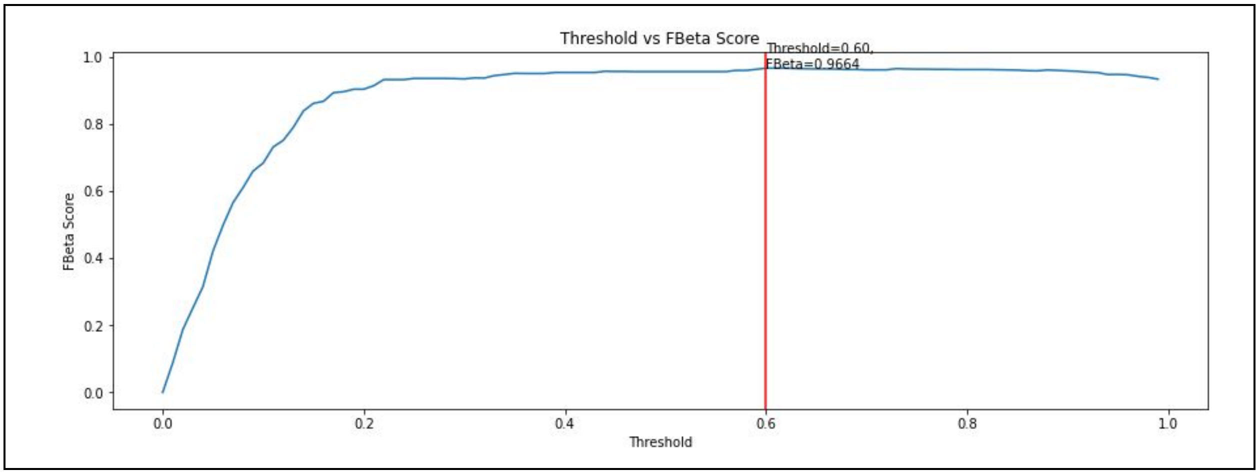


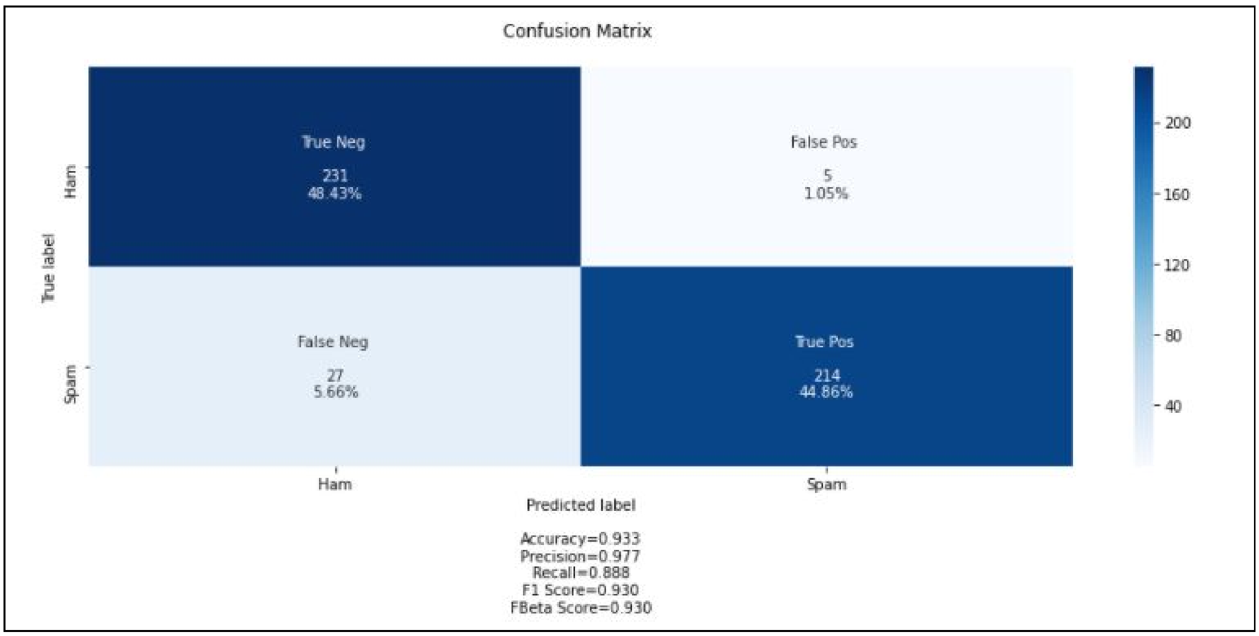
Confusion matrix with default threshold of 0.5

I chose a number of values less than 1 to set as β as this scenario requires more weight on precision and searched for the optimal threshold that maximizes Fbeta score. Among all the values of β, 0.4 gave the highest Fbeta Score with threshold =0.6. Another thing to notice here is the optimal value of the threshold is 0.6 for all the selected values of β.

	Beta	Threshold	FBeta Score
0	0.4	0.6	0.966388
1	0.5	0.6	0.960899
2	0.6	0.6	0.955242
3	0.7	0.6	0.949699
4	0.8	0.6	0.944456
5	0.99	0.6	0.935641

Here is the plot of threshold vs Fbeta score for β=0.4





Confusion matrix with threshold=0.6 and β=0.5

Adjusting the threshold to 0.6 from default of 0.5 reduced the false-positives by 0.68% as well as improved all of the metrics.

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

	CONTENT
3	OMG I LOVE YOU KATY PARRY YOUR SONGS ROCK!!!!!!!!!!!!!!!!!!! THATS A TOTAL SUBSCRIBE
5	If you pause at 1:39 at the last millisecond you can see that that chick is about to laugh. Takes a few tries.
1	This comment will randomly get lot's of likes and replies for no reason. I also like Jello. Strawberry jello.

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

Findings:

1. These are the comments where the model was not too confident about the classification but still classified it as Spam because predict_proba was above the threshold.
2. We saw during analysis of “Most predictive words” that subscribe, check, check channel, check video,comment were among the most predictive words for Spam comments. In the misclassified ham comments, we can see those words are prevalent and hence the reason for misclassification.

Extreme False-Positives

	CONTENT
0	thumbs up if u checked this video to see hw views it got
1	i check back often to help reach 2x10^9 views and I avoid watching Baby
2	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?
3	Lemme Top Comments Please!!

These are the comments when the model strongly thought that the comments were Spam but the model was wrong. This issue can be tackled by adding more training data. That way the machine learning model can learn when the same word makes a comment spam or ham.

False-Negatives:

	CONTENT
0	It's so hard, sad :(iThat little child Actor HWANG MINOO dancing very active child is suffering from brain tumor, only 6 month left for him .Hard to believe .. Keep praying everyone for our future superstar. #StrongLittlePsY #Fighting SHARE EVERYONE PRAYING FOR HIM http://ygunited.com/2014/11/08/little-psy-from-the-has-brain-tumor-6-months-left-to-live/
1	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
2	gofundme.com/grwmps
3	reminds me of this song https://soundcloud.com/popaegis/wrenn-almond-eyes
4	Limit sun exposure while driving. Eliminate the hassle of having to swing the car visor between the windshield and window. https://www.kickstarter.com/projects/733634264/visortwin

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

Findings:

1. These are the comments where the model was not too confident about the classification but classified them as Ham because predict_proba was below the threshold.
2. These spam comments are carefully curated so that they can pass through spam filters. None of these comments have any word from the list of most predictive words for spam comments.

Extreme False-Negatives

	CONTENT
0	🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨 NOW THAT I HAVE YOUR ATTENTION!!! 🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨SUBSCRIBE TO MY CHANNEL!!! 🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨 GRACIAS!💋 🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨
1	Haha , Miley Cyrus has done it once again Today someone leeched a porno video with her on a celeb site I believe the website link is : miley-celeb-news.co.uk in case you want to view it....
2	i love this so much. AND also I Generate Free Leads on Auto Pilot & You Can Too! http://www.MyLeaderGate.com/moretraffic

Like the solution for tackling false-positives, the issue of false-negative can also be tackled by adding more training data.

Summary:

1. Grid searched vectorizers and selected countvectorizer,
2. Grid searched 4 model, and did hyperparameter tuning. Logistic regression performed the best.
3. Chose Fbeta score to analyze the performance of logistic regression for various values of beta. Searched to find optimal value of threshold which maximizes the fbeta score.
4. Adjusted threshold improved the false-positives., i.e. improved the no. of non-spam comments categorized as spam.
5. Looked at the comments which were mis classified in both the classes with high probability.

Future Enhancement Recommendations

1. **Diversify training data**: The dataset used for this project is specific to music videos. In the future, more data from other domains such as tech, fashion, news, etc. can be added to training dataset. This will help the model to make more accurate predictions on real world scenarios.
2. **Topic Analysis:** A feature to show relevant topics emerging from the comments can be added. This can be done both on Spam and Ham comments and it will help the users to identify main talking points in the comments.

3. **App for user Interface:** An app can be developed to get predictions from the model in real time. This app will allow users to enter URL for any video and then it will show a list of non-spam comments and relevant topics in them.

   More analytics, like comments over time broken down by spam and ham, can also be added to the app to make it one stop shop for all the analytics needs.