YouTube Comments Classification

Springboard Capstone 2

Introduction

YouTube studio doesn’t show any analytics on content of the comments. With this project I aimed to fill this analytics gap.

In this project, 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.

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.

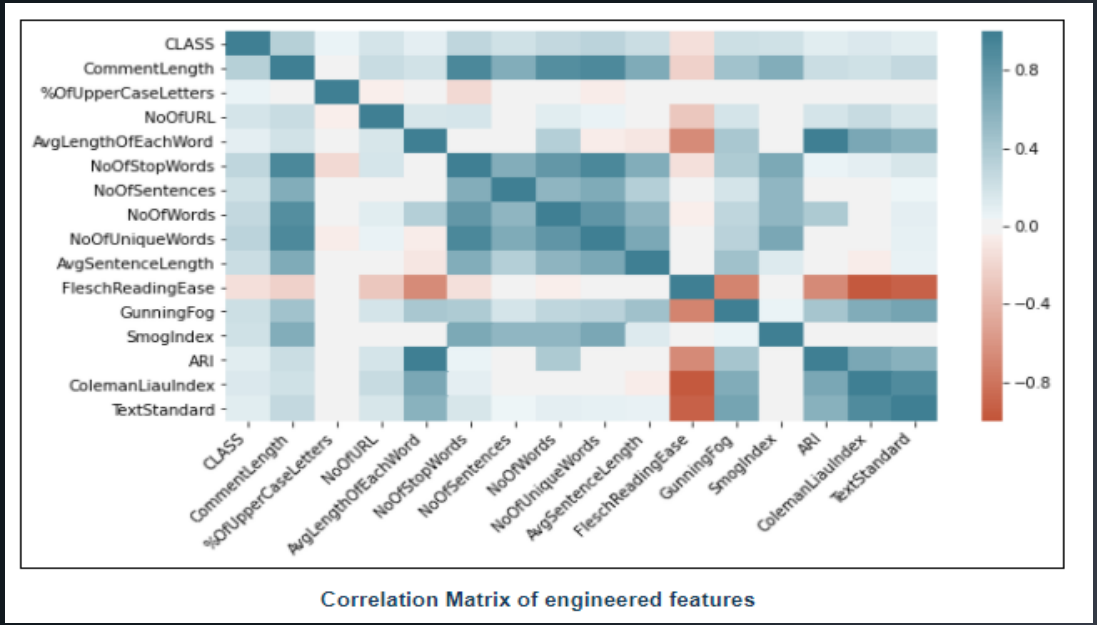
Data Source

UCI’s YouTube Spam Collection Data Set

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

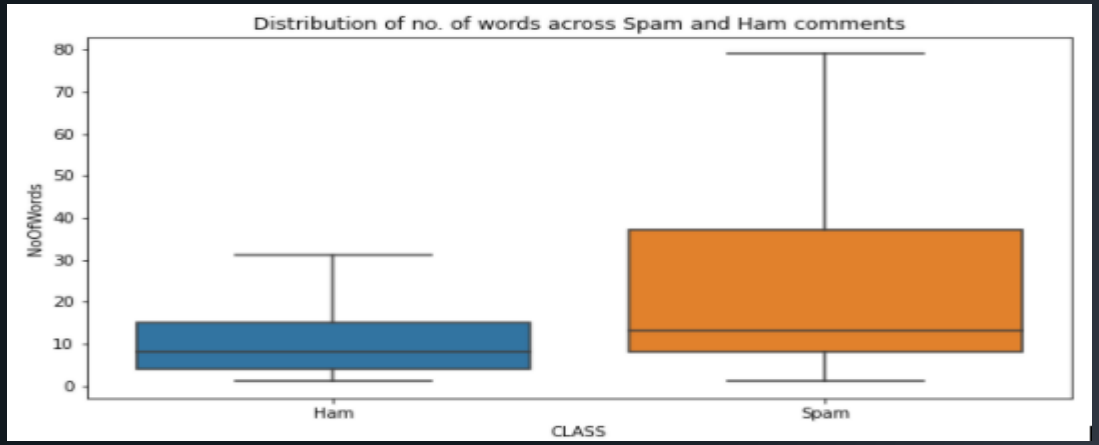
Exploratory Data Analysis

Correlation matrix



Correlation Matrix of engineered features

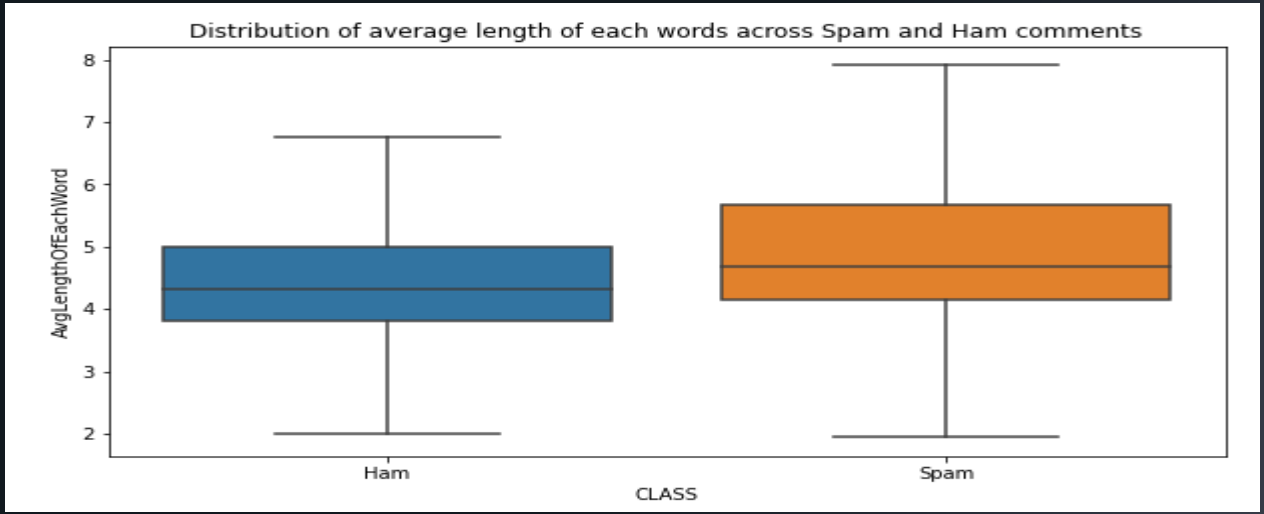
No. of words in each comment:



On average, spam comments are longer than ham comments.

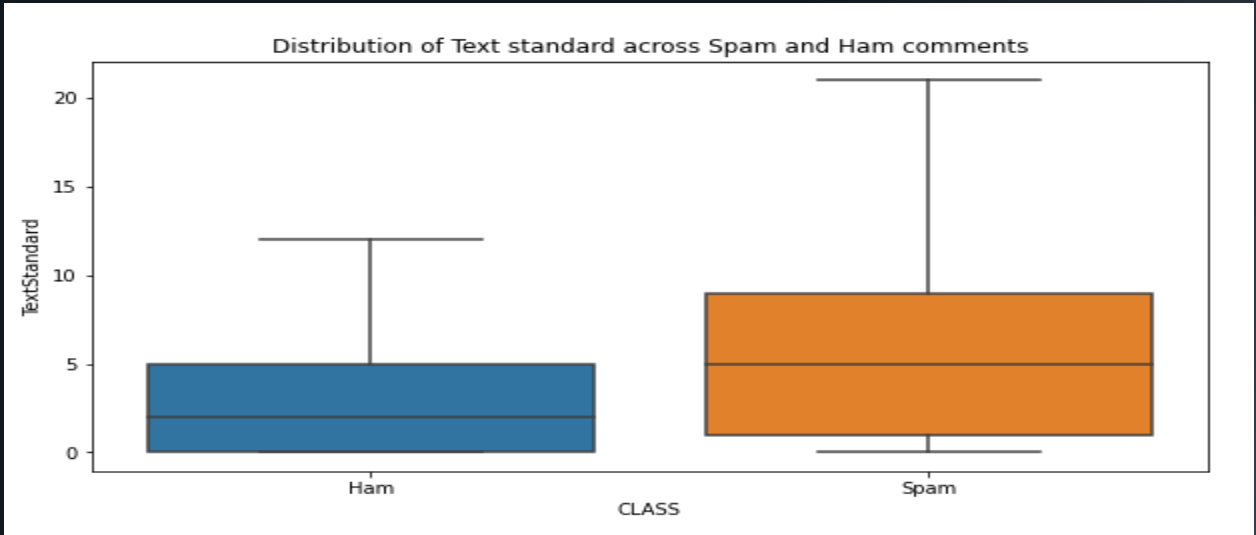
Non-spam comments are consistently shorter but Spam comments vary from short to long.

Average length of each word



SPAM comments have lengthier words than non-spam comments.

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.

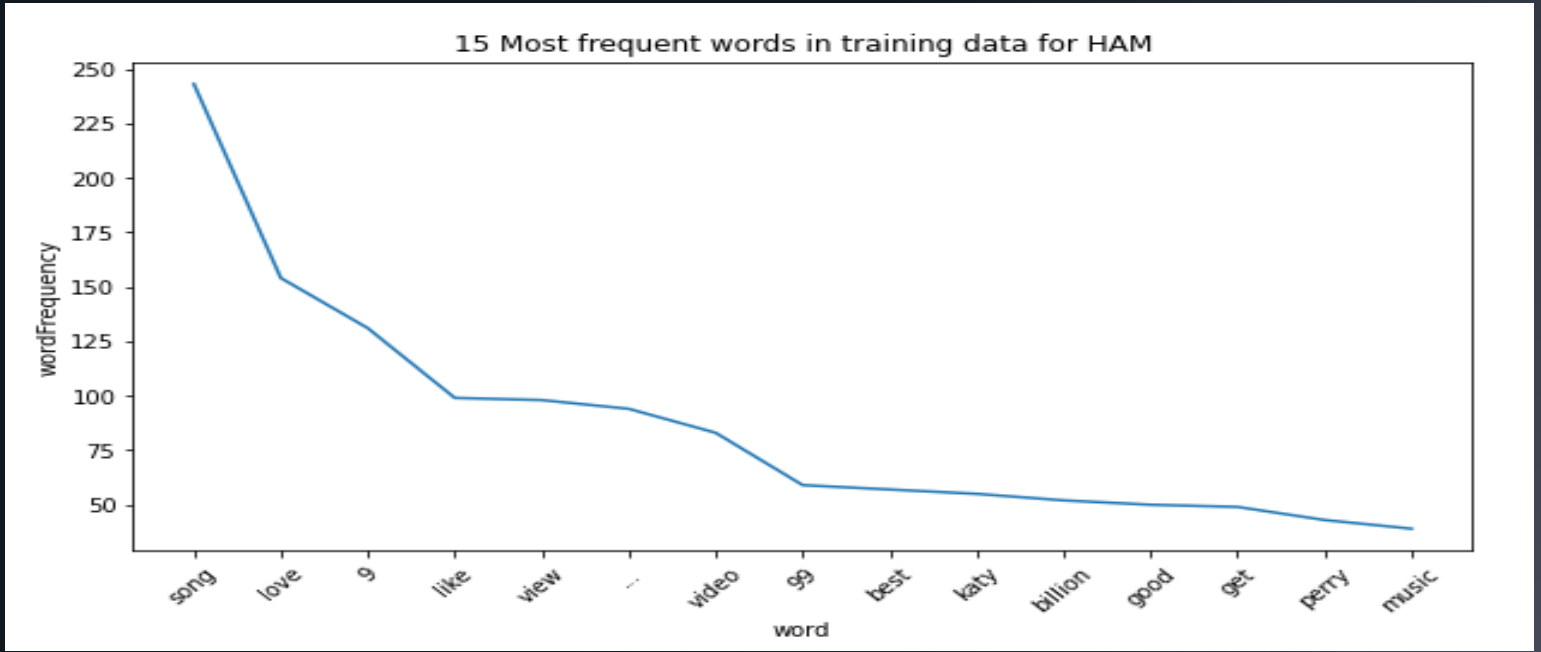
Hypothesis Testing

I performed hypothesis testing on below 3 questions.

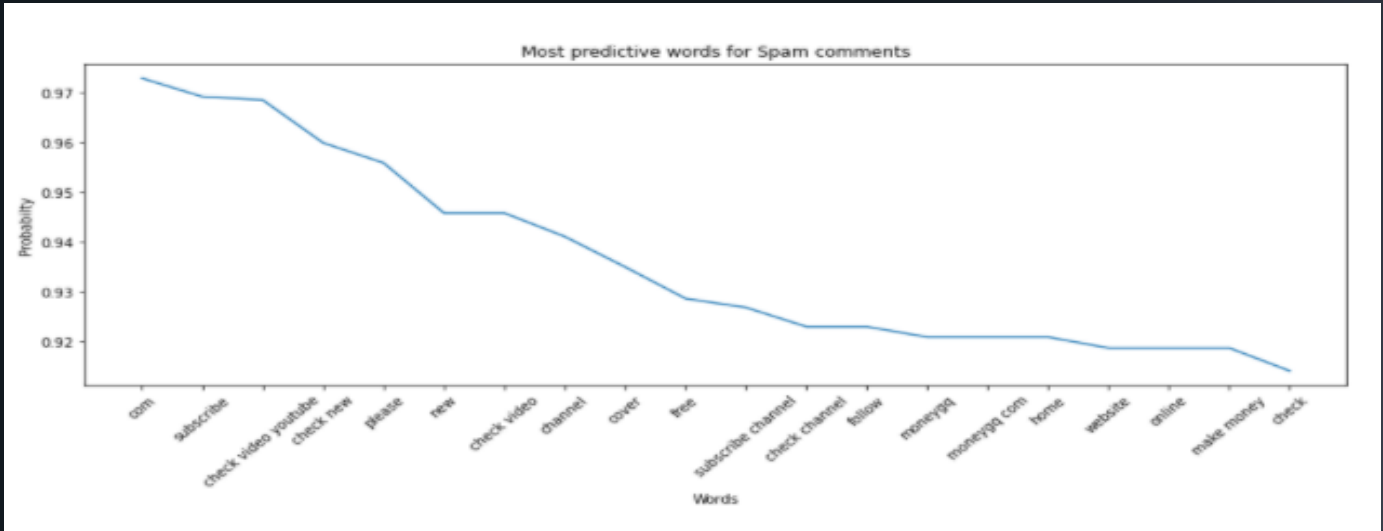
1. Are spam comments usually longer than non-spam comments?
2. Do spam comments have longer words than non-spam comments?
3. Do spam comments have a different text standard than non-spam comments?

Result: p_value is close to 0 for all the 3 cases and hence we reject the null hypothesis and can say that there is a difference in length , average word length and text standard spam and ham comments.

Most predictive words in Ham comments



Most predictive words in Spam comments



- Most of the spam comments have sentences like “Please like/ check/ subscribe/ follow to my youtube channel” or “please like my video” and are talking about making money.

Spam comments with money related words

visit " ww estiloproduction com " 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'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 --> http s://www.paidverts.com/ref/Marius1533

Data Wrangling

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 noise from the data.

Text Preprocessing

- Text data is contains various information that does not help the machine learning model. Stop words, spaces, URLs, etc. are few such piece of information.
- During the data preprocessing step, I removed all these texts by creating a function which takes the raw text and returns cleaned text fit for feeding into machine learning model.

Lemmatization

```
lemmatizer=WordNetLemmatizer()

word="slays"

print("As verb:",lemmatizer.lemmatize(word, 'v'))
print("As noun:",lemmatizer.lemmatize(word, 'n'))
```

```
As verb: slay
As noun: slays
```

- Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single term.
- Lemmatized as per part of speech of the word so the context of word is not lost.

Handling Emojis

```
text="ILOVE Katy Perry❤️❤️❤️❤️❤️❤️"
emoji.demojize(text)
```

```
'ILOVE Katy Perry:red_heart::red_heart::red_heart::red_heart::red_heart::red_heart:'
```

- I replaced the emojis with text using emoji library.

Machine Learning

Transforming text into something that can be consumed by machine learning model is called text-vectorization.

I tried two vectorizers from sklearn library.

1. Countvectorizer
2. TFIDFVectorizer.

```
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()]
                                }
                             ]
```

CountVectorizer outperformed TF-IDF with min_df =0.0001, i.e 0.01% and ngram_range=(1,2).

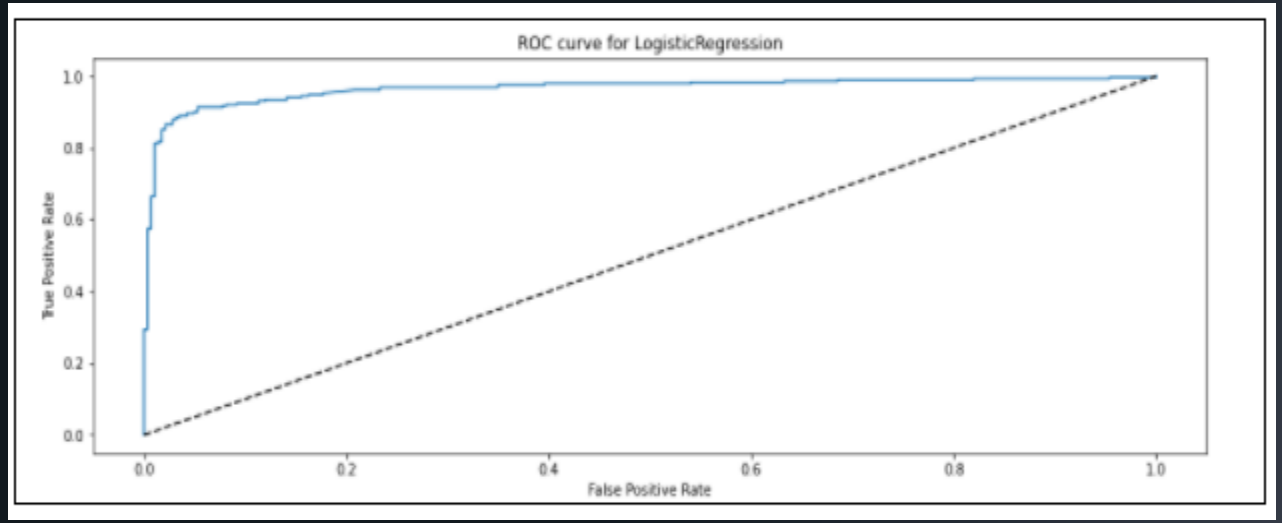
Classifier

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.

	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

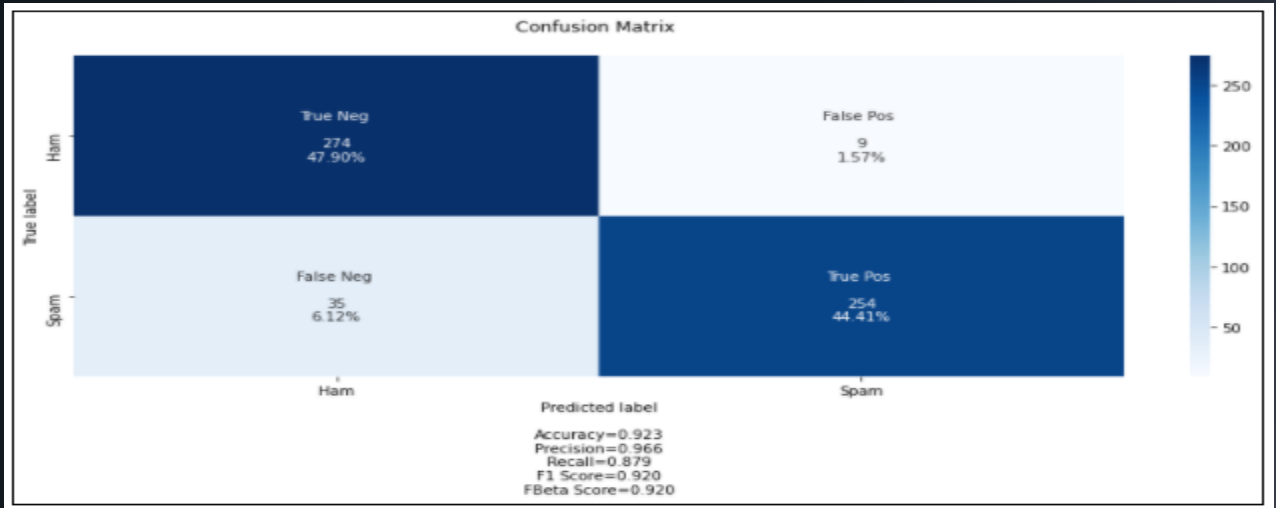
Logistic regression got selected as best performing model.

ROC curve



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.

Confusion Matrix



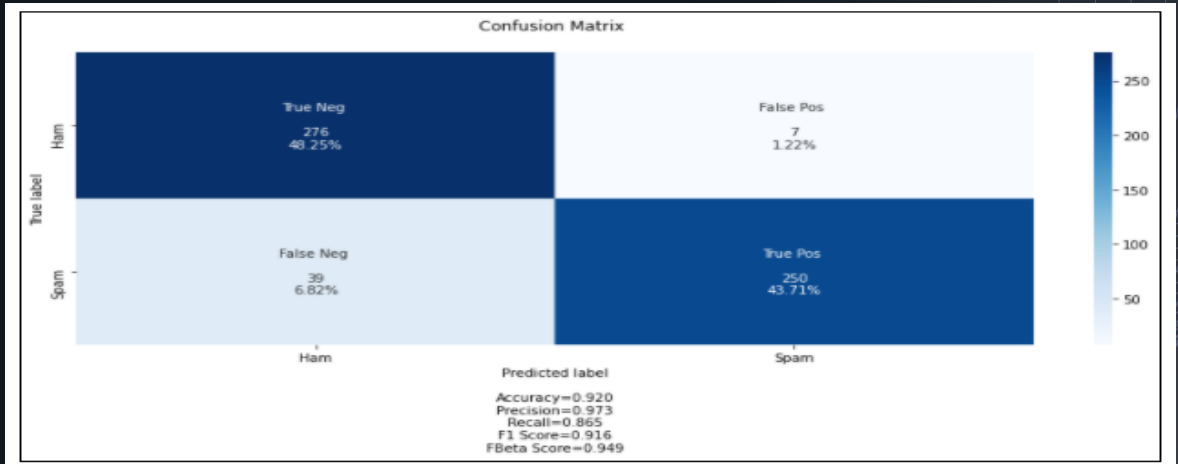
Confusion matrix with default threshold of 0.5

Adjust threshold

I chose 0.5 to set as β as this scenario requires more weight on precision and searched for the optimal threshold that maximizes Fbeta score. I got the highest FBeta score at threshold =0.43.

	Threshold	Precision	Recall	F1	FBeta	Accuracy	Beta
0	0.426372	1	0.992492	0.996232	0.998489	0.996252	0.5
1	0.45659	1	0.991004	0.995482	0.998188	0.995502	0.5
2	0.501593	1	0.989521	0.994733	0.997886	0.994753	0.5
3	0.505255	1	0.988042	0.993985	0.997585	0.994003	0.5
4	0.358747	0.998487	0.993976	0.996226	0.997582	0.996252	0.5

Confusion matric with adjusted threshold



Confusion matrix with threshold=0.43 and β=0.5

Future Steps

- Diversify training data
- Topic Analysis
- App for user Interface