Spring Board Capstone Project

Building A Spam Filter With Machine Learning

By: Zach Nguyen

Abstract

This paper aims to build a spam filter with a built in classification algorithm capable of distinguishing spams from ordinary text messages. Using a labeled corpus of text messages, the paper outlines the steps to pre-process and explore the data given, then proceed to run the core model to classify text messages as either "spam" or "ham" with 99.7% accuracy. Finally, a spam filter is made to deploy the algorithm to new messages.

Part I: Proposal

Problem: As technology improves at exponential rates, everything else improves with it. We seem to enjoy better lives in almost every aspect, apart from our privacy and ability to concentrate. Spam sms is one of the major culprits in this area. Not only can spam messages be downright annoying, but they can also be accessories to cybertheft crimes which can endanger the assets and livelihood of individuals. The need to filter spam accurately is enormous since it affects anyone with a cellular devices. It is a low-hanging fruit problem that can be solved with recent advances in NLP. Our goal for this project is to build a classification model to accurately classify good sms from the spams.

Client: In addition to the ordinary consumers, telecom companies would want to improve the quality of their products to customers by identifying spam sms in a timely fashion and preventing them from ever reaching customers. This way, they will be able to improve their service quality significantly and win over customers.

Success criteria: Recall that our goal is to detect spam sms in real world setting. Therefore, the following criteria are proposed in order of importance:

- Our first criteria is Precision/Specificity. We absolutely don't want to misclassify customer's real email as spam (commit False Positive) because they might lose important information. This means customers may still receive more spam messages than our best model can do. Nevertheless, it would reduce the chance of customers losing important emails to spam boxes. Ideally, we want our algorithm to be more conservative in its prediction and maximize the specificity of our product.
- Our second criteria is general Accuracy. We want to be able to achieve our main objective of detecting spam accurately in text messages in general.
- Our last criteria is Speed. Since customers expect to receive their sms immediately (especially when chatting over sms), we will want to make sure our algorithm does not take long to make a prediction.

Data: We will train our classifier with data from a compilation of many different sms datasets with ham/spam labels provided by UCI dataset directory[1]. The data only has approximately 5,500 observations, which can put a major limit into the model's generalization ability. Therefore, this dataset will serve only experimental purpose. An accurate model will need to be trained on much more examples.

Methodology: The methodology to train the classifier is outlined below:

- Preprocessing: The raw text will be processed through all the NLP data cleaning steps. Then it will be processed into tokens and vectorized so that the data frame because a

[1] Gómez Hidalgo, J.M., Almeida, SMS spam collection v.1, 2011: Retrieved from http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

sparse document-term-matrix. A separate preprocessing will be used to extract features for a baseline traditional model.

- Exploratory Data Analysis (EDA): Common visualization methods to explore the data will be utilized (with packages such as Pandas, Seaborn, nltk, Spacy). Analysis include word count, word length, character count, lexical diversity and TF-IDF analysis.
- Feature Engineering: We will use statistical analysis to recognize and incorporate important features into our model.
- Model Training: We will train two models. One with the features we engineered from EDA and one using Bag of Words and Tf-IDF techniques.
- Model Selection:
    - For the traditional models on the engineered features, we will try Random Forest, Logistic Regression and Gradient Boosted Trees. These algorithms are readily interpretable and could provide us with clues on how to improve our previous model. The best result of these models will be our baseline model to compare against the BoW models. We will quickly discuss the results and flaws of the traditional method.
    - For our Bag of Word models, we will try to classify the sms dataset with Multinomial Naive Bayes variations and SVC on both the Bi-Gram Bag of Word and TF-IDF model. We will discuss the results and flaws of the Bag of Word method.
    - Lastly, we will experiment with building stacks of the best classifiers of the two techniqyes above with the hope that the stacked model will inherit the predictive power from both previous models and negate the flaws of each of previous approaches.
- Model Selection: We will select the best model based on our ordered priority list of criteria (Precision, Accuracy). To improve precision, we will use the Fbeta metrics with Beta = 0.1 (a ten-fold bias towards precision). To improve accuracy, we will consider the secondary criteria of AUC score and accuracy score. We will consider speed in the end to assess how fast our model can be deployed into production environment.

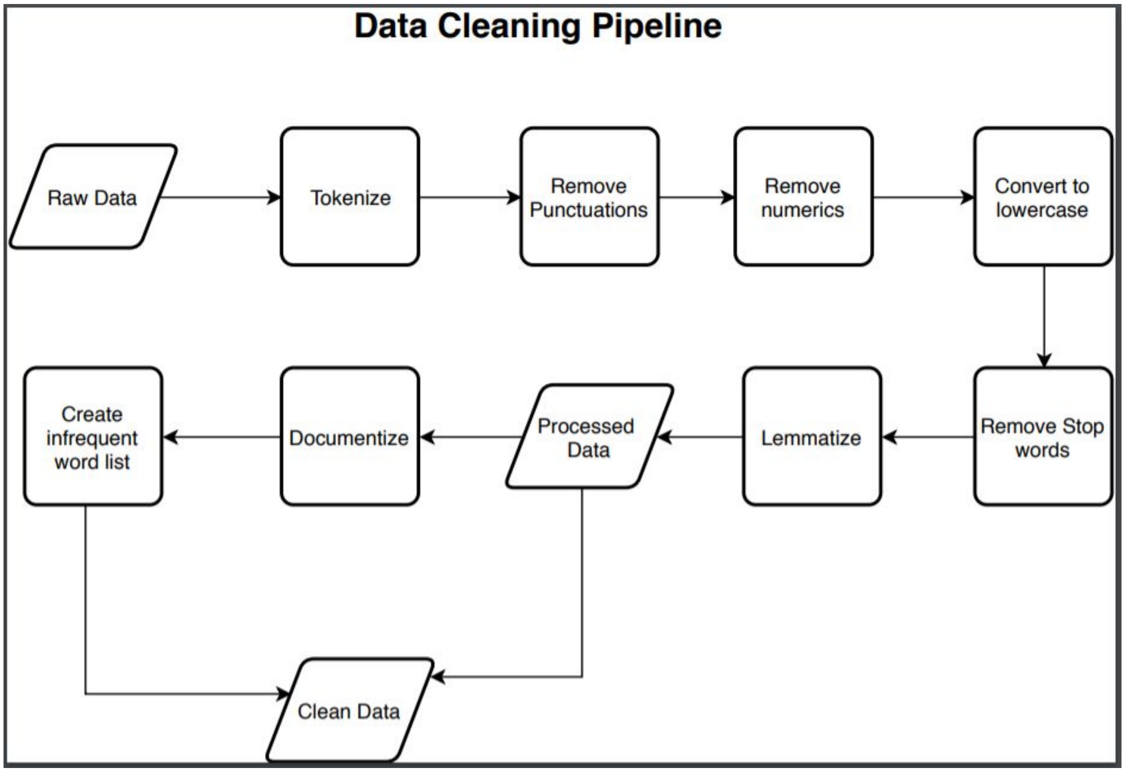
Deliverables: Slide Deck, Report, Code

Part II: Data Cleaning/ Wrangling

The following steps were performed for data cleaning:

- Use Regular Expression to replace discriminative symbols with meaningful extract.
- Importing nltk package for simple text transformation.
- Loading the raw data frame of text.
- Preprocess the text by giving meaningful labels to special symbols such as emails, urls and phone numbers.
- Applying the Tokenize function to every row.
- Removing punctuations.
- Removing stop words.
- Removing numbers.
- Lemmatizing the text to group words which have the same groups.
- Removing words that appear in the whole dataset less than a threshold t (currently set to t = 20). This is a solution to deal with the noise which might be caused by misspellings and typos which frequently appear in text data.

Note: These are initial steps for data cleaning used to conduct EDA and answer the question: What type of contents are in ham rather than spam sms? However, we could also undo some of these transformations to take into account punctuations, numerics and a few interesting stop words as well.

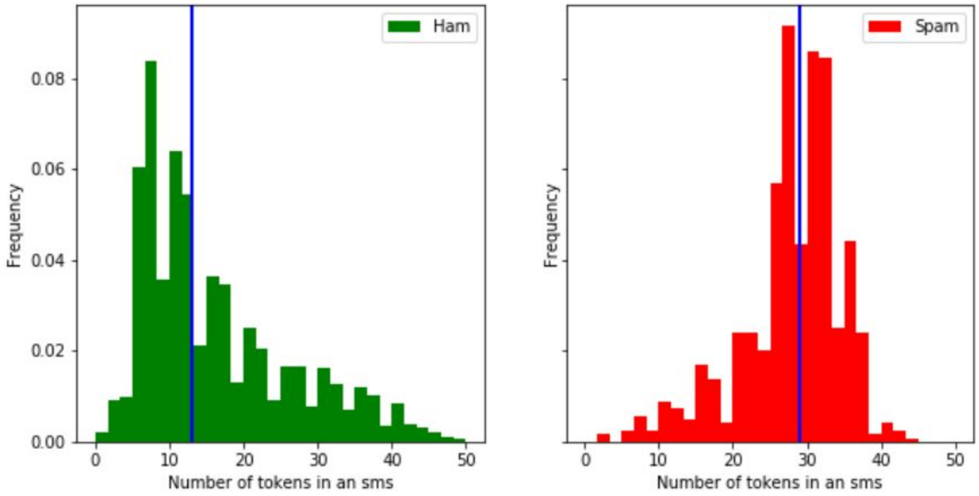


Part III. Exploratory Data Analysis

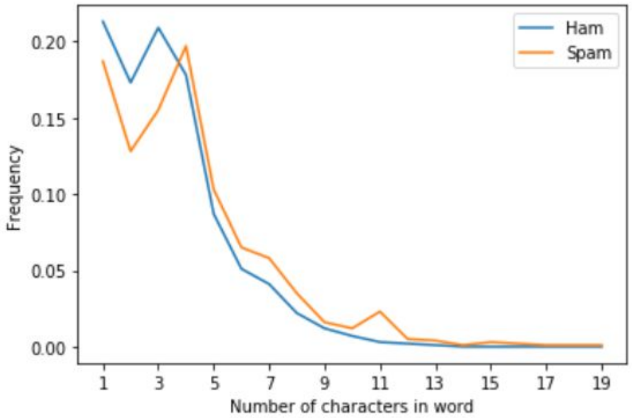
A. Classical feature extraction:

Exploratory Data Analysis was conducted to further get to know the data as well as to extract discriminative features. The following features were analyzed:

- **Length of the SMS**: The number of tokens in each sms was calculated to create the feature n_token for each document. Overall, spam seem to have a higher token count for each document. The following histograms help visualize the difference between ham and spam in terms of this variable. The blue line represents the median token count.

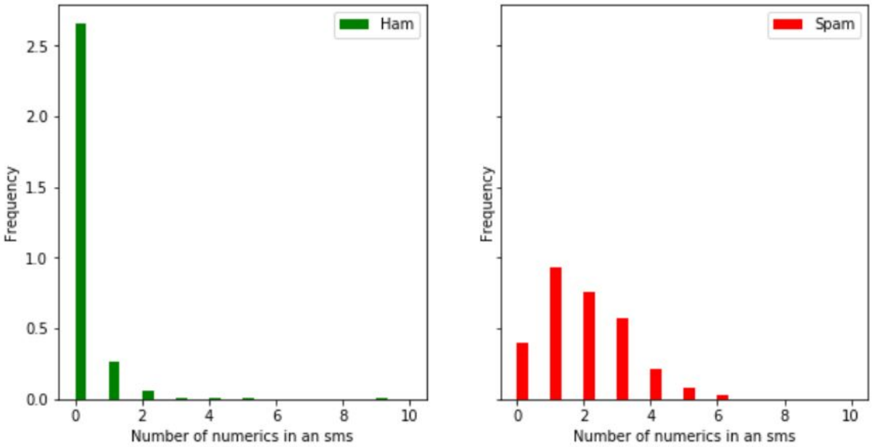


- **Length of sms words**: Length of sms words seem to be discriminative, although not by much as we can see in the following visualization. It seems both ham and spam undergo an exponential decay in character length (which is very typical since shorter words are much more frequent). However, permutation test shows that there is a statistically significant test with p-value of 0 for 10,000 permutation replicates between these two sets. Thus, we can extract n_token as one of our features.

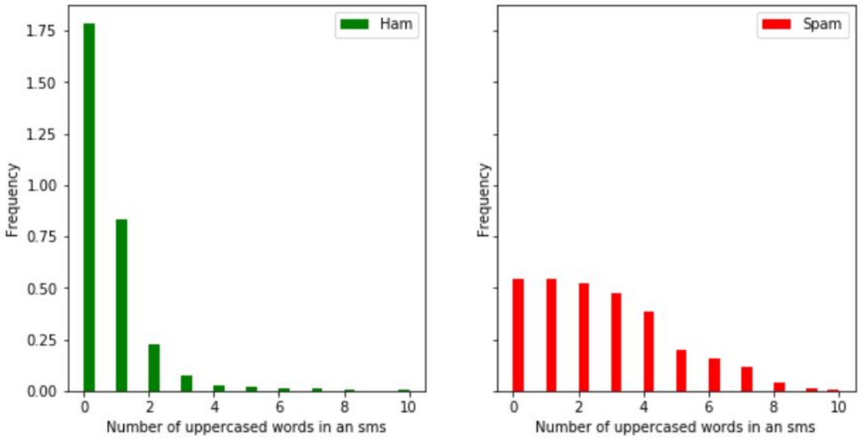


In addition to Length of sms words, maximum and minimum word length was also experimented and visualized. However, these features were shown to be non-discriminative.

- **Number of numerics**: The number of numeric characters within an sms seems to play an important role. It seems that most spam sms has numbers whereas most ham sms don't as visualized below by the number of numerics in a ham or spam document. Since we want to minimize dimensionality, we will extract has_num as a binary feature which describes whether the set has numbers.

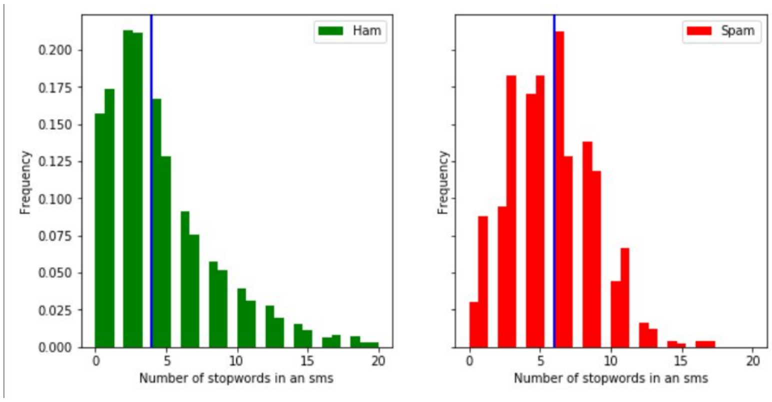


- **Number of uppercase:** In general, ham text has fewer uppercase tokens than spam text. However, the relationship is not entirely discriminatory since there is still a sizeable proportion of spam messages with few uppercase as shown in the histogram below:



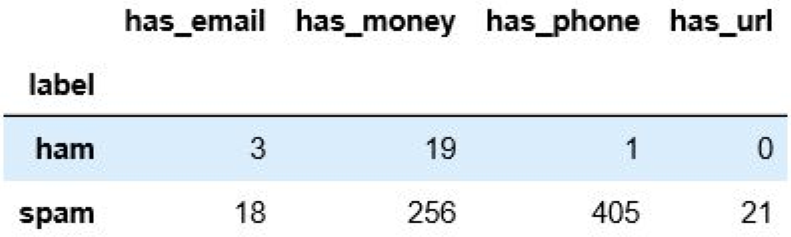
We will, thus keep the feature as n_upper.

- **Number of stopwords:** The distribution of stopwords in spam and ham is different with a different center (median) as shown below:



We will then keep the feature (as n_stop) as it has some predictive power.

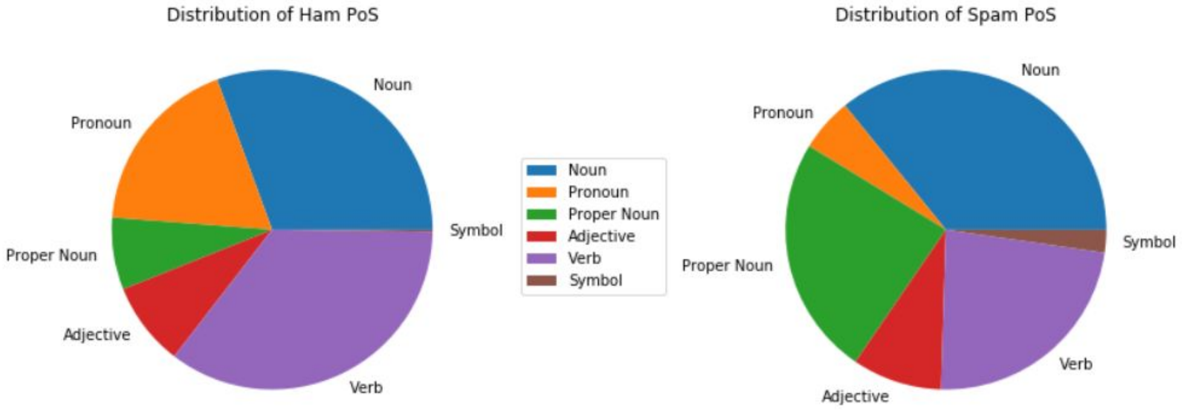
- **Discriminative symbols:** Analyze non-verbal information such as the appearance of email addresses, currencies, phone numbers and urls. The frequency of those within ham and spam is tabulated in the chart below. It is concluded that these factors can be discriminative in classifying ham and spam.



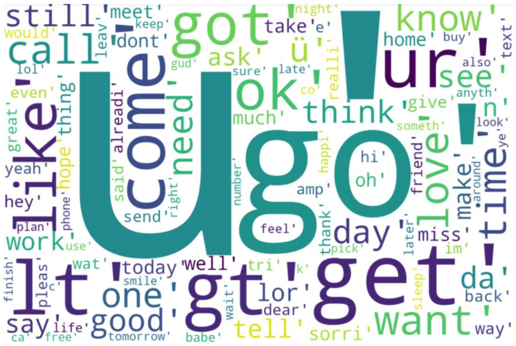
	has_email	has_money	has_phone	has_url
label				
ham	3	19	1	0
spam	18	256	405	21

B. Other visualizations

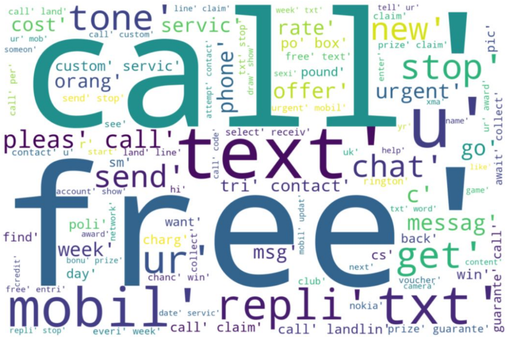
In addition, other NLP and visualization methods such as TF-IDF analysis of the most relatively frequent words, word clouds, Part of Speech with Spacy, and Topic Modelling with gensim was also employed to get a sense of the data. The following Part of Speech Frequency shows that part of speech can be discriminative in ham/spam sms, particularly Symbols, Pronouns, Proper Nouns and Verb.



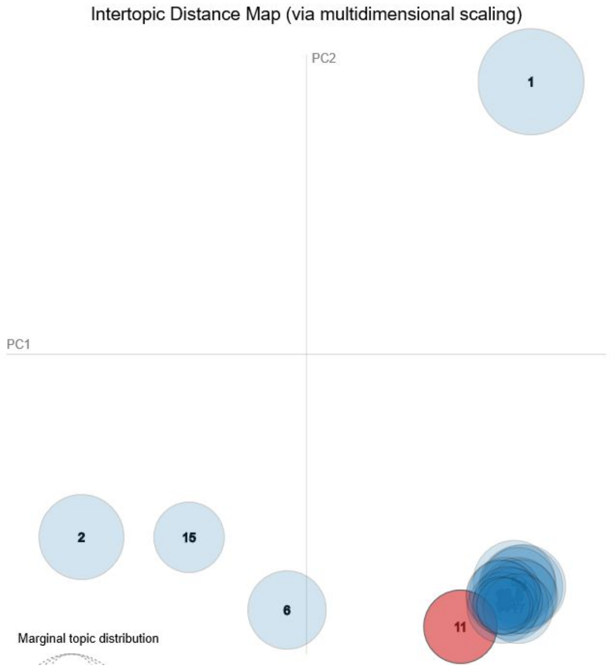
The following word clouds of lemmatized tokens show that ham messages are often conversational and informal, with an emphasis on quick and short words.



Whereas the spam messages can easily be flagged with certain words or phrases.



Finally, a visualization of LDA topic modelling shows that hammy and spammy words can be distinguished across the distance of Principle Component 1 (topic 2, 6 and 15 contain spammy words):

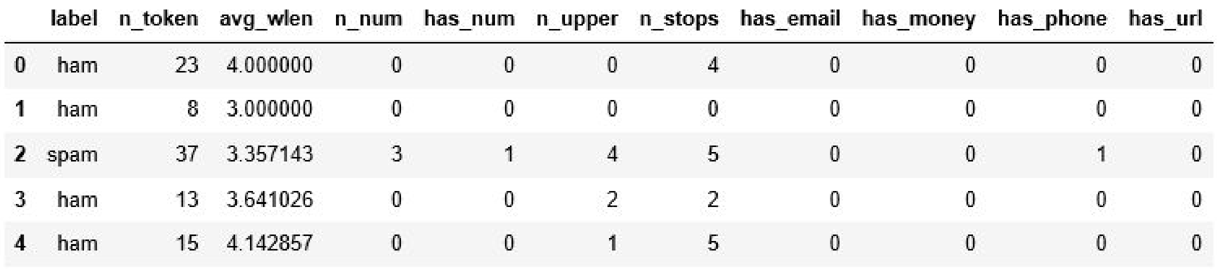


Part IV: Machine Learning models

The objective of the analysis is to build the machine learning component of a spam-filter. Thus, it is necessary to find the best way to predict the label of a given message with high precision and accuracy. After EDA, it is clear that we should explore in-depth machine learning analysis on the classical models as well as the bag of word models. Two dataframes are provided as input of the classical and the bag of words model. The dataframe with extracted features will be fed into Logistic Regression, Support Vector Classification, Random Forest and Gradient Boosted Trees. The raw text data frame will be used to feed into Bi-gram models (Multinomial Naive Bayes, Support Vector Classification) with both Bag of Words and TF-IDF vectorizer. We will apply Scaling, Hyper-Parameter Tuning and GridSearch Cross validation techniques across all models to find the best fitting models of both the classical and bag-of-word technique. Then, we will experiment with ensembling the model to improve our Fbeta, AUC and accuracy score (Metrics of the spam filter's accuracy). Note that the Fbeta parameter is set at 0.1 (which intuitively means we are favoring precision and punishing ham misclassification 10 times as hard). The parameter can easily be tweaked to align with changing business objectives.

A. Classical model

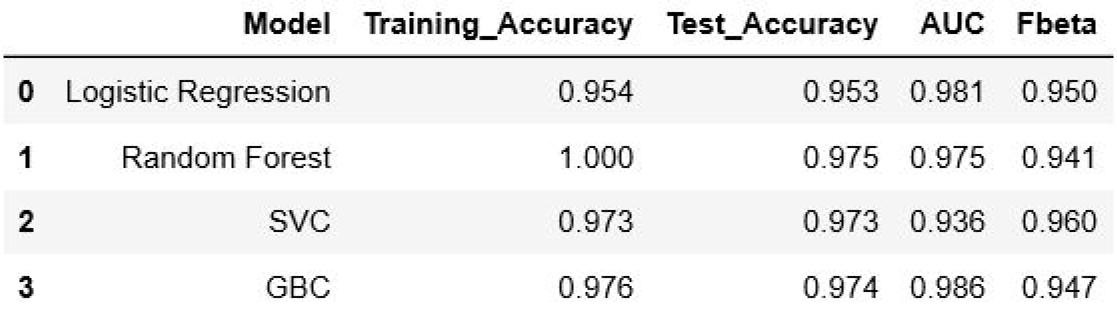
The final data frame that will engender our feature space and label space can be seen as below:



	label	n_token	avg_wlen	n_num	has_num	n_upper	n_stops	has_email	has_money	has_phone	has_url
0	ham	23	4.000000	0	0	0	4	0	0	0	0
1	ham	8	3.000000	0	0	0	0	0	0	0	0
2	spam	37	3.357143	3	1	4	5	0	0	1	0
3	ham	13	3.641026	0	0	2	2	0	0	0	0
4	ham	15	4.142857	0	0	1	5	0	0	0	0

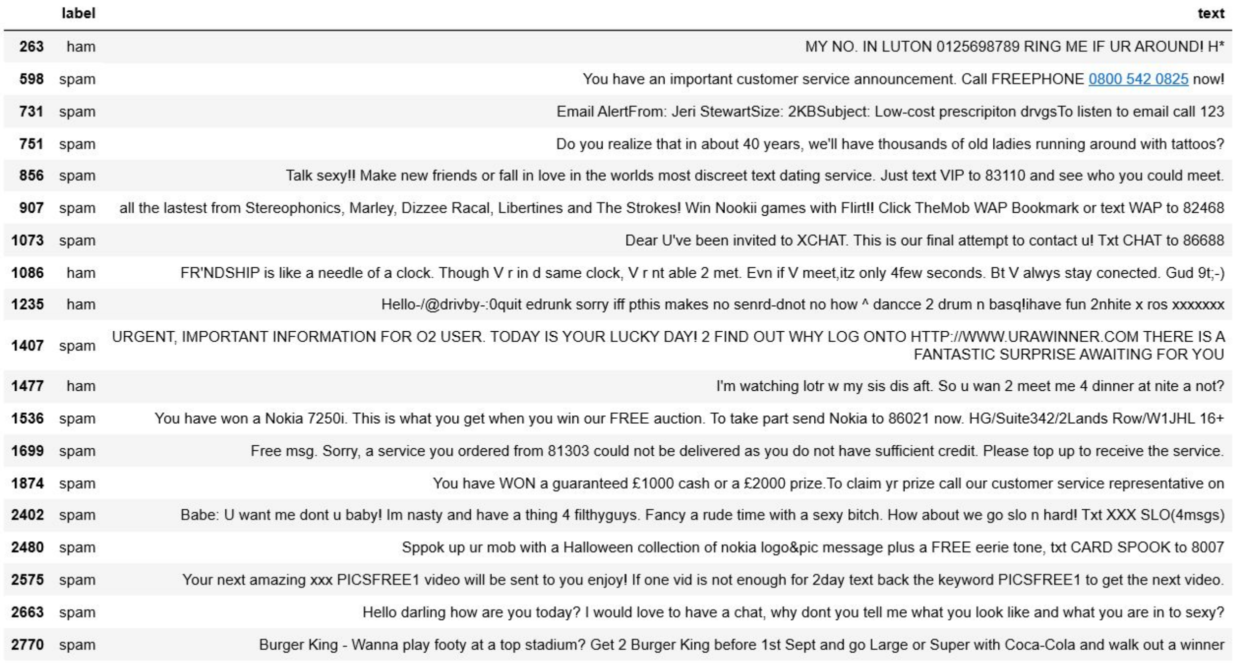
All the features were extracted manually, ready for scaling, divided into train and test set to be fed into our algorithms.

The final result of running the classical models can be seen from the table below:



	Model	Training_Accuracy	Test_Accuracy	AUC	Fbeta
0	Logistic Regression	0.954	0.953	0.981	0.950
1	Random Forest	1.000	0.975	0.975	0.941
2	SVC	0.973	0.973	0.936	0.960
3	GBC	0.976	0.974	0.986	0.947

The best of the classical model seems to be SVC with 0.973 Test accuracy and 0.936 AUC score, but a very high 0.96 Fbeta score. This means that our GBC model can outclass the SVC model in general spam filtering test all else equal. However, since we value our business decision (of not throwing away our customer's messages because the algorithm flagged them as spam), we will choose SVC to inherit our combined model from. 97% accuracy is already pretty decent, but let's see some of its misclassifications:

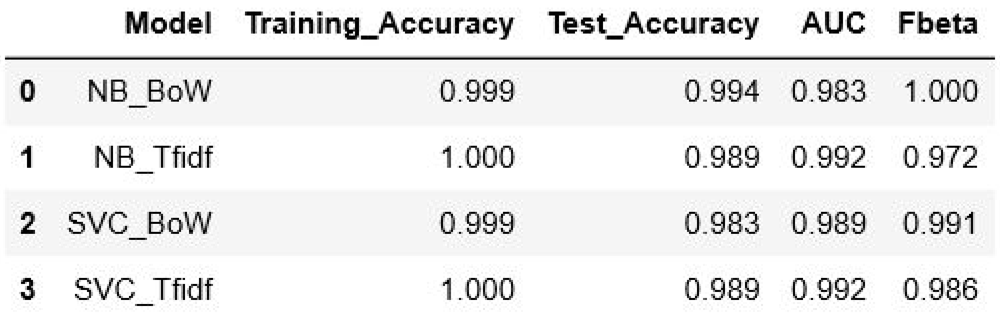


	label	text
263	ham	MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
598	spam	You have an important customer service announcement. Call FREEPHONE 0800 542 0825 now!
731	spam	Email AlertFrom: Jeri StewartSize: 2KBSubject: Low-cost prescripiton drvgsTo listen to email call 123
751	spam	Do you realize that in about 40 years, we'll have thousands of old ladies running around with tattoos?
856	spam	Talk sexy!! Make new friends or fall in love in the worlds most discreet text dating service. Just text VIP to 83110 and see who you could meet.
907	spam	all the lastest from Stereophonics, Marley, Dizzee Racal, Libertines and The Strokes! Win Nookii games with Flirt!! Click TheMob WAP Bookmark or text WAP to 82468
1073	spam	Dear U've been invited to XCHAT. This is our final attempt to contact u! Txt CHAT to 86688
1086	ham	FR'NDSHIP is like a needle of a clock. Though V r in d same clock, V r nt able 2 met. Evn if V meet,itz only 4few seconds. Bt V alwys stay conected. Gud 9t;-)
1235	ham	Hello-/@drivby-:0quit edrunk sorry iff pthis makes no senrd-dnot no how ^ dancce 2 drum n basq!ihave fun 2nhite x ros xxxxxxx
1407	spam	URGENT, IMPORTANT INFORMATION FOR O2 USER. TODAY IS YOUR LUCKY DAY! 2 FIND OUT WHY LOG ONTO HTTP://WWW.URAWINNER.COM THERE IS A FANTASTIC SURPRISE AWAITING FOR YOU
1477	ham	I'm watching lotr w my sis dis aft. So u wan 2 meet me 4 dinner at nite a not?
1536	spam	You have won a Nokia 7250i. This is what you get when you win our FREE auction. To take part send Nokia to 86021 now. HG/Suite342/2Lands Row/W1JHL 16+
1699	spam	Free msg. Sorry, a service you ordered from 81303 could not be delivered as you do not have sufficient credit. Please top up to receive the service.
1874	spam	You have WON a guaranteed £1000 cash or a £2000 prize.To claim yr prize call our customer service representative on
2402	spam	Babe: U want me dont u baby! Im nasty and have a thing 4 filthyguys. Fancy a rude time with a sexy bitch. How about we go slo n hard! Txt XXX SLO(4msgs)
2480	spam	Sppok up ur mob with a Halloween collection of nokia logo&pic message plus a FREE eerie tone, txt CARD SPOOK to 8007
2575	spam	Your next amazing xxx PICSFREE1 video will be sent to you enjoy! If one vid is not enough for 2day text back the keyword PICSFREE1 to get the next video.
2663	spam	Hello darling how are you today? I would love to have a chat, why dont you tell me what you look like and what you are in to sexy?
2770	spam	Burger King - Wanna play footy at a top stadium? Get 2 Burger King before 1st Sept and go Large or Super with Coca-Cola and walk out a winner

Here we can see the drawback of this approach. SMSs with clearly flaggable words like 'Txt', 'Free', the punctuation '!' were ignored as a spam signal (a few urls, phone numbers were missed because of its uniqueness). We can further code these features into our existing model to improve it, but this will clearly show the advantage of the Bag of Word model we will explore next.

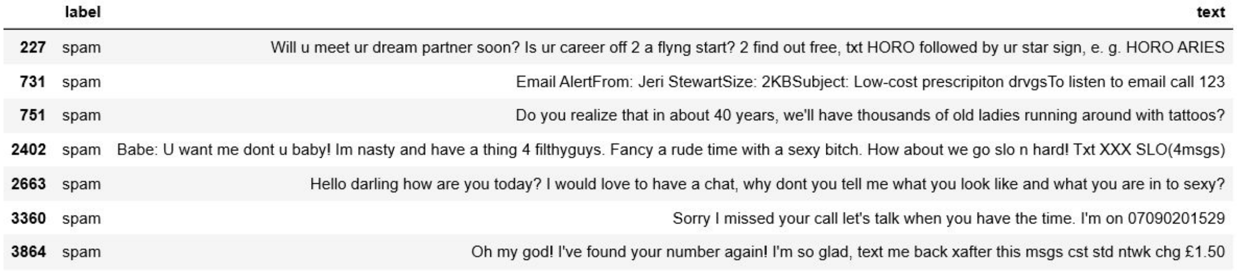
B. Bag of word model

With Bag of word model, we transform the train sets into Bag of Words and TD-IDF representation through scikitlearn CountVectorizer() and TfidfVectorizer(). The result is a Sparse Matrix training set. This matrix is fed into the Naive Bayes and Support Vector algorithms. The final result can be shown below.



	Model	Training_Accuracy	Test_Accuracy	AUC	Fbeta
0	NB_BoW	0.999	0.994	0.983	1.000
1	NB_Tfidf	1.000	0.989	0.992	0.972
2	SVC_BoW	0.999	0.983	0.989	0.991
3	SVC_Tfidf	1.000	0.989	0.992	0.986

Bag of word models are clearly superior to the traditional model. Naive Bayes and SVC are performing on par with each other, up to the 99% test accuracy mark with the Tf-idf transformation of vector space. Below are the misclassifications encountered with the Naive Bayes Bag of Word approach. Note that this model is very specific, as it only gets spam misclassified.

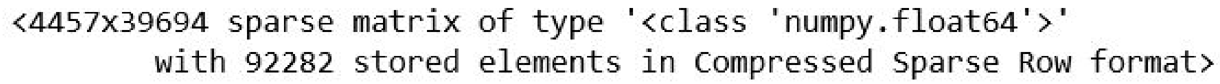


	label	text
227	spam	Will u meet ur dream partner soon? Is ur career off 2 a flyng start? 2 find out free, txt HORO followed by ur star sign, e. g. HORO ARIES
731	spam	Email AlertFrom: Jeri StewartSize: 2KBSubject: Low-cost prescripiton drvgsTo listen to email call 123
751	spam	Do you realize that in about 40 years, we'll have thousands of old ladies running around with tattoos?
2402	spam	Babe: U want me dont u baby! Im nasty and have a thing 4 filthyguys. Fancy a rude time with a sexy bitch. How about we go slo n hard! Txt XXX SLO(4msgs)
2663	spam	Hello darling how are you today? I would love to have a chat, why dont you tell me what you look like and what you are in to sexy?
3360	spam	Sorry I missed your call let's talk when you have the time. I'm on 07090201529
3864	spam	Oh my god! I've found your number again! I'm so glad, text me back xafter this msgs cst std ntwk chg £1.50

A look at where the Bag of Word model fails (with Support Vector Classification algorithm) shows that it I misclassified a few messages due to its over-reliant on the probability of spammy and hammy words. However, these are understandable limitations of the Bag of Word model as it is unable to rely on context or semantics to classify. An interesting note here is that Tfidf provides better AUC score, but lower Fbeta score. It looks as though the ordinary bag of word is more conservative in flagging spams. Once again, that is what we want, so we will use the Naive Bayes Bag of Word model for the combined model to inherit.

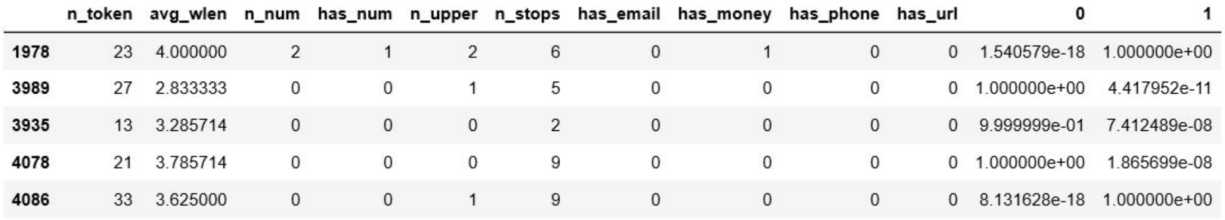
C. Stacked Model

Two techniques to stack the best classical and bag of word model was explored.
The first (stack_combined) was to simply append the features of the classical model with the bag of word to create a mix of sparse and scaled dense features to be trained with Naive Bayes. The final set is a bag of word sparse matrix with classical features incorporated:



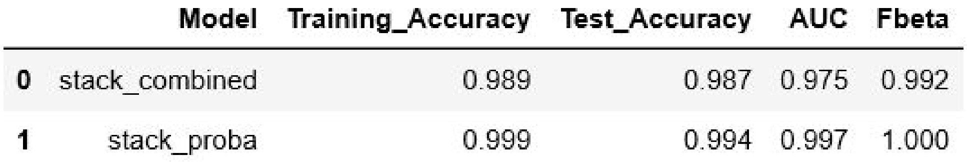
```
<4457x39694 sparse matrix of type '<class 'numpy.float64'>'
        with 92282 stored elements in Compressed Sparse Row format>
```

The second (stacked_proba) was to append the probabilities predicted by the bag of word model (Naive Bayes) as two more features in the classical training set to create the training set of the combined model. The final training set can be seen below:



	n_token	avg_wlen	n_num	has_num	n_upper	n_stops	has_email	has_money	has_phone	has_url	0	1
1978	23	4.000000	2	1	2	6	0	1	0	0	1.540579e-18	1.000000e+00
3989	27	2.833333	0	0	1	5	0	0	0	0	1.000000e+00	4.417952e-11
3935	13	3.285714	0	0	0	2	0	0	0	0	9.999999e-01	7.412489e-08
4078	21	3.785714	0	0	0	9	0	0	0	0	1.000000e+00	1.865699e-08
4086	33	3.625000	0	0	1	9	0	0	0	0	8.131628e-18	1.000000e+00

The results are shown below:



	Model	Training_Accuracy	Test_Accuracy	AUC	Fbeta
0	stack_combined	0.989	0.987	0.975	0.992
1	stack_proba	0.999	0.994	0.997	1.000

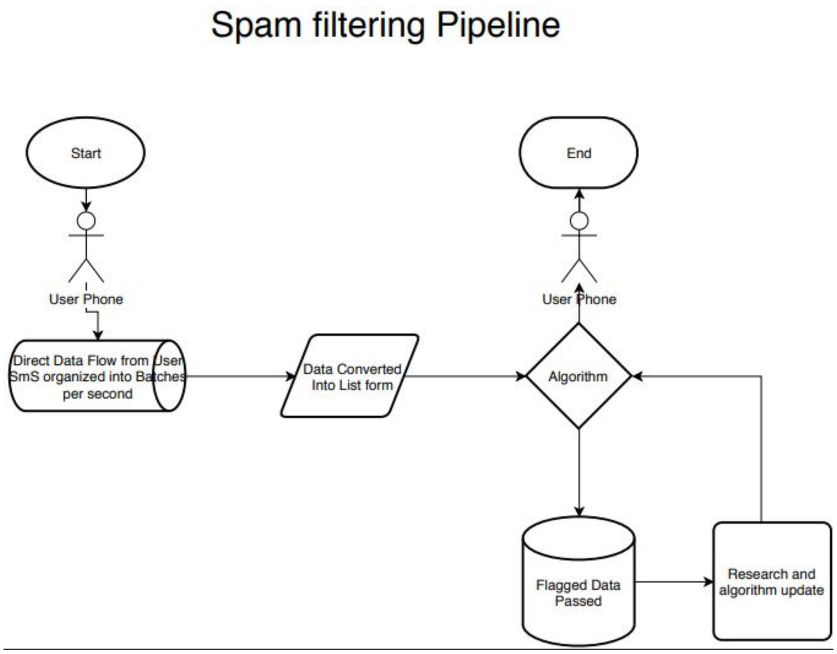
The stacked probability model performed the best, inheriting an almost perfect Fbeta score from Naive Bayes probability features while still having a higher AUC score due to the small nudges from the classical model features.

Part V: Conclusion, Limitations and Room for Improvements

Conclusion: With respect to the business outcome, the final model was able to achieve the following:

- An Fbeta score of nearly 1 with beta threshold of 0.1 means this model has high specificity. In business sense, this means you would expect the model to never misclassify user's messages as spam. This prevents the undesireable situation where the user cannot get an important message because it was flagged as potential spam.
- An AUC of 0.997: This means that the model is generally really good at distinguishing authentic messages from spam messages and can correctly make the distintion 99.7% of the time. Thus, you would expect the model to misclassify 3 cases out of 1000 cases.
- A speed of 640 ms (0.64 seconds) per 100 batches on a local machine: A minimum viable spam filter was build base on this algorithm ready to be launched for testing in production. The speed was tested on a local machine with a little over half a second for a batch of 100 messages. This means that using the computing resource of the local machine alone, we are able to safely run message filtering in batches of around ~ 150 messages per second. If we update by the second, there won't be a noticable effect for users.

The pipeline for such product can be outlined below:


Spam filtering Pipeline
Start
User Phone
Direct Data Flow from User SmS organized into Batches per second
Data Converted Into List form
Algorithm
End
User Phone
Flagged Data Passed
Research and algorithm update

Limitations: A few limitations of the model must be addressed as followed:

- Lack of training data**:** The corpus we trained the model on was frankly small compared to the millions of messages being sent out there. The model may have only internalized a small subset of all sms. If we collected data in the real world, the complexity might hinder model performance.
- Architecture: The model architecture is still very basic and cannot incorporate many new advances in semantics and syntax. Although the model is accurate within the confine of this dataset, the model's lack of semantic incorporation may reduce its accuracy in real world data.

Room for improvements: Despite our positive results, it's possible to improve our model even further.

- Improve pre-processing: Since the pre-processing still left out a few gaps (e.g: phone numbers with 6 digits not found by RE, links without http), we could definitely look further into improving it.
- Improve generality: We can increase the ability of the model to run as a spam filter effectively in the real world by feeding it larger datasets and more messy data (more up-to-date, different languages, emoticons ...etc).
- Improved technique: Many new techniques in NLP has been recently popularized. We could experiment with training the model using more recent NLP arsenals such as Word Embedding techniques and Transformers.