Spam Messages Classification

ABSTRACT :

Use of supervised machine learning for natural language processing to classify spam messages

Spam messages are messages sent to a large group of recipients without their prior consent, typically advertising for goods and services or business opportunities. In the recent period, the percentage of scam messages amongst spam have increased sharply. Scam messages typically trick people into giving away money or personal details by offering an attractive or false deal. Based on the statistics from the Singapore Police Force, from January till June 2020, the amount cheated through scams have increased by more than S$8 million!

INTRODUCTION :

A spam message classification is a step towards building a tool for scam message identification and early scam detection.

Dataset

The dataset is from Kaggle, a collection of spam SMS messages, with 5572 messages, all classified as either ‘ham’ or ‘spam’ . The dataset contains 13.4% spam and 86.6% ham.

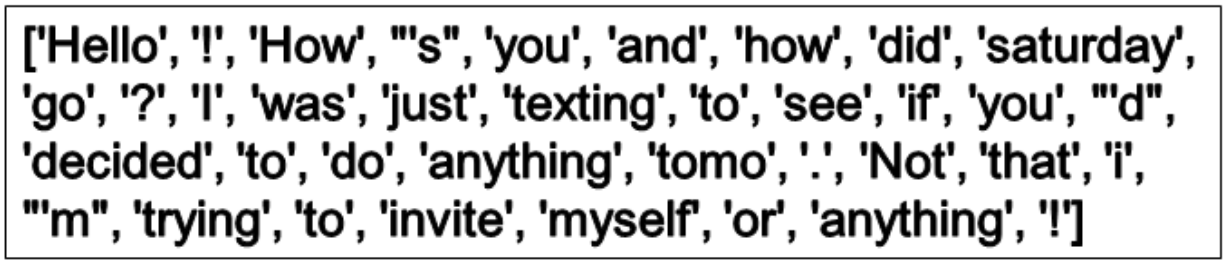
Methodology

The methodology is divided into

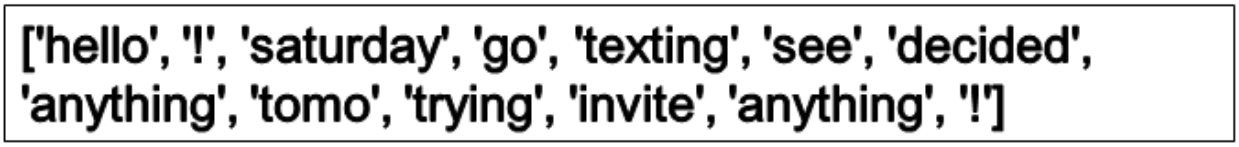
1. Data Pre-processing and Exploratory Data Analysis
2. Model Training, Comparison and Selection

3. Model Evaluation

Data Pre-processing



Text shown after Word-Tokenize step, image by author.



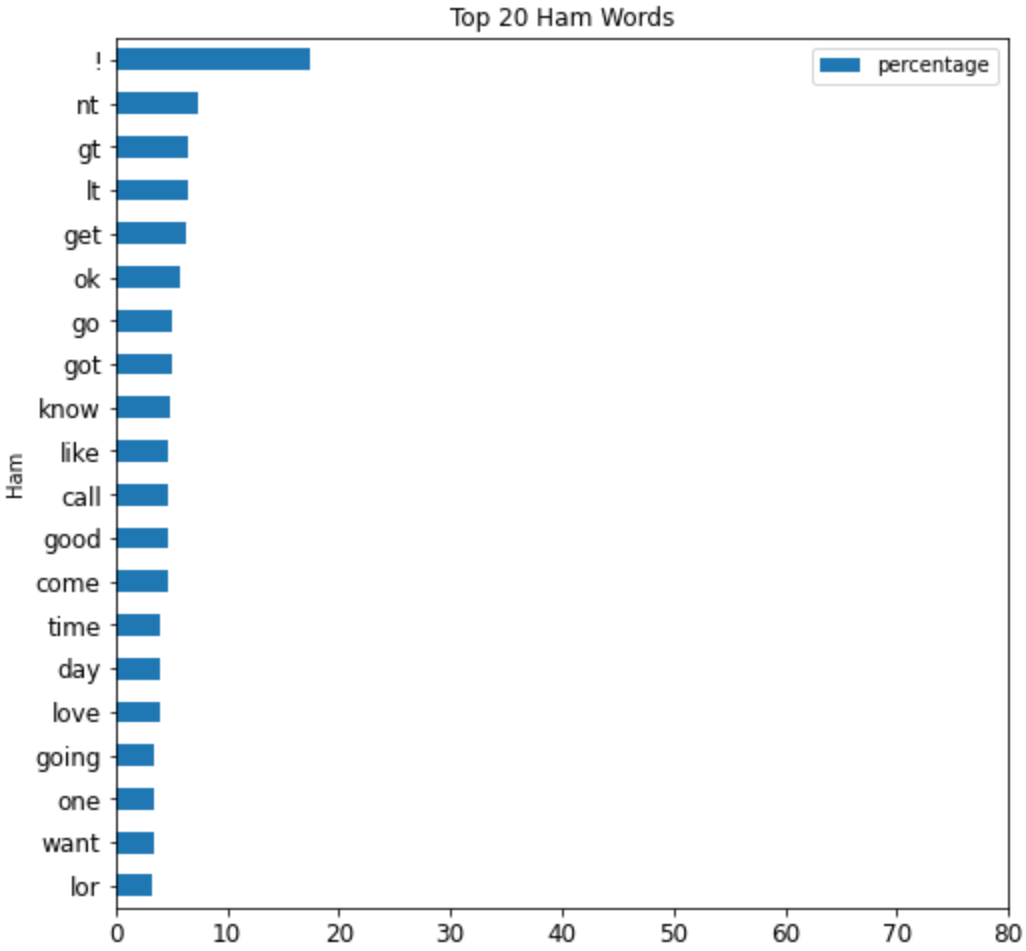
Same text shown after subsequent pre-processing steps, image by author.

With the help of the NLTK package, every message is word-tokenized. Conversion to lower case is carried out, punctuation is removed except exclamation mark, stopwords and words containing digits are removed too.

The example above shows a ‘cleaned’ message after all pre-processing steps.

Exploratory Data Analysis

The top 20 words used in spam and ham messages are illustrated using horizontal bar plots.


Top 20 Ham Words
percentage
Ham
!
nt
gt
lt
get
ok
go
got
know
like
call
good
come
time
day
love
going
one
want
lor
0
10
20
30
40
50
60
70
80

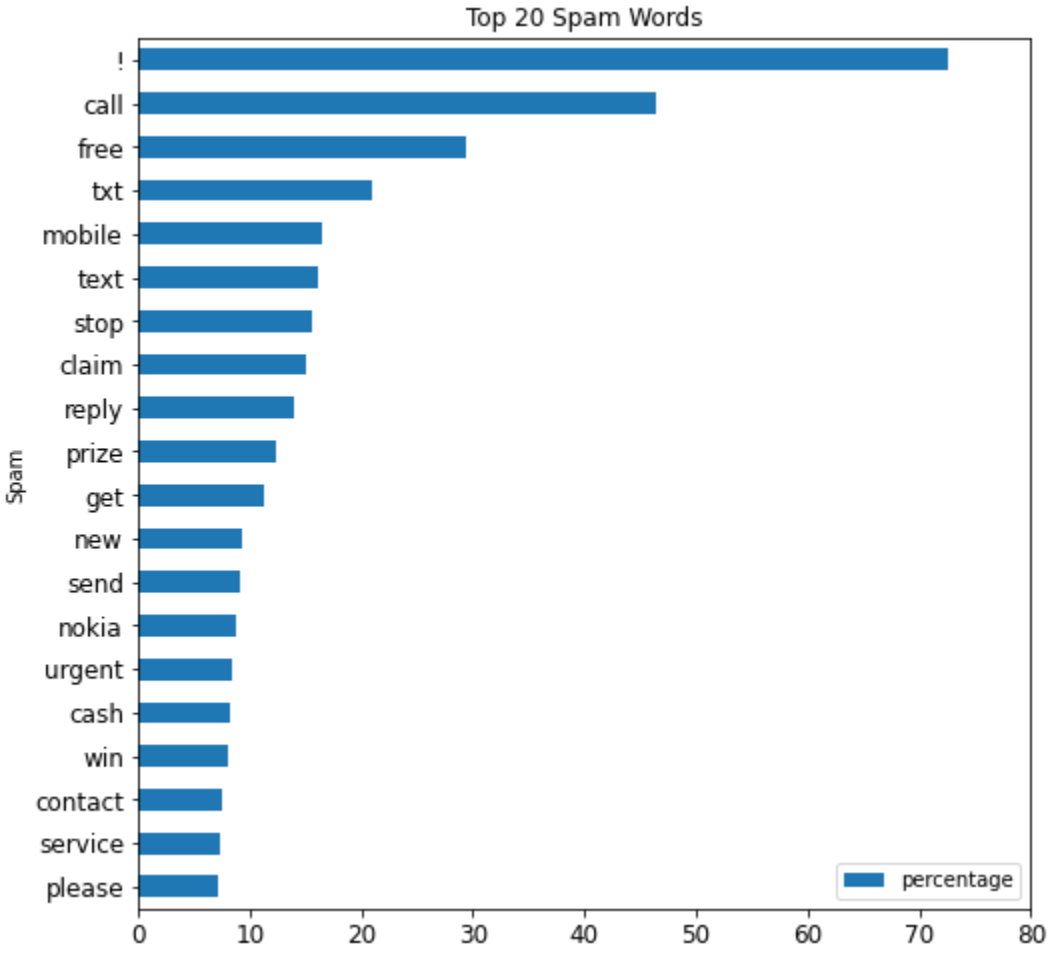
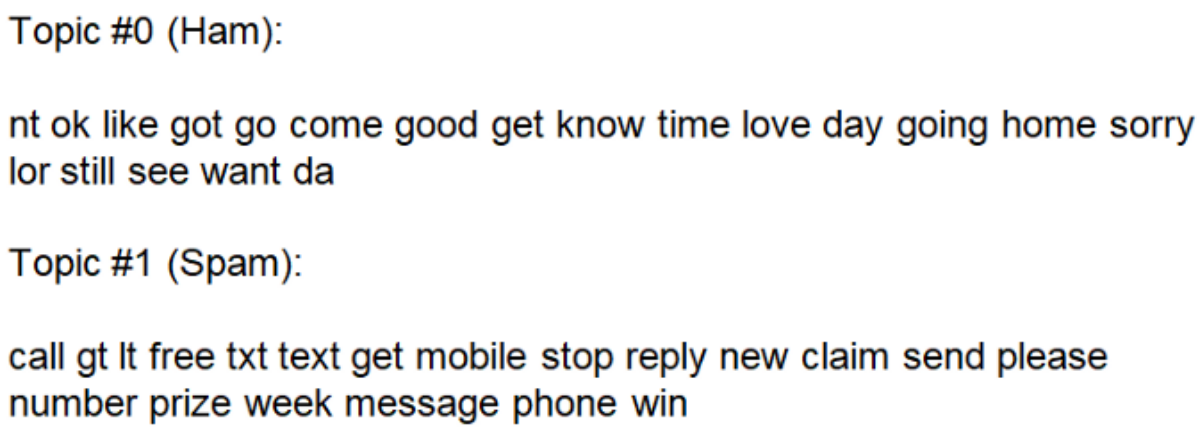


Image by author.

It can be seen that words common in spam messages are not commonly observed in ham messages. The percentage of occurrence of spam words are very high too in spam messages.

Topic modeling has also been used on the ‘cleaned’ text to discover the topics that occur in the collection of text (usually known as documents in NLP terms).

Here, Latent Dirichlet Allocation (LDA) is used to classify the documents into 2 topics. As seen below, the words in topic 0 seem to suggest that the topic is Ham and words in topic 1 seem to suggest that the topic is Spam.



Results from topic modeling by LDA, image by author.

Topic modeling is a useful technique for reducing dimensions before applying classification models.

Model Training

The models used are Logistic Regression, *k*-Nearest Neighbors classifier, Random Forest classifier, Bernoulli Naïve Bayes classifier and Complement Naïve Bayes classifier.

Count Vectorizer is used to encode the documents into vectors as input into the models for training and test. In addition, Tfidf vectorizer, Count Vectorizer with Latent Semantic Analysis, Count Vectorizer with Latent Dirichlet Allocation has been used with the Logistic Regression model.

The data is split into 80% train and validation set, 20% test set. GridSearch CV is applied across 10 folds on the data, to find the best hyperparameters for all the models. Prediction is based on the average of 10 predictions.

Model Comparison and Model Selection

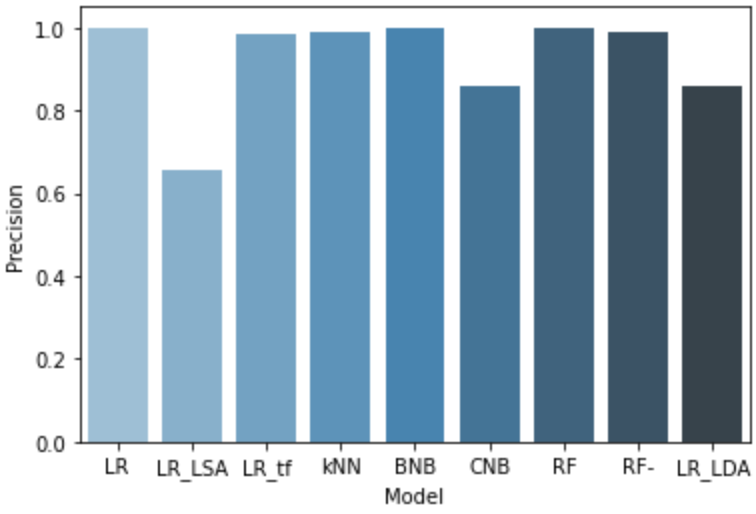


Image by author.

Precision is the fraction of true spam messages out of the captured spam messages.

The models that have 100% precision are Logistic Regression, Bernoulli Naïve Bayes classifier, Random Forest classifier.

Random Forest classifier model has also improved its accuracy from 99% to 100% with the inclusion of the exclamation mark in the text.

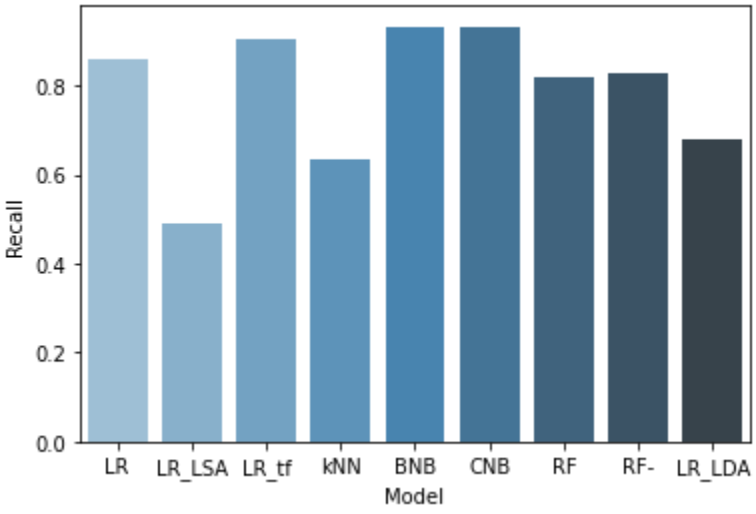


Image by author.

Recall is the fraction of true spam messages captured out of the total true spam messages present.

The Bernoulli Naïve Bayes classifier and Complement Naïve Bayes classifier have the highest recall score of 0.93.

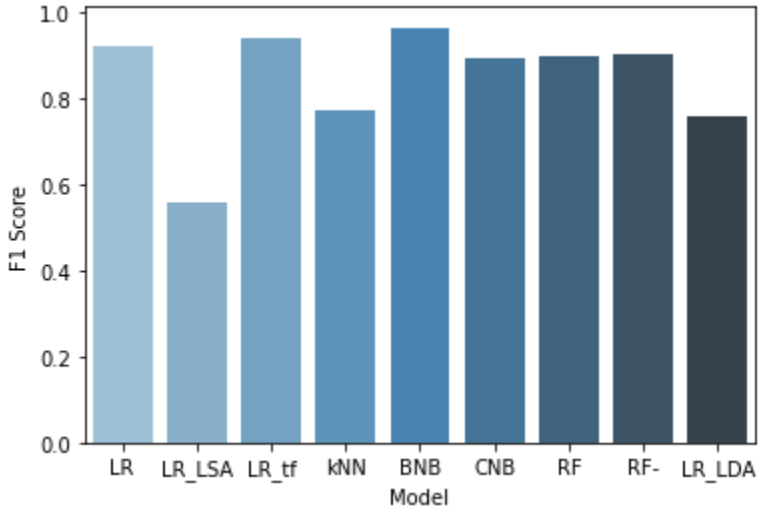


Image by author.

F1 score is the measure of the model's accuracy based on precision and recall. The Bernoulli Naïve Bayes classifier has the highest F1 score of 0.97.

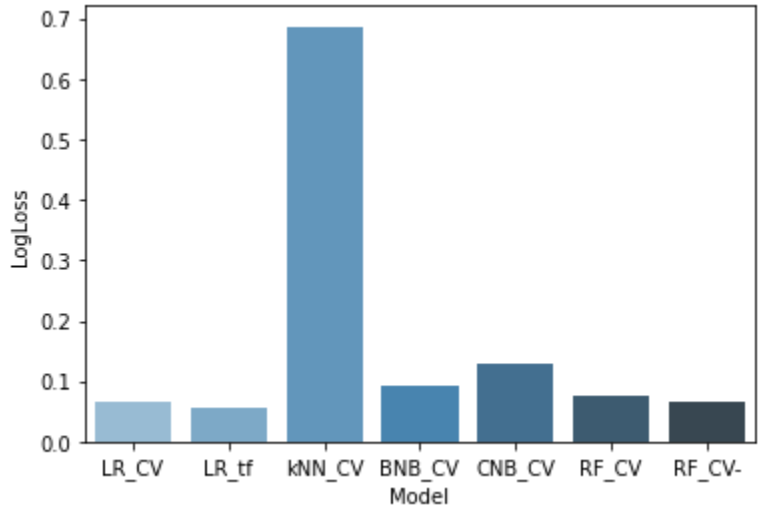


Image by author.

The log loss score heavily penalizes predicted probabilities far away from their expected value. Almost all models have done well to give a log loss of below 0.1, except *k*-Nearest Neighbor classifier and Complement Naïve Bayes classifier.

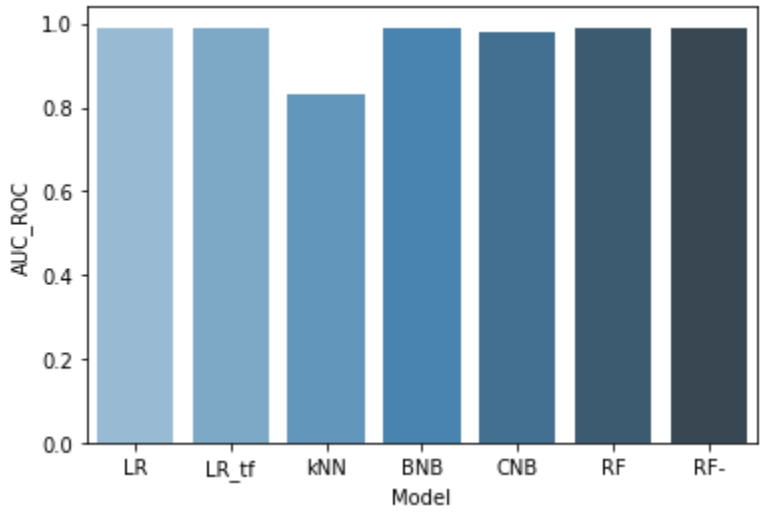


Image by author.

AUC tells how much the model is capable of, distinguishing between spam and ham messages.

Almost all models have done well giving AUC of 0.99 except the k-Nearest Neighbor classifier and Complement Naïve Bayes classifier.

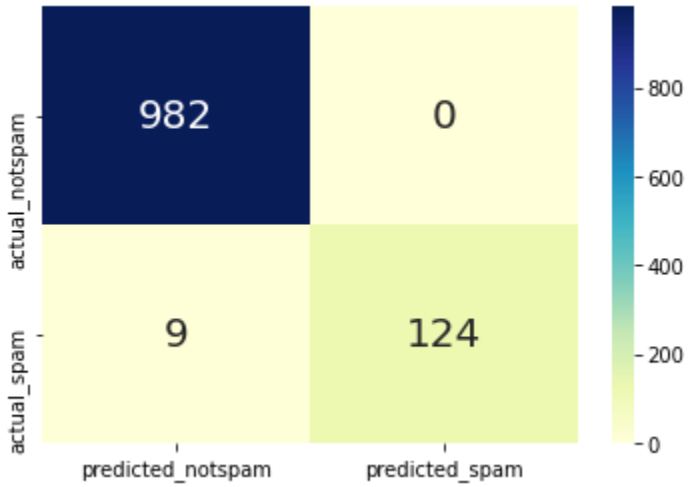
The table below summarizes the results.

Model	LR	LR_LSA	LR_tf	kNN	BNB	CNB	RF	RF-	LR_LDA
Precision	1	0.66	0.98	0.99	1	0.86	1	0.99	0.86
Recall	0.86	0.49	0.90	0.63	0.93	0.93	0.82	0.83	0.68
F1 Score	0.92	0.56	0.94	0.77	0.97	0.90	0.90	0.90	0.76
LogLoss	0.07	N.A.	0.06	0.69	0.09	0.13	0.07	0.07	N.A.
AUC ROC	0.99	N.A.	0.99	0.83	0.99	0.98	0.99	0.99	N.A.

Summary of results, image by author.

Model Evaluation

The selected model is the Bernoulli Naïve Bayes classifier. From the confusion matrix below, the model is able to give a precision score of 1 and a recall score of 0.93.



Confusion matrix from the Bernoulli Naïve Bayes classifier, image by author.

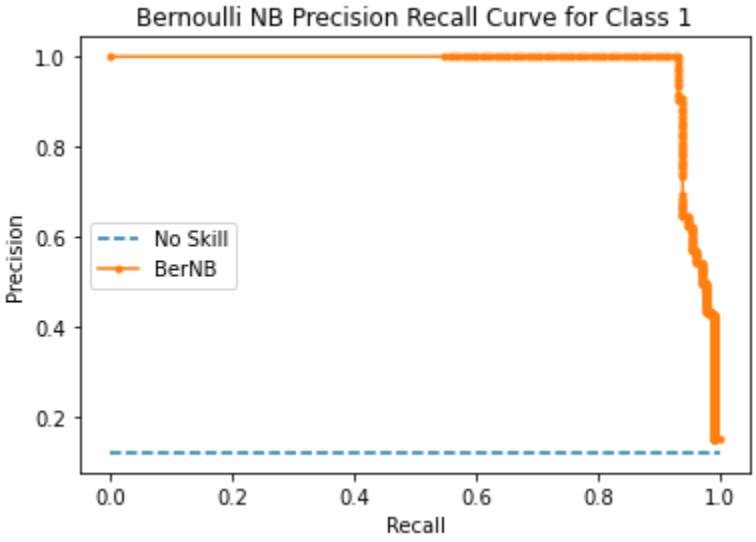


Image by author.

Conclusion

A model with a 100% precision has been built. Precision is valued over recall so that ham messages will not be misclassified as spam.

A more customized pre-processing step will contribute to a more precise model.

For this dataset, linear models with high bias and low variance like the Logistic Regression and Naïve Bayes classifier have performed better than non-linear models like *k*-Nearest Neighbor classifier and Random Forest classifier.

Acknowledgement

Special thanks to my main Metis instructor Mr. Neo Han Wei for the guidance and patience given in this entire boot camp. And thanks to my co-instructors Mr. Desmond Sek and Mr. A.M. Aditya for their teaching too.

Here is a link to my GitHub where you can find my codes and presentation slides. You can also contact me via LinkedIn.