
Spam Prediction Using Natural Language Processing

Shreyans S. Shah-200460185
Software Systems Engineering
University of Regina
3737 Wascana Parkway
Regina, SK S4S 0A2

Abstract

Spam Prediction Using Natural Language Processing is a Deep Learning project that is capable of detecting spam messages by analysing the patterns in the body of the email or messages. Users may submit their own messages into this system, and the algorithm will predict if the message is spam or not. It has been my experience to utilise numerous classifiers and to assess the correctness of the model. I have also employed various data processing methods with the aid of the NLTK toolkit.

Keywords:

Spam Detection, Deep Learning, Natural Language Processing, Text Classification, Email Processing, SMS processing, Sci-Kit Learn , sklearn , Natural Language Toolkit , nltk , Python

Source Code:

https://github.com/shahshreyans/SpamPrediction_NLP_Deep_Learning

1. Introduction

In today's business world, email and messaging is a primary source of communication. This communication can vary from personal, business, corporate to government. With the tremendous growth of email use has come a surge in SPAM emails. Spam emails, in the opinion of the majority of people, are those that are bothersome and are repeatedly utilised for the aim of advertising and brand promotion. Additionally, Because of the increasing popularity of mobile phone devices over the last several years, the Short Message Service (SMS) sector has evolved into a multi-billion-dollar enterprise. As a consequence of decreased costs for message services, an increase in the number of unwanted commercial ads and spam messages are being delivered to mobile phones has occurred. Apart from being annoying, spam emails and messages can also pose a security threat to computer system. Some of the most common types of spam emails and messages that pose a significant threat to security include fraudulent e-mails, identity theft, hacking, viruses, and malware, among other things. Spam email or messages are sent to a large number of individuals who do not recognise them and end up as a victim of online fraud and theft. As a result, the development of an autonomous system that would identify spam messages by identifying their pattern is the most important thing to do. In this project I'm using Kaggle dataset of spam and non-spam messages which will use to train the model, which will subsequently able to predict whether user messages are spam or non-spam.

2. State of The Art:

With the increasing use of Machine Learning algorithms in real-world issues, categorization of algorithms has become an unavoidable consequence of this trend. As a result, the vast majority of machine learning algorithms are grouped into the category of described issues, which is referred to as the state of the art. Computer Vision, Natural Language Processing, Speech, Medical Graphs, and other areas of cutting-edge technology are examples of state-of-the-art technology. Natural Language Processing is the state of art of this project as we classified text/message which is natural language. There are plethora projects which are lies in the text classification state of art. Some projects which I believe is worth eye catching like Fake tweets detection, User Complain classification, Fake comments removal .

3. Purpose:

The primary goal of this project is to use a deep learning classifier to determine if text messages fall into the categories of spam or not-spam and to categorise them accordingly.

4. System Architecture & Evolution:

This System can be divided into three major components:

- 1)Data Processing
- 2)Training Model
- 3) Prediction

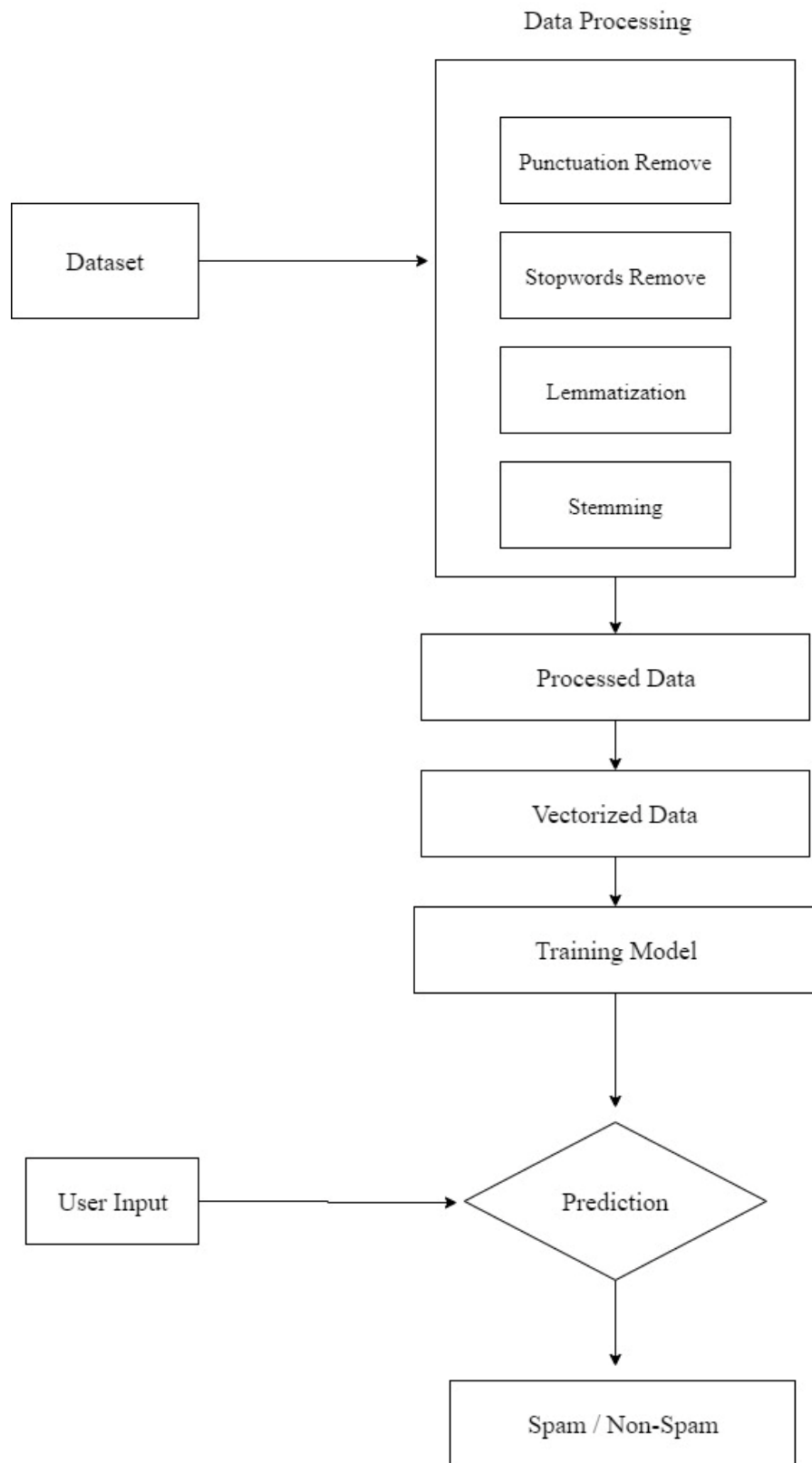


Figure 1: Flow Chart

Using a Kaggle dataset of spam and non-spam communications, as seen in the picture, I first analysed the data to ensure that it was suitable for efficient detection prior to training. Following the processing of the data, it is necessary to train the model with the assistance of classifiers. We can now predict the user input messages and evaluate the accuracy of the model.

4.1 Dataset:

The dataset utilised in this project was obtained from Kaggle, which is a well-known platform for machine learning datasets and resources. Sms train.csv and Sms test.csv are the two CSV files included in the dataset. Each file is divided into three columns: Sr.No, Message body, and Label. The message body contains the message's content, while the label contains a Spam or Non-Spam value based on the message's classification

Sms Train dataset has 973 unique values, of which 847 are non-spam and 126 are spam. Sms Test Dataset contains 125 distinct values, of which 76 are spam and 49 are not.

4.2 Data Processing:

Now, the dataset must be handled before to training, since any extraneous or unnormalized data might impair the program's accuracy. Additionally , our data is in text format, we must transform the text data into a numerical array before we can use it as training data because the machine cannot interpret text data. In this Project I processed data using four different method.

4.2.1 Punctuation Remove:

There are several punctuation marks in our text messages that are completely ineffective for mode prediction and training purposes. Therefor, I have removed all the punctuation using simple String Replace and Punctuation methods.

4.2.2 Stop Words Remove:

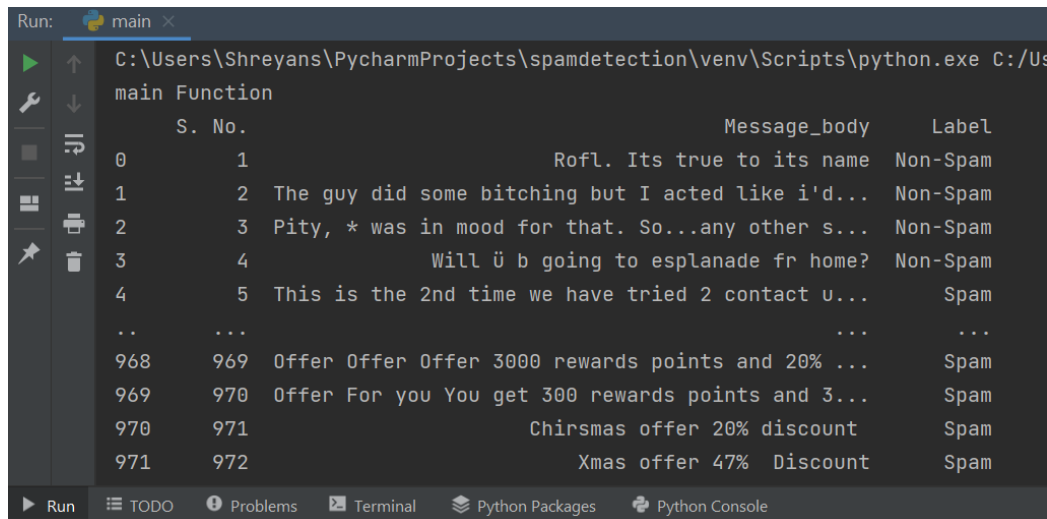
Stop Words are the terms that are often used in text input or searches and are thus classified as such. Stop words are essentially a collection of terms that are often used in any language. Stop Words examples are a,the ,an , and , but ,what For effective data processing or data mining, it is necessary to exclude certain often used phrases like Stop Words. So, in this project, I've eliminated the Stop Words by using the NLTK library, which stands for Natural Language Tool Kit and is a critical library for natural language processing applications.

4.2.3 Lemmatization:

Lemmatization is a process that make a word into more meaningful way like eating word will become eat after Lemmatization. So, we may define lemmatization as the process of turning words back to their original form. This will help to increase the accuracy of search as dataset's words will be in original form. Again, NLTK provide us a WordNetLemmatizer which will do lemmatize process.

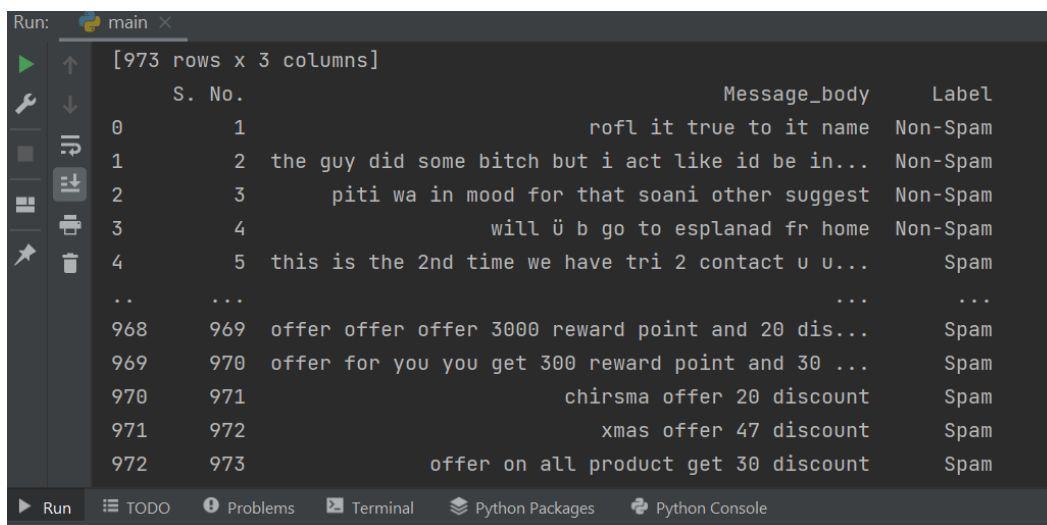
4.2.4 Stemming:

Stemming is a process that will remove affix from the words like eats become eat, believes become believ after use of stemming. NLTK provide different stemmer like Porter Stemmer , Lancaster Stemmer, Snowball Stemmer and they follow different algorithms. I tested every stemmer I could find, but Snowball Stemmer provided me with the highest level of accuracy, which is why I chose it for this particular project.



| | S. No. | Message_body | Label |
|-----|--------|---------------------------------------------------|----------|
| 0 | 1 | Rofl. Its true to its name | Non-Spam |
| 1 | 2 | The guy did some bitching but I acted like i'd... | Non-Spam |
| 2 | 3 | Pity, * was in mood for that. So...any other s... | Non-Spam |
| 3 | 4 | Will ü b going to esplanade fr home? | Non-Spam |
| 4 | 5 | This is the 2nd time we have tried 2 contact u... | Spam |
| .. | ... | ... | ... |
| 968 | 969 | Offer Offer Offer 3000 rewards points and 20% ... | Spam |
| 969 | 970 | Offer For you You get 300 rewards points and 3... | Spam |
| 970 | 971 | Chirsmas offer 20% discount | Spam |
| 971 | 972 | Xmas offer 47% Discount | Spam |

Figure 2:DataSet Before Processing



| | S. No. | Message_body | Label |
|-----|--------|---------------------------------------------------|----------|
| 0 | 1 | rofl it true to it name | Non-Spam |
| 1 | 2 | the guy did some bitch but i act like id be in... | Non-Spam |
| 2 | 3 | piti wa in mood for that soani other suggest | Non-Spam |
| 3 | 4 | will ü b go to esplanad fr home | Non-Spam |
| 4 | 5 | this is the 2nd time we have tri 2 contact u u... | Spam |
| .. | ... | ... | ... |
| 968 | 969 | offer offer offer 3000 reward point and 20 dis... | Spam |
| 969 | 970 | offer for you you get 300 reward point and 30 ... | Spam |
| 970 | 971 | chirsma offer 20 discount | Spam |
| 971 | 972 | xmas offer 47 discount | Spam |
| 972 | 973 | offer on all product get 30 discount | Spam |

Figure 3: Dataset After Processing

4.3 Vectorizing:

After Our data is processed in meaningful manner we have to transform our text data into numerical transform array. Now Skarn feature extraction library have different vectorizer which will transform text data to a matrix or array. In

this Project I have used Count Vectorizer to transform text data into transform Metrix. You can see in below diagram Count vectorize convert text into this type of matrix.

| | big | count | create | dataset | different | features | hello | is | james | my | name | notebook | of | python | this | to | try | trying | vectorizer | words |
|---|-----|-------|--------|---------|-----------|----------|-------|----|-------|----|------|----------|----|--------|------|----|-----|--------|------------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Figure 4:

4.4 Training The Model:

Now, After Vectorizing Messages we can train the model to predict the messages . For classification I have used SKLearn classifier which will train the model. Many deep learning techniques, such as Nave Bias, Linear Support Vector Classifier, K-Nearest Neighbor classifier, Random Forest classifier and Decision Tree classification, have been tested. Nave Bias and Linear SVC are best fit to this problem. Therefore, in this project I have used Linear SVC which will give 97% accuracy for the prediction.

5. Results:

Here I'm attaching photographs of testing accuracy and classification report of different classifier.

- 1) Linear SVC: 96% Accuracy

| | | | | |
|------------------------------------|-----------|--------|----------|---------|
| Linear SVC (Accuracy Score): 0.968 | | | | |
| | precision | recall | f1-score | support |
| Non-Spam | 0.92 | 1.00 | 0.96 | 49 |
| Spam | 1.00 | 0.95 | 0.97 | 76 |
| accuracy | | | 0.97 | 125 |
| macro avg | 0.96 | 0.97 | 0.97 | 125 |
| weighted avg | 0.97 | 0.97 | 0.97 | 125 |

Figure 5:Classification Matrix of SVC

2) Multinomial Naïve Bias : 88% Accuracy

| | | | | |
|-------------------------------------------|-----------|--------|----------|---------|
| MultiNominal Bias (Accuracy Score): 0.888 | | | | |
| | precision | recall | f1-score | support |
| Non-Spam | 0.78 | 1.00 | 0.88 | 49 |
| Spam | 1.00 | 0.82 | 0.90 | 76 |
| accuracy | | | 0.89 | 125 |
| macro avg | 0.89 | 0.91 | 0.89 | 125 |
| weighted avg | 0.91 | 0.89 | 0.89 | 125 |

Figure 6 Classification Matrix of Multinomial Bias

3) Random Forest Classifier :81% Accuracy

| | | | | |
|----------------------------|-----------|--------|----------|---------|
| RFC(Accuracy Score): 0.816 | | | | |
| | precision | recall | f1-score | support |
| Non-Spam | 0.68 | 1.00 | 0.81 | 49 |
| Spam | 1.00 | 0.70 | 0.82 | 76 |
| accuracy | | | 0.82 | 125 |
| macro avg | 0.84 | 0.85 | 0.82 | 125 |
| weighted avg | 0.87 | 0.82 | 0.82 | 125 |

Figure 7: Classification Matrix of RFC

4) K-nearest neighbor :80% Accuracy

| | | | | |
|----------------------------|-----------|--------|----------|---------|
| KNN(Accuracy Score): 0.808 | | | | |
| | precision | recall | f1-score | support |
| Non-Spam | 0.67 | 1.00 | 0.80 | 49 |
| Spam | 1.00 | 0.68 | 0.81 | 76 |
| accuracy | | | 0.81 | 125 |
| macro avg | 0.84 | 0.84 | 0.81 | 125 |
| weighted avg | 0.87 | 0.81 | 0.81 | 125 |

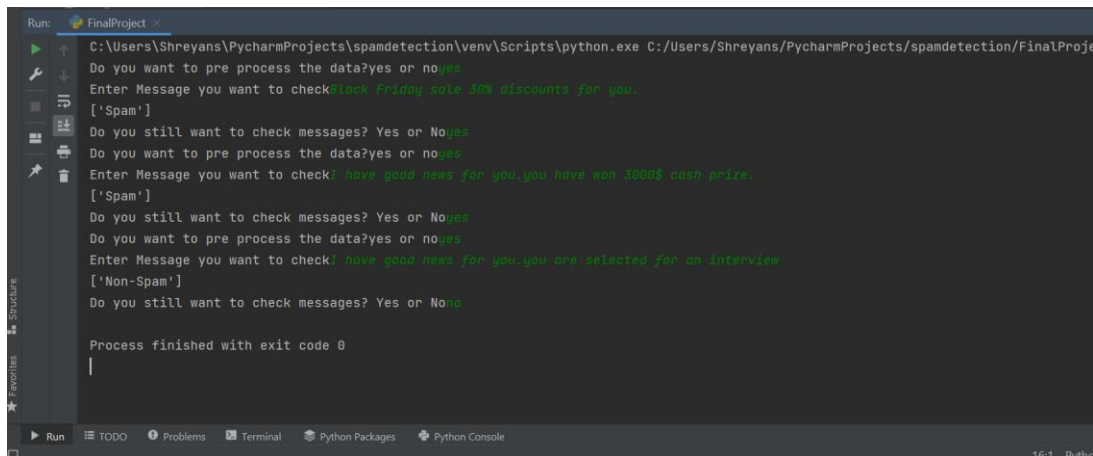
Figure 8: Classification Matrix of KNN

5) Decision Tree :80% Accuracy

| | | | | |
|--------------------------|-----------|--------|----------|---------|
| DTC(Accuracy Score): 0.8 | | | | |
| | precision | recall | f1-score | support |
| Non-Spam | 0.66 | 1.00 | 0.80 | 49 |
| Spam | 1.00 | 0.67 | 0.80 | 76 |
| accuracy | | | 0.80 | 125 |
| macro avg | 0.83 | 0.84 | 0.80 | 125 |
| weighted avg | 0.87 | 0.80 | 0.80 | 125 |

Figure 9: Classification Matrix of DTC

6) System Screenshots:



```
Run: FinalProject x
C:\Users\Shreyans\PycharmProjects\spamdetection\venv\Scripts\python.exe C:/Users/Shreyans/PycharmProjects/spamdetection/FinalProje
Do you want to pre process the data?yes or noyes
Enter Message you want to check!black Friday sale 30% discounts for you.
['Spam']
Do you still want to check messages? Yes or Noyes
Do you want to pre process the data?yes or noyes
Enter Message you want to check! have good news for you,you have won 3000$ cash prize.
['Spam']
Do you still want to check messages? Yes or Noyes
Do you want to pre process the data?yes or noyes
Enter Message you want to check! have good news for you,you are selected for an interview
['Non-Spam']
Do you still want to check messages? Yes or Nonno

Process finished with exit code 0
|
```

Figure 10: User Interface of Project

6. Conclusion and Future Work:

Email and message communication have become a vital element of people's communications as the speed of business and online work has increased significantly. Spam and advertising emails are becoming more prevalent as the use of email communication grows, which will result in a rise in online fraud as a result. So, these systems are quite beneficial in preventing and detecting this kind of fraudulent conduct, and users may work stress-free. This system has a 96 percent accuracy rate for detecting spam communications, which is extremely outstanding. We can train a data model and predict user input messages using a dataset of spam and non-spam classified messages. We have seen that different classifier and their usage in predication of messages. After applying and testing all the classifier I found Linear SVC is best approach for this problem.

After successfully predicting if a user input message is spam or not, I will now work on creating a User Interface that will store spam messages and save those messages into a train dataset, which will be used to further improve the accuracy of the system in the future.

7. References:

- [1]Kaggle dataset of Spam message: <https://www.kaggle.com/datatattle/email-classification-nlp>
- [2]Count Vectorizer : <https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c>
- [3]Stemming and Lemmatization :
https://www.tutorialspoint.com/natural_language_toolkit/natural_language_toolkit_tokenizing_text
- [4]MultiClass text classification :
<https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f>
- [5]Sci-kit Learn classifier:
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
- [6]State of art:
<https://towardsdatascience.com/overview-state-of-the-art-machine-learning-algorithms-per-discipline-per-task-c1a16a66b8bb>
- [7] Data frame methods:
<https://www.geeksforgeeks.org/python-pandas-dataframe-add/>
- [8] Python Machine Learning Book:
Author: WEI-MENG LEE , Publication by:John wiley & Sons,Inc.