Email Spam Detection Model

Contributors :- Pratik Datey, Shruthy Radhakrishan

Github Link :-https://github.com/pratikdatey/Email-Spam-Classifier

---

ABSTRACT

Email is the worldwide use of communication applications. It is because of the ease of use and faster than other communication applications. However, its inability to detect whether the mail content is either spam or ham degrades its performance. Nowadays, a lot of cases have been reported regarding stealing of personal information or phishing activities via email from the user. This project will discuss how machine learning helps in spam detection.

Machine learning is an artificial intelligence application that provides the ability to automatically learn and improve data without being explicitly programmed. Binary classifier will be used to classify the text into two different categories; spam and ham. The algorithm will predict the score more accurately. The objective of developing this model is to detect and score words faster and accurately.

INTRODUCTION

Technology has become a vital part of life in today's time. With each passing day, the use of the internet increases exponentially, and with it, the use of email for the purpose of exchanging information and communicating has also increased, it has become second nature to most people. While e-mails are necessary for everyone, they also come with unnecessary, undesirable bulk mails, which are also called Spam Mails.

- **Email :**

  Electronic mail (email) is a messaging system that electronically transmits messages across computer networks. Anyone is free to use email services through Gmail, Yahoo or people can even register with an Internet Service Provider (ISPs) and be provided with an email account. Only an internet connection is required, otherwise being a free service.

- **Spam :**

  Bulk mails that are unnecessary and undesirable can be classified as Spam Mails. These spam emails hold the power to corrupt one's system by filling up inboxes, degrading the speed of their internet connection.

- **Spam Detection :**

  Many spam detection techniques are being used now-a- days. The methods use filters which can prevent emails from causing any harm to the user. The contributions and their weaknesses have been identified. There are several methods that are accessible to spam, for example location of sender, its contents, checking IP address or space names. Spammers use refined variations to avoid spam identification.

PROBLEM STATEMENT

A tight competition between filtering methods and spammers is going on per day, as spammers began to use tricky methods to overcome the spam filters like using random sender addresses or appending random characters at the beginning or end of mail's subject line. There is a lack of machine learning that focuses on the model development that can predict the activity. Spam is a waste of time to the user since they have to sort the unwanted junk mail and it consumes storage space and communication bandwidth. Rules in other existing systems must be constantly updated and maintained, making it more of a burden to some users and it is hard to manually compare the accuracy of classified data.

MARKET/ CUSTOMER/ BUSINESS NEED ASSESSMENT

Nearly 85% of all emails are spam. That translates into an average daily volume of 122.33 billion messages globally. The number of daily spam messages oscillates regularly, and the latest spam traffic statistics show that it’s currently declining. Between June 2020 and January 2021, the average daily spam volume dropped from 316.39 billion to just over 122 billion.

A spam filter is necessary for the establishment of a safe and efficient working environment. Spam filters detect unsolicited, unwanted, and virus-infected email (called spam) and stop it from getting into email inboxes. Internet Service Providers (ISPs) use spam filters **to make sure they aren't distributing spam**. Although no spam filtering solution has ever been able to assure 100%

protection against spam and unwanted emails, they still form an indispensable part of any business email system. There are various ways in which spam filters function. Any spam filter solution that's the most suitable for companies' working mechanisms and proves most beneficial for your objectives is undoubtedly the best one for you.

EXTERNAL SEARCH (Information and Data Analysis)

These are some of the sources I visited for more information and need for an Email spam model.

- Statistics about Email spam filter
- https://www.duocircle.com/content/spam-filter
- What is a spam filter?

DATASET DESCRIPTION :

I am going to use this **Dataset** for my code implementation for this report.

The dataset contains the Email **text** feature and labels. In the data '**0**' shows email is '**ham**' and '**1**' shows mail text is '**spam**'. From this data we have to create a model who predicts the Email is spam or ham.

First import the basic libraries for data preprocessing

```
: import numpy as np
  import nltk
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns
  import re
  from nltk.corpus import stopwords
  from nltk.stem import PorterStemmer,WordNetLemmatizer
  from sklearn.preprocessing import LabelEncoder
  from sklearn.model_selection import train_test_split
  from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
  from sklearn.linear_model import LogisticRegression
  from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
  import warnings
  warnings.filterwarnings(action = 'ignore')
```

Fig.No. 01 - Import Libraries

BENCHMARKING

Most email services, such as Gmail, Yahoo Mail, Microsoft Outlook, and Apple Mail have algorithms that filter out spam and junk mail by tucking them away in a folder. It directly sends that mail to the spam folder. It is used mostly in companies as well as in our personal life also. It will help to prevent companies or personal data from hackers or attackers.

APPLICABLE PATENTS

- **SpamTitan Email Security and Protection**
- **ZEROSPAM**
- **Spambrella**
- **MailChannels**

APPLICABLE REGULATION(Government and Environmental)

1. Data collection and Privacy of Regulations of Customers.
2. Patents on ML algorithms developed.
3. Laws related to privacy for collecting data from users Protection/ ownership regulations.
4. Ensuring open-source, academic and research community for an audit of Algorithms.
5. Review of existing work authority regulations.

BUSINESS OPPORTUNITY

Spam messages accounted for 45.37 percent of email traffic in December 2021. For some business professionals, the risk of spam is too much to rely on simple tricks and tips to tackle it. There are a wide range of anti-spam programs available that can simplify the process for you, and offer a positive return on investment by saving precious working hours fighting unwanted and unsafe emails.

Spam filters can help you by preventing unwanted emails from entering your inbox. Spam filters are also helpful because they provide an extra layer of security for your network. Email is a popular attack vector for hackers and other malicious actors seeking to infect computers with malware. This can be effective if the spam filter is not updated with the correct information on a regular basis. Therefore, it is important to make sure your spam filter has adequate spam intelligence. If it does, it can block hundreds or thousands of spam emails every month.

CONCEPT DEVELOPMENT

Machine learning algorithms use statistical models to classify data. In the case of spam detection, a trained machine learning model must be able to determine whether the sequence of words found in an email are closer to those found in spam emails or safe ones.

Step1 : Data collection

The dataset contained in a corpus plays a crucial role in assessing the performance of any spam filter. Many open sources data sets are freely available in the public domain.

Dataset Link

```
# Import data
data=pd.read_csv('spam_ham_dataset.csv')
data=data.drop(['Unnamed: 0','label'],axis=1)

print('shape of data',data.shape) #Checking shape of data
data.head()
```

```
shape of data (5171, 2)
```

	text	label_num
0	Subject: enron methanol ; meter # : 988291\r\n...	0
1	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	Subject: photoshop , windows , office . cheap ...	1
4	Subject: re : indian springs\r\nthis deal is t...	0

```
data.dtypes
```

```
text         object
label_num     int64
dtype: object
```

Fig.No. 02 - Import Data

Step 2 : Data Preprocessing

At the preprocessing step, we mainly perform tokenization of mails. **Tokenization** Is a process where we break the content of an email into words and transform big messages into a sequence of representing symbols termed tokens. These tokens are extracted from email body, header,subject and image We know that the more number of attributes the more time of complexity model. These attributes can be huge and hence techniques like **stemming, lemmatization, noise removal, stopwords removal** can be used.

```
#Checking null values
data.isna().sum()
```

```
text         0
label_num    0
dtype: int64
```

```
ps=PorterStemmer()
wl=WordNetLemmatizer()
```

```
tfid=TfidfVectorizer(ngram_range=(1,1))
```

```
corpus=[]
for i in range(0,len(data['text'])):
    msg=re.sub('[^a-zA-Z]',' ',data['text'][i])
    msg=msg.lower()
    msg=msg.split()
    msg=[wl.lemmatize(word) for word in msg if word not in set(stopwords.words('english'))]
    msg=' '.join(msg)
    corpus.append(msg)
```

Fig.No. 03 - Data Preprocessing

Step 3: Model Building

After splitting the data into training and testing, we can use different types of classification algorithms. In that **MultinomialNB, Passive Aggressive Classifier** these two algorithms perform very well on text data.

```
trainx,testx,trainy,testy=train_test_split(corpus,y,test_size=0.3,random_state=42)
```

```
trainx=tfid.fit_transform(trainx)
testx=tfid.transform(testx)
```

```
print(trainx.shape)
print(trainy.shape)
print(testx.shape)
print(testy.shape)
```

```
(3619, 35274)
(3619,)
(1552, 35274)
(1552,)
```

Fig.No. 04 - Model Building

LogisticRegression

```
logreg=LogisticRegression()
logreg.fit(trainx,trainy)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
logreg_predict=logreg.predict(testx)
logreg_predict1=logreg.predict(trainx)
```

Fig.No. 05 - Logistic Model Building

MultinomialNB

```
from sklearn.naive_bayes import MultinomialNB
nb=MultinomialNB()
nb.fit(trainx,trainy)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
nb_pred=nb.predict(testx)
nb_pred1=nb.predict(trainx)
```

Fig.No. 06 - Multinomial Naive Bayes Model Building

PassiveAggressiveClassifier

```
from sklearn.linear_model import PassiveAggressiveClassifier
```

```
pc=PassiveAggressiveClassifier(random_state=42,max_iter=1000)
pc.fit(trainx,trainy)
```

```
▾ PassiveAggressiveClassifier
PassiveAggressiveClassifier(random_state=42)
```

```
pc_pred=pc.predict(testx)
pc_pred1=pc.predict(trainx)
```

Fig.No. 07 - Passive Aggressive Model Building

Step 4: Performance Analysis

Now algorithms Is ready so we must check performance of the model

```
print(accuracy_score(testy,logreg_predict))
print(accuracy_score(trainy,logreg_predict1))
```

```
0.9858247422680413
0.995855208621166
```

```
confusion_matrix(testy,logreg_predict)
```

```
array([[1105,   16],
       [   6,  425]], dtype=int64)
```

```
print(classification_report(testy,logreg_predict))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1121
           1       0.96      0.99      0.97       431

    accuracy                           0.99      1552
   macro avg       0.98      0.99      0.98      1552
weighted avg       0.99      0.99      0.99      1552
```

Fig.No. 08 - Logistic model Performance Analysis

```
print(accuracy_score(testy,nb_pred))
print(accuracy_score(trainy,nb_pred1))
```

```
0.9201030927835051
0.962420558165239
```

```
confusion_matrix(testy,nb_pred)
```

```
array([[1121,    0],
       [ 124,  307]], dtype=int64)
```

```
print(classification_report(testy,nb_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      1.00      0.95      1121
           1       1.00      0.71      0.83       431

    accuracy                           0.92      1552
   macro avg       0.95      0.86      0.89      1552
weighted avg       0.93      0.92      0.92      1552
```

Fig.No. 09 - Multinomial Naive Bayes Model Performance Analysis

```
print(accuracy_score(testy,pc_pred))
print(accuracy_score(trainy,pc_pred1))
```

```
0.9884020618556701
1.0
```

```
confusion_matrix(testy,pc_pred)
```

```
array([[1112,    9],
       [   9,  422]], dtype=int64)
```

```
print(classification_report(testy,pc_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1121
           1       0.98      0.98      0.98       431

    accuracy                           0.99      1552
   macro avg       0.99      0.99      0.99      1552
weighted avg       0.99      0.99      0.99      1552
```

Fig.No. 10 - Passive Aggressive Model Performance Analysis

STEP 1: PROTOTYPE SELECTION

A) Feasibility

This project can be developed and deployed within a few years as SaaS(Software as a Service) for anyone to use in personal and professional life.

B) Viability

With each passing day, the use of the internet increases exponentially, and with it, the use of email for the purpose of exchanging information and communicating has also increased. At this increasing rate this service will be useful and able to survive in the long term future.

C) Monetization

This service is directly monetizable as it can be directly released as a service on completion which can be used in any companies (Startup to Multinational companies)

STEP 2: PROTOTYPE DEVELOPMENT

Github Link: https://github.com/pratikdatey/Email-Spam-Classifier

STEP 3: BUSINESS MODELING

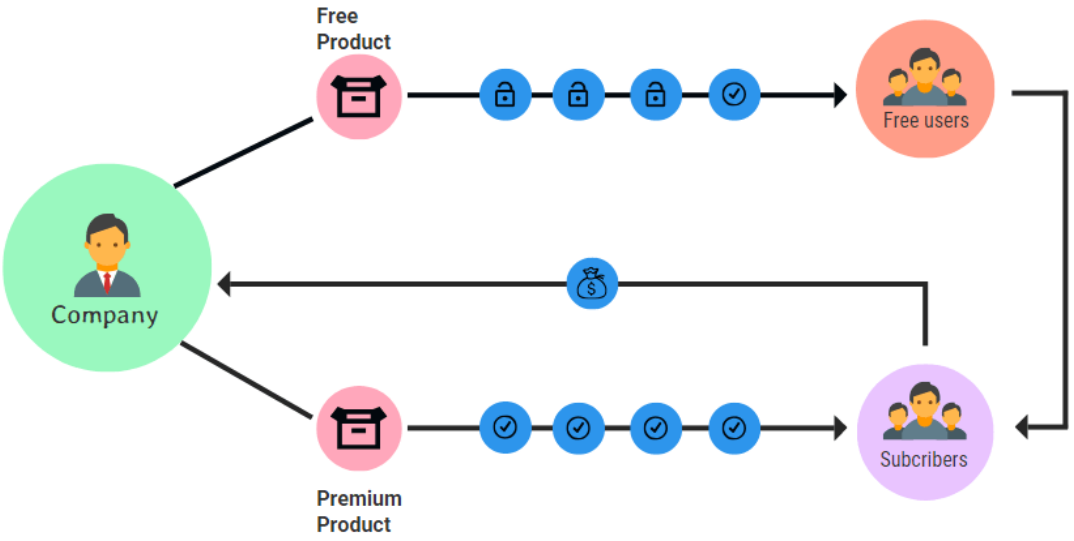

Fig.No. 11 - Subscription Based Model

For this service model , the best benefit is to use a Subscription Based Model, Where at the beginning we give free services to engage customer retention and increase our customer count. Later after some period of time it will be charged some subscription fees to use further service.

The major problem is converting free users into paid users. We will provide 2-3 months free service to gain experience of our service. Afterwards if consumers want to continue our service, they have to pay subscription charges. In this subscription business model we take subscription charges and give 1or 2 year access to our service.

STEP 4: FINANCIAL MODELING

Financial Equation:

$Y = X * (1 + R)^t$

Where,
Y= Total Profit
X = Price of product/ Service
R = Growth rate
T = Time Interval (In Year)

Let's assume, the price of our service/product is 1000 Rs.
And Growth rate of our service is 2.5% in 1 year .

Calculation :-
$Y= (1000)*(1+2.5)^1$
Y= 3500

The Total Profit of our service per user goes around 3500 Rs in 1 year.

CONCLUSION

The performance of a classification technique is affected by the quality of data source. Irrelevant and redundant features of data not only increase the elapsed time, but also may reduce the accuracy of detection. Each algorithm has its own advantages and disadvantages. As stated before, supervised ML is able to separate messages and classify the correct categories efficiently. It is also able to score the model. For instance, Gmail's interface is using an algorithm based on a machine learning program to keep their users' inbox free of spam messages.

From this project, it can be concluded that a machine learning algorithm is one of the important parts in order to create spam detection applications. To make it more efficient, improvements need to be implemented in future.