FLIP ROBO

Email Spam Detection

Submitted by:

Lahari S.K

INTRODUCTION

Spam emails are messages randomly sent to multiple addresses of all types of group but mostly by lazy advertisers and criminals leading to phishing sites.

Spam emails can not only be annoying but also dangerous. It is very important to detect spam emails and segregate them because they may cause following problems

- Critical mails are missed or delayed
- Millions of computers compromised
- Billions of dollars lost worldwide
- Identity theft
- Spam can crash mail servers and fill up hard drives.

Objective of this project is to give customers the knowledge of relevant emails and the fake emails.

Data Description

The dataset contains 2893 rows and 3 columns. The columns are namely message, subject and the label. The label column is the target column. The dataset is in the form of CSV.

Out of 2893 instances 2412 is spam and 481 is ham mail. That is 83% of the given instances belongs to spam mails and the remaining are ham mails.

```
#Reading data
data
```

	subject	message	label
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0
1	NaN	lang classification grimes , joseph e . and ba...	0
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0
3	risk	a colleague and i are researching the differin...	0
4	request book information	earlier this morning i was on the phone with a...	0
...	...	...	...
2888	love your profile - ysuolvpv	hello thanks for stopping by ! ! we have taken...	1
2889	you have been asked to join kiddin	the list owner of : " kiddin " has invited you...	1
2890	anglicization of composers ' names	judging from the return post , i must have sou...	0
2891	re : 6 . 797 , comparative method : n - ary co...	gotcha ! there are two separate fallacies in t...	0
2892	re : american - english in australia	hello ! i ' m working on a thesis concerning a...	0

2893 rows × 3 columns

Data Pre-processing Done

Creating additional columns to know length

Let’s create two more columns known namely “sublength” and “messagelength” in order to calculate the length of the subject and message.

```
data["sublength"]=data.subject.str.len()
data["messagelength"]=data.message.str.len()
```

```
data
```

	subject	message	label	sublength	messagelength
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0	39.0	2856
1	NaN	lang classification grimes , joseph e . and ba...	0	NaN	1800
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0	50.0	1435
3	risk	a colleague and i are researching the differin...	0	4.0	324
4	request book information	earlier this morning i was on the phone with a...	0	24.0	1046
...	...	...	...	...	...
2888	love your profile - ysuolvpv	hello thanks for stopping by ! ! we have taken...	1	28.0	262
2889	you have been asked to join kiddin	the list owner of : " kiddin " has invited you...	1	34.0	2163
2890	anglicization of composers ' names	judging from the return post , i must have sou...	0	34.0	1039
2891	re : 6 . 797 , comparative method : n - ary co...	gotcha ! there are two separate fallacies in t...	0	54.0	2949
2892	re : american - english in australia	hello ! i ' m working on a thesis concerning a...	0	36.0	700

2893 rows × 5 columns

Converting to lower case

Let’s convert the subject and message column into lower case using the lower() function.

```
#Converting subject and messageto lower
data['subject']=data['subject'].str.lower()
data['message']=data['message'].str.lower()
```

```
data
```

	subject	message	label	sublength	messagelength
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0	39.0	2856
1	NaN	lang classification grimes , joseph e . and ba...	0	NaN	1800
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0	50.0	1435
3	risk	a colleague and i are researching the differin...	0	4.0	324
4	request book information	earlier this morning i was on the phone with a...	0	24.0	1046
...	...	...	...	...	...
2888	love your profile - ysuolvpv	hello thanks for stopping by ! ! we have taken...	1	28.0	262
2889	you have been asked to join kiddin	the list owner of : " kiddin " has invited you...	1	34.0	2163
2890	anglicization of composers ' names	judging from the return post , i must have sou...	0	34.0	1039
2891	re : 6 . 797 , comparative method : n - ary co...	gotcha ! there are two separate fallacies in t...	0	54.0	2949
2892	re : american - english in australia	hello ! i ' m working on a thesis concerning a...	0	36.0	700

Replacing regular expression

Lets make use of regular expressions to replace some strings.

```
#Replacing emailaddresses with email
data["message"]=data["message"].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$','email_address')

#Replacing webaddress
data["message"]=data["message"].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','webaddress')

#Replacing moneysymbols
data["message"]=data["message"].str.replace(r'£|/$','dollars')

#Replacing phonenumber,paranthesis,spaces,dashes,nospaces)
data['message']=data['message'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phone number')

#Replacing phone number with phone
data['message']=data['message'].replace(r'\d+(\.\d+)?','number')
```

Removing punctuation

Lets remove all the punctuation marks .

```
#Removing punctuation marks
data['message']=data['message'].str.replace(r'[^\w\d\s]',' ')

#Removing whitespaes between terms with a single space
data['message']=data['message'].str.replace(r'\s+',' ')

#Removing tailing and heading whitespaces
data['message']=data['message'].str.replace(r'^\s+|\s+?$',' ')

```

Removing stop words

Lets remove all the stop words and see the length after pre-processing steps.

```
from nltk.corpus import stopwords
import string
import nltk

stop_words=stopwords.words("english")

data['message']=data['message'].apply(lambda x:' '.join(term for term in x.split() if term not in stop_words))

```

```
#New column after cleasing
data["clean_length_message"]=data.message.str.len()
data["clean_length_subject"]=data.subject.str.len()
```

```
#Total length removal
print("Original length of message",data.messagelength.sum())
print("After cleansing",data.clean_length_message.sum())

print("Original legth ofsubject",data.sublength.sum())
print("Aftre cleansing ",data.clean_length_subject.sum())
```

```
Original length of message 9070005
After cleansing 6308430
Original legth ofsubject 91647.0
Aftre cleansing  84741.0
```

Model/s Development and Evaluation

Algorithms Used

1.Naive Bayes

```
naive.fit(X_train,Y_train)
y_pred=naive.predict(x_test)
print("Finle score -->",accuracy_score(y_test,y_pred))
```

```
Finle score --> 0.835635359116022
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      1.00      0.91       585
           1       1.00      0.14      0.25       139

    accuracy                           0.84       724
   macro avg       0.92      0.57      0.58       724
weighted avg       0.86      0.84      0.78       724
```

2.SVM

```
from sklearn import svm
```

```
svm=svm.SVC()
svm.fit(X_train,Y_train)
y_pred=svm.predict(x_test)
print("Finle score -->",accuracy_score(y_test,y_pred))
```

```
Finle score --> 0.9737569060773481
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       585
           1       1.00      0.86      0.93       139

    accuracy                           0.97       724
   macro avg       0.98      0.93      0.96       724
weighted avg       0.97      0.97      0.97       724
```

The best model is SVM.