Spam E-mail Detection

Submitted by:

HEMANT PATAR

ACKNOWLEDGMENT

*I would like to express my special thanks of gratitude to Datatarained & FlipRobo who gave me the golden opportunity to do this internship project on the topic (*spam E-mail detection*), which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them.*

The sample data is provided to us from FlipRobo's client database.

Training from Datatrained helped me in completing the project.

INTRODUCTION

- Business Problem

Spam E-mails can not only be annoying but also dangerous to users.

- Unwanted E-mails creates junk in inbox.
- Critical E-mail messages are missed or delayed.
- Computer or LAN may be compromised.
- Identity theft
- Critical data may be compromised.
- Spam can crush E-mail servers.
- Monetary theft by lucrative spam E-mails.

Spam E-mails are messages randomly sent to multiple addresses by all sort of groups, but mostly lazy advertisers & criminals who wish to lead you to phishing sites.

Spam prevention is often neglected, although some simple measures can dramatically reduce the risk of getting compromised.

Here in this project, we are going to create an automated spam detection model.

- It increases security and control.
- It provides sensitivity to the client and adapts well to the future spam techniques.
- It considers a complete message instead of single words with respect to its organisation.
- It reduces network resource costs.
- It also reduces IT administration costs.

Let’s start

- Data Sources and their formats

  The sample data is provided to us from FlipRobo’s client database.

```
sed=pd.read_csv('messages.csv',encoding='latin1')
sed
```

	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

Let’s get some basic info.

Here we see that the dataset contains 2893 entries with no nan values in features message & label.

```
sed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2893 entries, 0 to 2892
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   subject  2831 non-null   object
 1   message  2893 non-null   object
 2   label    2893 non-null   int64 
dtypes: int64(1), object(2)
memory usage: 67.9+ KB
```

```
print("shape = >",sed.shape)
```

```
shape = > (2893, 3)
```

Here we check the spam and non-spam count and percentage in the dataset.

```
print("non spam and spam counts","\n",sed.label.value_counts())
```

```
non spam and spam counts
 0    2412
1     481
Name: label, dtype: int64
```

```
print("spam ratio =","\n",round(len(sed[sed["label"]==1])/len(sed["label"]),2)*100,"%")
print("non spam ratio =","\n",round(len(sed[sed["label"]==0])/len(sed["label"]),2)*100,"%")
```

```
spam ratio =
 17.0 %
non spam ratio =
 83.0 %
```

Now we create a new feature length to check the length of each message.

```
sed['length'] = sed.message.str.len()
sed.head(5)
```

	subject	message	label	length
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0	2856
1	NaN	lang classification grimes , joseph e . and ba...	0	1800
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0	1435
3	risk	a colleague and i are researching the differin...	0	324
4	request book information	earlier this morning i was on the phone with a...	0	1046

Convert every alphabet to lower case in message.

```
sed['message']=sed['message'].str.lower()
```

Here we pre-process the messages by replacing email address, web address, phone numbers, numeric characters and currency symbols.

```
sed['message'] = sed['message'].str.replace('^.+@[^\.].*\.[a-z]{2,}$','emailaddress')
sed['message'] = sed['message'].str.replace('^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','webaddress')
sed['message'] = sed['message'].str.replace('^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phonenumber')
sed['message'] = sed['message'].str.replace('\d+(\.\d+)?','numbr')
sed['message'] = sed['message'].str.replace('£|\$','dollers')
```

Here we pre-process the punctuations.

```
sed['message'] = sed['message'].str.replace('[^\w\d\s]',' ')
sed['message'] = sed['message'].str.replace('\s+',' ')
sed['message'] = sed['message'].str.replace('^\s+|\s+?$','')
sed['message'] = sed['message'].str.replace('\_',' ')
```

Tokenization is a critical step in NLP. We cannot simply jump into the model building part without cleaning the text first. It is done by removing stopwords.

```
stop_words = set(stopwords.words('english') + ['u','ü','ur','4','2','im','dont','doin','ure'])

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

Now we create another feature named clean_length to compare the cleaned message length from unprocessed message length.

```
sed['clean_length'] = sed.message.str.len()
sed.head()
```

	subject	message	label	length	clean_length
0	job posting - apple-iss research center	content length numbr apple iss research center...	0	2856	2047
1	NaN	lang classification grimes joseph e barbara f ...	0	1800	1454
2	query : letter frequencies for text identifica...	posting inquiry sergei atamas satamas umabnet ...	0	1435	1064
3	risk	colleague researching differing degrees risk p...	0	324	210
4	request book information	earlier morning phone friend mine living south...	0	1046	629

Almost one third amount of unwanted data was either cleaned or processed.

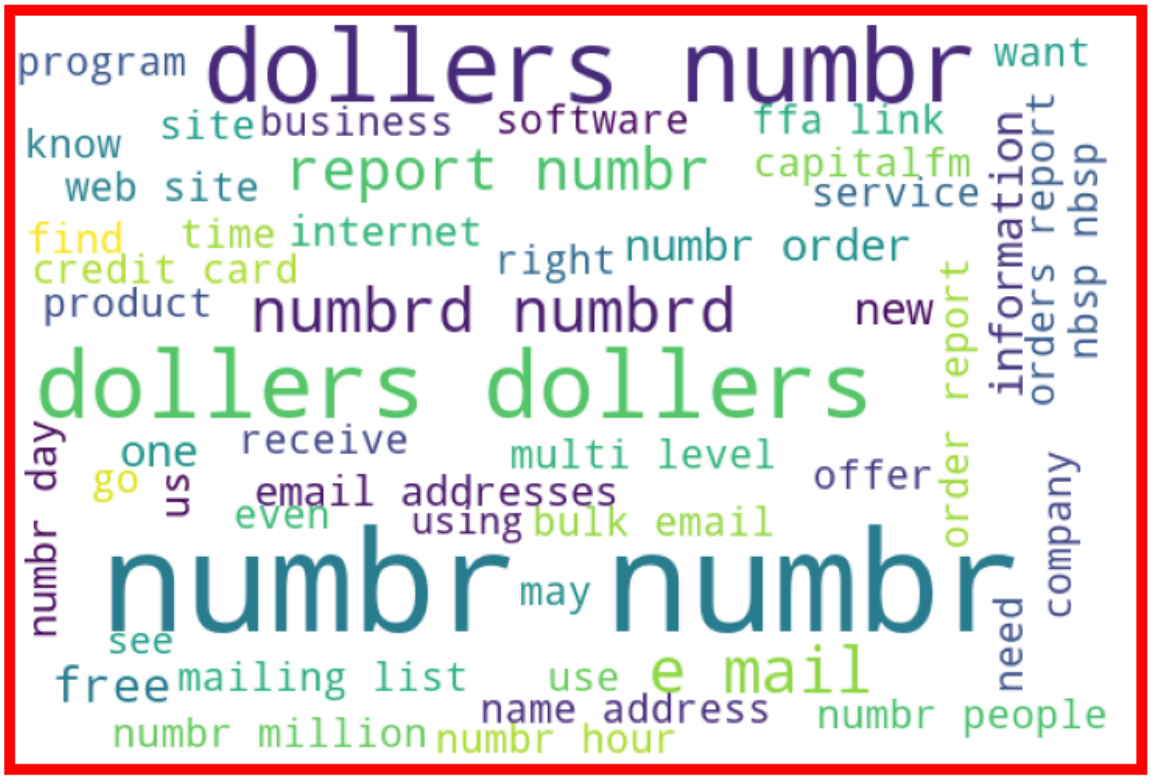
```
print('Original Length', sed.length.sum())
print('Clean Lenght', sed.clean_length.sum())
```

```
Original Length 9344743
Clean Lenght 6605523
```

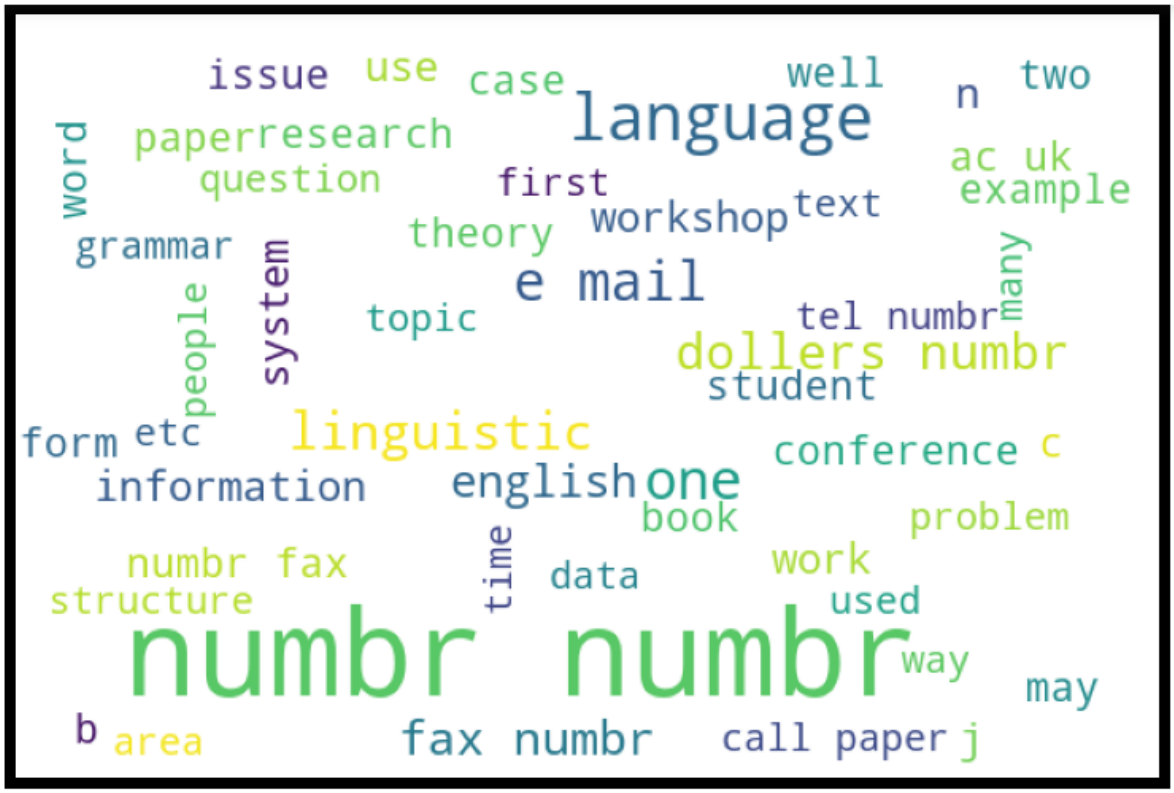
Let’s do a final visual check on the processed dataset.

```
# sed.to_csv('file1.csv')
```

Let’s visualise some popular terms in spam messages using word cloud.



Now visualise some popular terms in non-spam messages using word cloud.



We have our processed data ready for modelling.

We use Tfidfvectorizer to convert the alphabetical messages into numeric form.
After vectorizing the messages, we fit and feed it as input for model and feature label as output.
Here I have used two algorithms naive bias and SVM to compare the accuracy percentage.
In naive bias the accuracy percentage raised from 83% to max 88% even after minimizing the test size in train test split.
But in SVM the accuracy percentage went up to 98% showing better result than naïve bias, the further process is done in svm.

```
tf_vec = TfidfVectorizer()

# naive = MultinomialNB()

SVM = SVC(C=1.0, kernel='linear', degree=3, gamma='auto')

features = tf_vec.fit_transform(sed['message'])
X = features
y = sed['label']
```

```
X_train,x_test,Y_train,y_test = train_test_split(X,y,random_state=42) #test_size=0.20 random_state=42 test_size=0.15

# naive.fit(X_train,Y_train)
# y_pred = naive.predict(x_test)

SVM.fit(X_train,Y_train)
y_pred = SVM.predict(x_test)

print('Final score = > ',accuracy_score(y_test,y_pred))
```

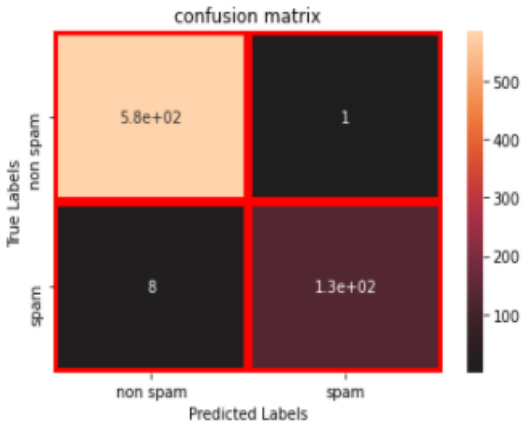
```
print('Final score = > ',accuracy_score(y_test,y_pred))
```

```
Final score = >  0.9875690607734806
```

Finally, we test and predict our svm model.

```
conf_mat = confusion_matrix(y_test,y_pred)
conf_mat
```

```
array([[584,   1],
       [  8, 131]], dtype=int64)
```



Our SVM spam detection model predicted 131 spam out of 139 spam E-mails. Only 9 were not predicted correctly 8 spams were detected not spam and 1 not spam was detected spam. And total 584 not spam E-mail detected correctly as non-spam E-mails.

From the above spam detection model one can find spam E-mails with 98% accuracy.

It was a good project full of learning's.
The project was saved as ipynb file & this pdf file as report.

THANK YOU