Data mining- Assignment 2: Classification

Anousheh Khajeh nassiri

May 1, 2018

This assignment aims to predict for each email whether it is legitimate or spam. First thing to do is to read the data. Reading data is done in "reading_data.ipynb" file. All the 9688 spam emails are read, their subjects and bodies are separated and each listed in a list. Then a data frame is constructed with 3 columns: Subject, Body and Label. We labeled all the spammed emails as 1 so all the labels in this data frame is 1.

	subject	body	label
0	you don _ t know how to get into search engine...	submitting your website in search engines may ...	1
1	sterling balance sheet strengthens underpriced...	secured data inc . (scre)'emerging leader in...	1
2	topcoat best m = eds here anyplace'	i asked a conductor one day at what time the't...	1
3	updates for clients sun , 03 jul 2005 .'	subject : updates for clients sun , 03 jul 200...	1
4	re : knjecmo , intrepid investors report for s...	dane dempsey ,'emerging enterprise solutions ,...	1

```
print(df.subject[0])
```

```
you don _ t know how to get into search engine results ?'
```

```
print(df.body[0])
```

```
submitting your website in search engines may increase'your online sales dramatically .'lf yo
u invested time and money into your website , you'simply must submit your website'oniine othe
rwise it wili be invisibie virtualiy , which means efforts spent in vain .'lf you want'people
to know about your website and boost your revenues , the oniy way to do'that is to'make your
site visible in places'where people search for information , i . e .'submit your'website in m
uitipie search engines .'submit your website oniine'and watch visitors stream to your e - bus
iness .'best regards ,'norasweeney _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ not interested . . . _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _'
```

Figure 1: Spam email data frame

Data frame is then located in a folder named "csv" for further usage. Same procedure is followed for reading the user's inboxes with the difference that this time they are all labeled 0 as they are non-spam emails.

	subject	body	label
0	re : cornhusker'	if the plants become an external counterparty ...	0
1	brandywine meter # : 981225 ; march , 2000 act...	there was no flow at meter 981225 for the mont...	0
2	southern union - 03 / 01 prod - austin spot de...	daren -'per janet , the price of $ 5 . 235 + ...	0
3	eastrans november first of the month nominations'	effective 11 / 1 / 00 deliveries to eastrans i...	0
4	calpine daily gas nomination (weekend)'	>'ricky a . archer'fuel supply'700 louisiana ,...	0

Figure 2: Not spam emails data frame

The last part is for reading streams. Obviously these are not labeled as we should be able to predict their labels to distinguish between spam and non-spam emails.

	subject	body
0	test our internet pharmacy , buy viagra and ot...	no visit to the doctor needed - safe and easy ...
1	hpl nom for may 16 , 2001'	(see attached file : hplno 516 . xls)'- hpln...
2	black marlin ua 4 meters'	michael ,'can you back date deals 83347 and 83...
3	read : this email will change your life'	dear homeowner ,'you have been pre - approved ...
4	special pricing on rxdrugs . to be precise , p...	our chernist - site provides customers a quick...

Figure 3: Not spam emails data frame

By cleaning the data, we prepare it for data analysis. This is a crucial step as data almost never comes in a way that does not need to be cleaned! In the "main_anousheh.ipynb " file, 3 data frames that are saved in a "csv" folder are read. Non-spams are attached to spams and together they form our data.

1 Preprocessing

In preprocess function, data is a bit cleaned for example "'s", "/"," -", "‘", digits and punctuations except "?" and "!" are removed and the string is converted to the lower case.

Stop words are the words we want to filter out before training the classifier. These are usually high frequency words that are not giving any additional information to our labeling. In fact, they actually confuse our classifier. In English, they could be the, is, at, which, and etc.

Stemming is the process for reducing inflected words to their word stem (base form). As an example, the classifier does not understand that the verbs "investing" and "invested" are the same, and treats them as different words with different frequencies. By stemming them, it groups the frequencies of different inflection to just one term in this case, "invest". In stemming process only the word's stem is left so it reduces the total variety of words. All the words are tokenized, and after the above changes, listed words are put in a sentence with a space between each. This preprocessing is done on both Subject and Body. Using English stop words for cleaning our data brought up the idea to observe the frequency of different languages in the emails. As shown below, most of the emails are in English.

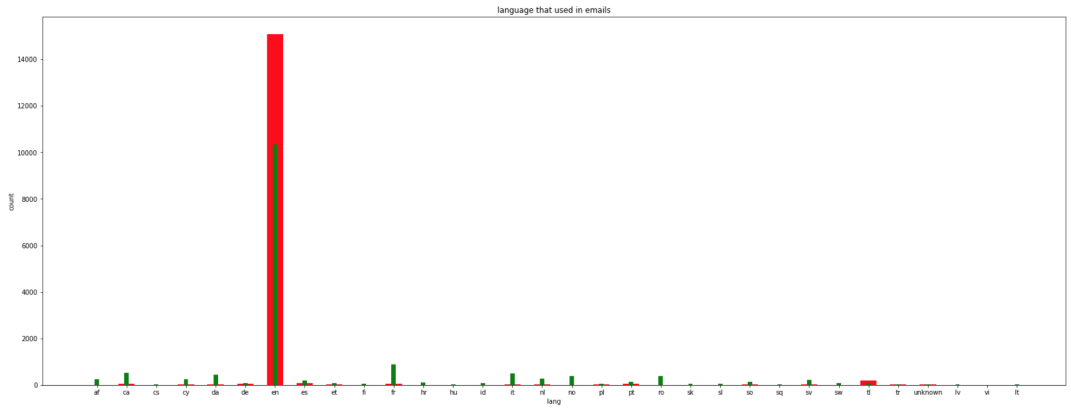

Figure 4: Most of the emails are in English

Red color is used to show the language used in the body of the emails and green is used for subject of emails. So, in the pre-processing step, the English stop words are removed,

punctuations are also removed except ! and ?. Stemming is performed and words have been tokenized.

2 Features

The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier. BoW is one of the basic and most commonly used methods for document representation due to its simplicity. It is essentially an algorithm that forms a vector by counting how many times each word appears in the document. It is needless to mention that Bag-of-words suffers from several limitations, for instance, it fails to capture relationships between words since it deals with each word individually. It also suffers from high dimensionality of representation and sparsity. A simple twist to BoW would be Tf-idf. Tf-idf stands for term frequency-inverse document frequency. Tf-idf captures the importance of a word in a document so unlike BoW, it doesnt emphasize a word more than its needed. The Tf-idf score (weight) of term t in plot p is given by: $\mathbf{w_{i,p}} = \mathbf{tf_{i,p}} \times \mathbf{log}\frac{\mathbf{N}}{\mathbf{df_i}}$ where $\mathbf{tf_{i,p}}$ is the number of times i-th term has appeared in plot p. Since the naive representation of $\mathbf{tf_{i,p}}$ causes bias towards long plots, as a given term has more chance to appear in longer plot, all $\mathbf{w_p}$ values are normalized as $\|\mathbf{w_p}\| = 1$. df_i is the number of plots containing the i-th term and $\mathbf{N}$ is the total number of plots. We have also tried Tf-idf as a more sophisticated version of BoW.

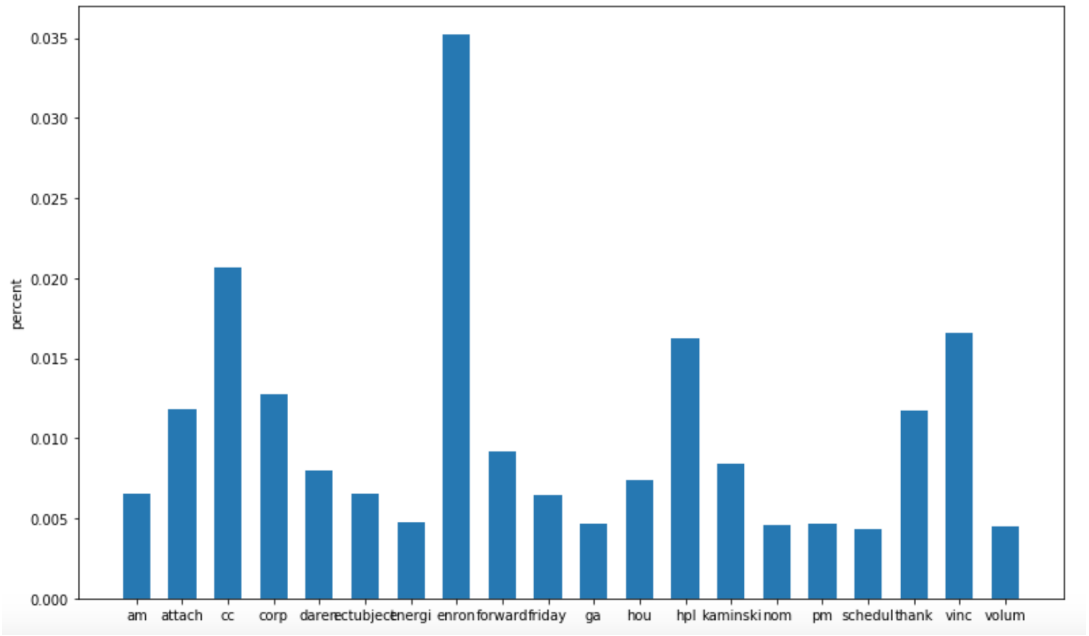

Figure 5: Top 20 features

We have a total amount of 102257 features. This high number of features is one of the downsides of using BoW. 102256 features are the unique words acquired from tf-idf and one column is for text length feature. 20 top features are depicted in the figure5. "enron" has the most percentage then "cc", "vinc" and "hpl" respectively have more percent. Another feature that we looked into was the length of text. It appeared that spam emails are longer than non-spam emails.

3 Training

The total number of emails including both inboxes and spams are 15766. We used 33% of it for test which will be 5203 emails. The remaining 77% of emails are used for training our model which will be 10563 emails. In logistic regression model we tuned various hyper parameters; c= 0.1, 1, 10, 30, 50. The accuracy of prediction with c=30 is the highest (0.978).

Prediction on our test dada has a high accuracy of 98%.

4 Prediction

For three classes of 0 for non spam, 1 for spam and 2 for suspicious emails we want to predict the probability based on random Gaussian data.

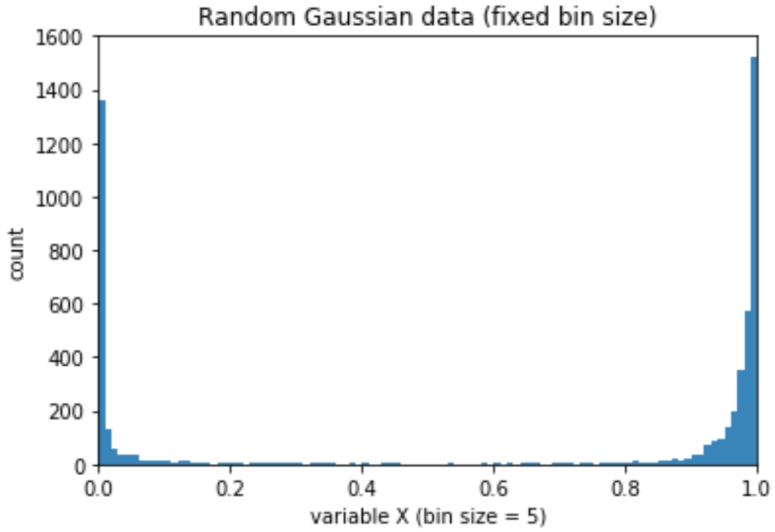

Figure 6: random Gaussian data

With a bit of approximation, if the probability is between [0, 0.19), we predict the email as non spam, if more than 0.85, we predict it as spam, and in between (0.19, 0.85], we predict it as suspicious. (As shown in figure 6)

As the output of counter indicated the number of emails are : (0: 1770, 1: 3200, 2: 233). Also depicted in figure 7.

After training we predict the labels of stream of our model. The final results of labels will be either 0,1 or 2. (Shown in figure 8)

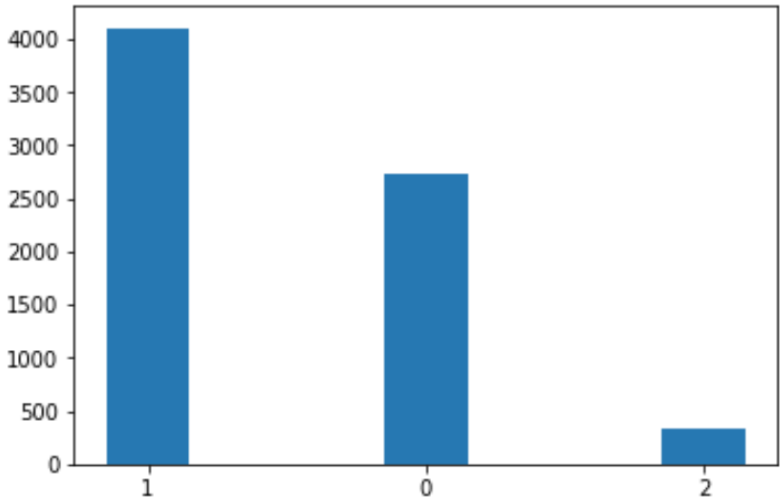

Figure 7: Counts

	id	subject	body	label
0	0	test internet pharmaci buy viagra med	visit doctor need safe easi 'b '' like email e...	1
1	1	hpl nom may	see attach file hplno xl hplno xl	0
2	2	black marlin ua meter	michael 'can back date deal start black marlin...	0
3	3	read email chang life	dear homeown 'you pre approv home low'fix rate...	1
4	4	special price rxdrug precis put buck back pocket	chernist site provid custom quick legitim acce...	1

Figure 8: Final result