Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3094 - Security Data Science

Laboratorio #2 - Detección de Spam

Andrea Amaya - 19357
Brandon Hernández - 19376
Guatemala, Ciudad de Guatemala 19 de febrero de 2023

Link repositorio: https://github.com/andreamalin/LAB02-SDS

Análisis Exploratorio

Para iniciar la manipulación de la información se observaron las primeras cinco filas de cada uno de los data sets:

	Body	Label
0	\r\nSave up to 70% on Life Insurance.\r\nWhy S...	1
1	1) Fight The Risk of Cancer!\r\nhttp://www.adc...	1
2	1) Fight The Risk of Cancer!\r\nhttp://www.adc...	1
3	##...	1
4	I thought you might like these:\r\n1) Slim Dow...	1

Figura 1: Primeros valores de completeSpamAssassin.csv

	Body	Label
0	Subject: stock promo mover : cwtd\r\n * * * ur...	1
1	Subject: are you listed in major search engine...	1
2	Subject: important information thu , 30 jun 20...	1
3	Subject: = ? utf - 8 ? q ? bask your life with...	1
4	Subject: " bidstogo " is places to go , things...	1

Figura 2: Primeros valores de enronSpamSubset.csv

Como se puede observar cuentan con las mismas columnas, por lo que se puede concatenar tal cual están. Además se observa que la columna Body está sucia y necesita limpieza. Al combinar ambos conjuntos de datos, se obtuvo un conjunto de datos con 16046 filas.

Luego de combinar los datos, se continuó por limpiar la data, en donde se siguieron estos pasos:

- Se eliminaron todas las columnas que fueran nulas, ya que si al contar con solo dos columnas, que falte una significa que no se podrá clasificar; las stepwords, caracteres especiales, espacios extra, links y todas las columnas que tuvieran *empty* como texto.
- Se pasaron todas las palabras a estado presente y a singular.

Con los textos limpios se hizo un wordcloud, que se muestra a continuación

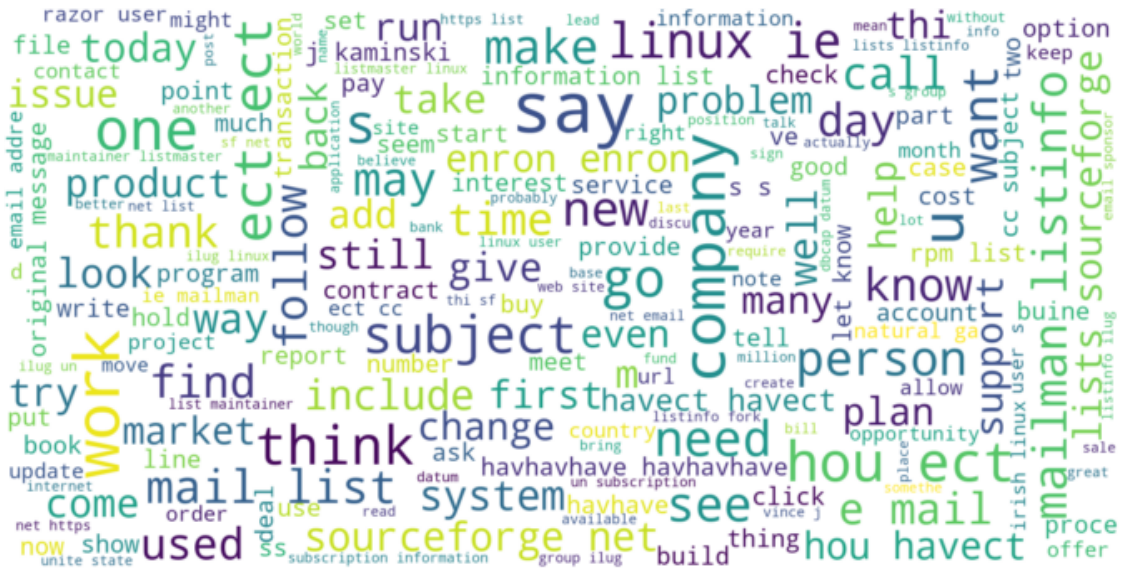

Figura 3: Wordcloud del conjunto de datos

Modelos

Para ambos modelos se utilizó el multinomial de Naive Bayes como modelo predictor, al cual se le entrenó con el 70% de los datos y se puso a prueba con el 30%.

Bag of N grams

En este modelo se trabajó para n = 1 y 2, con una sensibilidad de (0.05 - 0.8), del cual se obtuvieron los siguientes resultados:

lso	and	another	around	...	work	world	would	would like	write	wrote	www	year	ymy	you
0	2	0	0	...	0	0	0	0	0	0	0	0	1	2
0	0	0	0	...	0	0	0	0	0	0	7	0	1	1
0	0	0	0	...	0	0	0	0	0	0	6	0	1	1
0	2	0	0	...	0	0	0	0	0	0	0	0	4	1
0	0	0	0	...	0	0	0	0	0	0	5	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	...	0	0	1	0	0	0	0	0	0	0

Figura 4: Resultado de aplicar el modelo BoW con n = 1 y 2

Luego de esto se entrenó con estos datos el modelo multinomial y se obtuvieron los siguientes resultados:

```
TP:  2406
FP:  258
FN:  337
TN:  1493

array([[2406,  258],
       [ 337, 1493]], dtype=int64)
```

Figura 5: Matriz de confusión con modelo BoW con n = 1 y 2

```
Accuracy: 0.867601246105919
Precision: 0.8526556253569388
Recall:  0.8158469945355191
F1:  0.8338452946104439
```

Figura 6: Métricas con modelo BoW con n = 1 y 2

TF-IDF

En este modelo se trabajó con una sensibilidad de (0.05 - |), del cual se obtuvieron los siguientes resultados:

	account	act	add	addres	all	allow	also	and	another	around	...	without	work
0	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.18	0.0	0.0	...	0.0	0.00
1	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
2	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
3	0.05	0.0	0.00	0.00	0.05	0.06	0.00	0.11	0.0	0.0	...	0.0	0.00
4	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
14974	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
14975	0.00	0.0	0.03	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
14976	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.04
14977	0.00	0.0	0.33	0.00	0.00	0.00	0.00	0.00	0.0	0.0	...	0.0	0.00
14978	0.00	0.0	0.00	0.19	0.00	0.00	0.15	0.00	0.0	0.0	...	0.0	0.00

Figura 7: Resultado de aplicar el modelo TF-IDF

Luego de esto se entrenó con estos datos el modelo multinomial y se obtuvieron los siguientes resultados:

```
TP:  2469
FP:  195
FN:  303
TN:  1527

array([[2469,  195],
       [ 303, 1527]], dtype=int64)
```

Figura 8: Matriz de confusión con modelo TF-IDF

```
Accuracy: 0.8891855807743658
Precision: 0.8867595818815331
Recall:  0.8344262295081967
F1:  0.8597972972972973
```

Figura 9: Métricas con modelo TF-IDF

Discusión

Como se puede observar dentro de las métricas el modelo de TF-IDF cuenta con mejores resultados que el Bag of N grams. Con este conjunto de datos queda mejor el TF-IDF, este al ser una relación matemática con la importancia de la palabra dentro de la oración; el modelo verifica de manera más sencilla que palabras son más impactantes en un correo de spam y cuáles no.

Como se puede observar en la ***figura 8***, se cuenta con los TP que son los valores que cuentan con que no son spam (0), mientras que el TN son todos los correos de spam (1). Mientras que los FP son todos los que son spam pero fueron clasificados como no spam. Por último el FN, son todos aquellos que no son de spam, pero fueron clasificados como que sí lo son. Queda claro que el modelo no es perfecto y explica por que hay veces en las que algún correo que no es de spam cae en la bandeja de spam y hay que irlo a buscar y por eso las personas indican que si no se ve en la bandeja de entrada, se debe de buscar en la bandeja de spam.

Luego en la **figura 9**, se observan las métricas en donde el modelo TF-IDF cuenta con una exactitud de ~89%, una precisión de ~89% un recall de ~83% y un F1 ~86%. En donde se puede verificar que el modelo fue exacto al momento de clasificar los textos, además de que los que clasificó como no spam son en verdad no spam, el cual se puede observar con la métrica de precisión. Pero de todos los que no clasificó bien (FP y FN) se relacionan con los que no son spam (VP) y se observa en la métrica de recall para observar cual es el impacto de clasificar mal. Mientras que F1 es una métrica que mide cómo se relacionan entre el recall y la precisión. Con base a esto queda claro que, el modelo se desempeñó bien al tener unas métricas no menores al 80% el cual es el mínimo valor aceptable.