

Naive Bayes Classifier

Spam E-mail Detection

25 de agosto de 2025

Introducción

Manejo de datos

Desarrollo del clasificador

Aplicación

Nuevos mensajes

Conclusiones

Situación problema

El problema consiste en clasificar automáticamente emails como spam o no spam y poder alivianar la bandeja de entrada de usuarios de correo electrónico, usando clasificadores ingenuos de Bayes.



Figura: Correo spam físico

A screenshot of an email inbox interface. The top bar shows a search field and a status filter set to "Any Status". The inbox list has columns for Subject, Sender, and Date. The list contains several emails, with the one from "kenneth draper" selected. The selected email's subject is "I find more savings online".

Subject	Sender	Date
check this out man...	Nelda Romano	Thursday 14:59:37
Help me!	Oswaldo MANNING	Thursday 12:47:59
Have Arthritis pain? There is help for you.	Ona	Thursday 03:45:36
down on her, and	Reginald Shullis	Wednesday 06:02:05
natural enlargement	diane george	Tuesday 16:37:15
We Subject	fakien schmitt	Monday 10:28:09
only Youngest have Shocking sexuality other	Kristie Sapp	Monday 01:07:32
Reduces stress	frankie kim	06:02:2005 16:27
PERSONAL	warv2005	06:02:2005 04:56
We need to render the delight of having the best	Christina Gudrungrt	06:02:2005 02:10
I find more savings online	kenneth draper	05:02:2005 22:30
Faster cheaper meals	Lila White	05:02:2005 16:37
Breaking News	Osw H. Edwardsd	05:02:2005 14:40
We have your wanted meals at low prices only.	Iacien hyatt	04:02:2005 06:59
100% zum einladen...1679438	Isai Rios	03:02:2005 03:34
Enjoy your wanted meals.	tracy alani	03:02:2005 02:28
Confirm Your Washington Mutual Online Banking	Washington Mutual On.	02:02:2005 22:03
and PINNACLE SYSTEM, MACROMEDIA, SYMANTEC, PC GAMES...	Valerie Beem	02:02:2005 19:11
Free!	Cecilia Fuller	02:02:2005 06:57
You can save more thru ordering meals on our site.	mrl senick	02:02:2005 01:21
The most insane action	Katrina Souza	31:01:2005 08:19
You don't have to be fat. Now!	Kirstin	28:01:2005 03:22

Figura: Correo spam electrónico

Preprocesamiento

El conjunto de datos trabajado puede encontrarse en el siguiente hipervínculo: **Mail Dataset** Para el procesamiento se realizó lo siguiente

- ▶ Limpieza del dataset
 1. Eliminación de columnas innecesarias.
 2. Renombrado de columnas.
 3. Codificación de etiquetas.
- ▶ Preprocesamiento de texto (NLP)
 1. Estandarizar a minúsculas los strings `.lower()`.
 2. Remoción de caracteres especiales.
 3. Remoción de stopwords.
 4. Stemming y lemmatization.
- ▶ División de datos para entrenamiento y evaluación.

Funciones implementadas

Se construyeron las siguientes funciones como bloques en el diseño del clasificador.

- ▶ `flatten_list (list[list] -> list)`:
Esa función toma una lista de palabras y las aplanar en una sola lista concatenando todas las palabras de todos los emails.
- ▶ `bag_of_words (pd.DataFrame -> dict)`:
Retorna la tabla de frecuencias para las palabras de un conjunto de mails.
- ▶ `probability_words (pd.DataFrame -> dict)`: Retorna la tabla de frecuencias relativas para las palabras de un conjunto de mails.
- ▶ `clasiffy_email (list -> int)`: retorna entre 1, 0 para clasificar un email en spam

Construcción de Bolsa de Palabras

Algorithm 1 Construcción de Bolsa de Palabras

Require: corpus: conjunto de emails tokenizados

Ensure: diccionario $\{palabra : frecuencia\}$

- 1: $lista_palabras \leftarrow \emptyset$
 - 2: **for** cada email en corpus **do**
 - 3: **for** cada palabra en email **do**
 - 4: agregar palabra a $lista_palabras$
 - 5: agrega 1 al contador de la palabra
 - 6: **end for**
 - 7: **end for**
 - 8: calcular frecuencia de cada palabra única en $lista_palabras$
 - 9: **return** diccionario con palabras y sus frecuencias
-

Cálculo de Probabilidades de Palabras

Algorithm 2 Probabilidades de Palabras

Require: df: conjunto de emails (spam o no-spam) procesados

Ensure: diccionario $\{palabra : probabilidad\}$

- 1: $lista_palabras \leftarrow$ aplanar todas las palabras de df
 - 2: $tamaño_total \leftarrow$ longitud de $lista_palabras$
 - 3: $frecuencias \leftarrow$ bag_of_words(df)
 - 4: $probabilidades \leftarrow \emptyset$
 - 5: **for** cada $(palabra, frecuencia)$ en $frecuencias$ **do**
 - 6: $probabilidades[palabra] \leftarrow \frac{frecuencia}{tamaño_total}$
 - 7: **end for**
 - 8: **return** $probabilidades$
-

Clasificador de correos

Algorithm 3 Clasificar correo

Require: email: lista de palabras procesadas del correo

Require: p_{spam} : probabilidad a priori de spam

Require: p_{no_spam} : probabilidad a priori de no-spam

Ensure: 1 si el email es spam, 0 en caso contrario

1: $spam_prob \leftarrow \log(p_{spam})$

2: $not_spam_prob \leftarrow \log(p_{no_spam})$

3: **for** cada palabra en email **do**

4: $spam_word_prob \leftarrow probability_spam_words.get(palabra, 10^{-6})$

5: $not_spam_word_prob \leftarrow probability_ham_words.get(palabra, 10^{-6})$

6: $spam_prob \leftarrow spam_prob + \log(spam_word_prob)$

7: $not_spam_prob \leftarrow not_spam_prob + \log(not_spam_word_prob)$

8: **end for**

9: **return** 1 if $spam_prob > not_spam_prob$ else 0

Funcionamiento del clasificador

Para implementar la solución computacionalmente, se utilizó el suavizamiento de Laplace, que asigna una probabilidad muy pequeña ($1e^{-6}$), pero no cero, a las palabras no encontradas en la bolsa de palabras. De esta manera, se evitan ceros cuando hay una palabra no vista antes.

Además, trabajamos en el espacio logarítmico para evitar el underflow numérico y convertir productos en sumas.

$$\log\left(P(c) \prod_{i=1}^n P(w_i | c)\right) = \log P(c) + \sum_{i=1}^n \log P(w_i | c)$$

Funcionamiento del clasificador

Por último, usamos un treshold (o umbral) de 0.5 para disminuir la cantidad de falsos positivos.

Evaluación del modelo

Al construir el modelo clasificador Naive Bayes y entrenarlo se empleó para la evaluación del desempeño la matriz de confusión y las 4 métricas derivadas de la matriz, **accuracy**, **precision**, **recall**, **F1 score**

Aplicación

Estas fueron las métricas finales del modelo:

	Predicted	
	positives	negatives
actual positives	131	13
actual negatives	5	965

Metrics	
Accuracy	0.983842
Precision	0.963235
Recall	0.909722
F1 Score	0.935714

Ejemplos

idx	email (tokens)	spam	prediction
3	[u, dun, say, earli, hor, u, c, already, say]	0	0
4	[nah, think, goe, usf, live, around, though]	0	0
7	[per, request, mell, mell, oru, minnaminingt...]	0	0
8	[winner, valu, network, custom, select, receiv...]	1	1
10	[gon, na, home, soon, want, talk, stuff, anymo...]	0	0
5551	[wen, get, spiritu, deep, great]	0	0
5555	[yeh, indian, nice, tho, kane, bit, shud, go, ...]	0	0

Generación de mensajes

A partir del modelo anterior, se generaron mensajes de spam y no spam con palabras aleatorias que son probables de encontrar en ambas bags of words. El resultado fue el siguiente:

Clase	Mensaje (tokens)
No spam	sleep sm salam wait gim ah love hand librari thought usc hungri lot watch sian hundr what cheap mornin school
Spam	ur 1 xxx today com frnd 18p winner activ 69911 4 stop life get mind 2 capit 2optout xma phone

Cuadro: Ejemplos de mensajes clasificados por el modelo.

Conclusiones

- ▶ El clasificador Naive Bayes resultó ser altamente efectivo para identificar correos spam con alta **precisión** y **recall**.
- ▶ El preprocesamiento de los emails, desde la limpieza, tokenización, el uso de stopwords y stemming fueron clave para la implementación y éxito del modelo.
- ▶ Para poder generar un email coherente se tendría que implementar procesos extra para aprender la relación de cada palabra con las demás, es decir, un bloque de atención, haciéndolo más parecido a un modelo de arquitectura transformer. En este caso únicamente se generaron aleatoriamente palabras de la clase spam o no spam sin ninguna relación una con la otra.
- ▶ Se experimentó con técnicas para mejorar el modelo como el suavizamiento de laplace y probabilidad logaritmica.