Bayesiano ingenuo para el análisis de opinión

Como vamos a comprobar ahora, el teorema bayesiano ingenuo nos va a permitir hacer el análisis de la opción de los internautas en función de sus publicaciones en las redes sociales.

Al analizar los distintos mensajes que han dejado los usuarios en las redes sociales o en sitios de Internet sobre un producto o un tema en particular (deportes, política u otros), podemos determinar si están bastante contentos, descontentos, felices, tristes, de acuerdo o en desacuerdo, y actuar en consecuencia.

¿Cómo? Analizando las palabras de los distintos mensajes y clasificando estos mensajes en distintas categorías: de acuerdo, en desacuerdo, contento o descontento. Esta clasificación permite, por ejemplo, avisar directamente a los responsables de comunicación de un producto en caso de descontento ; al hacerlo así, se puede responder al usuario con mucha rapidez y evitar una mala imagen de la marca.

Tomemos como caso de estudio una serie de conversaciones sobre los gatos extraída de las redes sociales. A partir de la aparición de las palabras en una frase se puede identificar la opinión de la persona que ha publicado el mensaje: si le gustan o no los gatos.

El conjunto de las observaciones que vamos a utilizar en este capítulo se basará en textos en inglés. El motivo es que, desgraciadamente, no es tan fácil encontrar conjuntos de datos en español y, por su composición, resulta mucho más fácil trabajar con el idioma inglés.

| **Mensaje** | **Opinión** |
| --- | --- |
| I love cats | Positiva |
| I hate cats | Negativa |
| Cats are beautiful | Positiva |
| Cats are awful | Negativa |
| Cats are intelligent | Positiva |

Ahora imaginamos un comentario nuevo: «I love awful cats». ¿A la persona que ha publicado este comentario le gustan o no le gustan los gatos? Es la pregunta difícil a la que tendrá que responder nuestro algoritmo de clasificación de textos.

1. Fase 1: Normalización de los datos

La normalización del texto consistirá en eliminar los acentos, los caracteres especiales y ponerlo en minúsculas:

| **Mensaje** | **Opinión** |
| --- | --- |
| i love cats | Positiva |
| i hate cats | Negativa |
| cats are beautiful | Positiva |
| cats are awful | Negativa |
| cats are intelligent | Positiva |

2. Fase 2: Eliminación de las stop words

Las *stop words* son las palabras de un idioma más utilizadas y que no aportan valor añadido al sentido de la frase. Por eso conviene eliminarlas. Veremos que en Python hay funciones que pueden hacer esto por nosotros.

Este es el resultado de la eliminación de las *stop words* (simplemente eran las palabras «I» y «are»:

| **Mensaje** | **Opinión** |
| --- | --- |
| love cats | Positiva |
| hate cats | Negativa |
| cats beautiful | Positiva |
| cats awful | Negativa |
| cats intelligent | Positiva |

La lista de las stop words del idioma inglés está disponible en esta dirección: <https://gist.github.com/sebleier/554280>

3. Fase 3: El stemming

La fase de *stemming* consiste en eliminar los sufijos y los prefijos de las palabras. En nuestro caso, podemos eliminar las «s» de las palabras «cats», el sufijo «ful» de la palabra «beautiful» y «ent» en la palabra «intelligent».

| **Mensaje** | **Opinión** |
| --- | --- |
| love cat | Positiva |
| hate cat | Negativa |
| cat beauti | Positiva |
| cat awful | Negativa |
| cat intellig | Positiva |

4. Fase 4: La lematización

La lematización tiene en cuenta el análisis morfológico de las palabras para obtener solo nombres.

En nuestro caso, los datos no se modifican porque solo tenemos nombres.

5. Fase 5: Determinar la cantidad de incidencias de cada palabra

Esta es una fase relativamente sencilla porque consiste en hacer una tabla, donde las columnas son la lista completa de las palabras del conjunto de las frases y en las filas aparecen las distintas frases.

Luego, solo hay que tomar una palabra dada y contar la cantidad de veces que aparece en cada una de las frases.

| **Frase** | **love** | **cat** | **hate** | **beauti** | **awful** | **intellig** | **Opinión** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| love cat | 1 | 1 | 0 | 0 | 0 | 0 | Positiva |
| hate cat | 0 | 1 | 1 | 0 | 0 | 0 | Negativa |
| cat beauti | 0 | 1 | 0 | 1 | 0 | 0 | Positiva |
| cat awful | 0 | 1 | 0 | 0 | 1 | 0 | Negativa |
| cat intellig | 0 | 1 | 0 | 0 | 0 | 1 | Positiva |

6. Fase 6: Determinar las probabilidades para la opinión positiva

Ahora vamos a calcular las distintas probabilidades correspondientes a las opiniones positivas. Para ello solo vamos a considerar las frases positivas:

| **Frase** | **love** | **cat** | **hate** | **beauti** | **awful** | **intellig** | **Opinión** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| love cat | 1 | 1 | 0 | 0 | 0 | 0 | Positiva |
| cat beauti | 0 | 1 | 0 | 1 | 0 | 0 | Positiva |
| cat intellig | 0 | 1 | 0 | 0 | 0 | 1 | Positiva |

* Probabilidad de opiniones positivas, que designaremos P(+) = 3 frases de 5, es decir 3/5, es decir = 0,6.
* Cantidad de palabras: 6 (love, cat, hate, beauti, awful,intellig).
* Cantidad total de apariciones: 6 (suma de los 1 de la tabla).

Para calcular las probabilidades, vamos a utilizar la fórmula siguiente:



Aquí se explica el hecho de que, si la probabilidad de la palabra encantador es igual a 0 para una opinión positiva, la frase «este gato es encantador» también tendrá una probabilidad 0. Pero, como no queremos obtener una probabilidad igual a 0, para evitarlo llevaremos a cabo una operación de suavizado llamada suavizado de Laplace (Laplace smoothing), que consiste en añadir 1 al dividendo.

Aquí podemos ver un ejemplo que permite calcular la probabilidad de que la opinión sea positiva si se utiliza la palabra «love»:

* Cantidad de apariciones de la palabra «love» usada en una frase que tiene una opinión positiva = 1.
* Cantidad total de palabras = 6.
* Cantidad total de apariciones = 6.



Ahora hacemos el mismo proceso para las otras palabras:

|  |  |  |
| --- | --- | --- |
| P(love|positiva) | Probabilidad de uso de la palabra «love» para opiniones positivas | (1+1)/(6 + 6) = 0,1666 |
| P(cat|positiva) | Probabilidad de uso de la palabra «cat» para opiniones positivas | (3+1)/(6+6) = 0,3333  (La palabra «cat» se utiliza 3 veces en las frases que expresan una opinión positiva) |
| P(hate|positiva) | Probabilidad de uso de la palabra «hate» para opiniones positivas | (0+1)/(6+6) = 0,0833  (La palabra «hate» no se utiliza en las frases que expresan una opinión positiva) |
| P(beauti| positiva) | Probabilidad de uso de la palabra «beauti» para opiniones positivas | (1+1)/(6 + 6) = 0,1666 |
| P(awful| positiva) | Probabilidad de uso de la palabra «awful» para opiniones positivas | (0+1)/(6+6) = 0,0833 |
| P(intellig| positiva) | Probabilidad de uso de la palabra «intellig» para opiniones positivas | (1+1)/(6 + 6) = 0,1666 |

7. Fase 7: Determinar las probabilidades para la opinión positiva

Ahora extraemos las frases que expresan una opinión negativa:

| **Frase** | **love** | **cat** | **hate** | **beauti** | **awful** | **intellig** | **Opinión** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| hate cat | 0 | 1 | 1 | 0 | 0 | 0 | Negativa |
| cat awful | 0 | 1 | 0 | 0 | 1 | 0 | Negativa |

* Dos frases de cinco expresan una opinión negativa. Por lo tanto, la probabilidad de que una frase tenga una opinión negativa es de 2/5, es decir, P(opinión negativa) = 0,4.
* Cantidad de palabras: 6 (love, cat, hate, beauti, awful, intellig).
* Cantidad total de apariciones: 4 (suma de los 1 de la tabla).

A continuación calculamos las probabilidades de las distintas palabras en función de una frase que expresa una opinión negativa:

|  |  |  |
| --- | --- | --- |
| P(love|negativa) | Probabilidad de uso de la palabra «love» para opiniones negativas | (0+1)/(4+6) = 0,1 |
| P(cat| negativa) | Probabilidad de uso de la palabra «cat» para opiniones negativas | (2+1)/(4+6) = 0,3  (La palabra cat se usa 2 veces en las frases que expresan una opinión negativa) |
| P(hate| negativa) | Probabilidad de uso de la palabra «hate» para opiniones negativas | (1+1)/(4 + 6) = 0,2 |
| P(beauti| negativa) | Probabilidad de uso de la palabra «beauti» para opiniones negativas | (0+1)/(4+6) = 0,1 |
| P(awful| negativa) | Probabilidad de uso de la palabra «awful» para opiniones negativas | (1+1)/(4 + 6) = 0,2 |
| P(intellig| negativa) | Probabilidad de uso de la palabra «intellig» para opiniones negativas | (0+1)/(4+6) = 0,1 |

8. Fase 8: Determinar la opinión de una frase nueva

Una vez determinado el conjunto de las probabilidades, podemos pasar a la fase de clasificación de nuestra frase nueva:

"I love awful cats"

No podemos olvidar aplicarle todas las fases de normalización, de eliminación de *stop words*, aplicación de *stemming* y la lematización antes de determinar su opinión.

Este es el resultado de las operaciones:

love awful cat

Ahora podemos pasar a la fase de clasificación de nuestra frase realizando las siguientes acciones:

* Calcular la probabilidad de que la frase exprese una opinión positiva.
* Calcular la probabilidad de que la frase exprese una opinión negativa.
* Tomar la probabilidad cuando tiene un valor más intenso.

P(Opinión Positiva) = P(+) \* P(love|+) \* P(awful|+) \* P(cat|+)

P(Opinión Positiva) = 0,6 \* 0,1666 \* 0,0833 \* 0,3333

P(Opinión Positiva) = 0,0027

P(Opinión negativa) = P(-) \* P(love|-) \* P(awful|-) \* P(cat|-)

P(Opinión negativa) = 0,4\*0,1\*0,2\*0,3

P(Opinión negativa) = 0,0024

Como 0,0027 es mayor que 0,0024, la frase tiene una opinión positiva.

Al proceder de esta manera acabamos de utilizar el teorema bayesiano ingenuo aplicado a la clasificación. Para los más curiosos, esta es la fórmula aplicada:

Imagen que contiene Icono

Descripción generada automáticamente

Para las dos opiniones V que nombraremos de manera individual Vj (Vj opinión negativa y Vj opinión positiva).

La probabilidad de un tipo de opinión P(Vj) corresponde a la probabilidad Vj multiplicada por la suma de los productos de las probabilidades de cada palabra que pertenece a Vj.

P(Opinión positiva) = P(+) \* P(love|+) \* P(awful|+) \* P(cat|+)

P(Opinión negativa) = P(-) \* P(love|-) \* P(awful|-) \* P(cat|-)

La opinión de la frase se determina tomando la probabilidad máxima (argMax) de las probabilidades Vj calculadas.

argMax(0,0027; 0,0024) = 0,0027