Social Media, Marketing y Business Intelligence

Repercusión de Marcas en las Redes Sociales mediante la utilización de Técnicas de Análisis de Sentimientos.

Fecha de Creación: 13/05/2012

info@stratebi.com

@stratebi

www.stratebi.com - www.todobi.com – 91.788.34.10

1. Introducción

Comenzamos aquí un documento en el que una de sus finalidades será el ser capaces de medir **la repercusión de una marca o producto dentro de las redes sociales**, en los casos de estudio que veremos a continuación centramos nuestros esfuerzos en los términos "pentaho" y "firefox", pero no debemos quedarnos aquí ya que para cualquier corporación llámese como se llame y en particular para el **departamento de Marketing** de una empresa toda la información que a través de este trabajo nos va a revelar puede resultar muy valiosa.

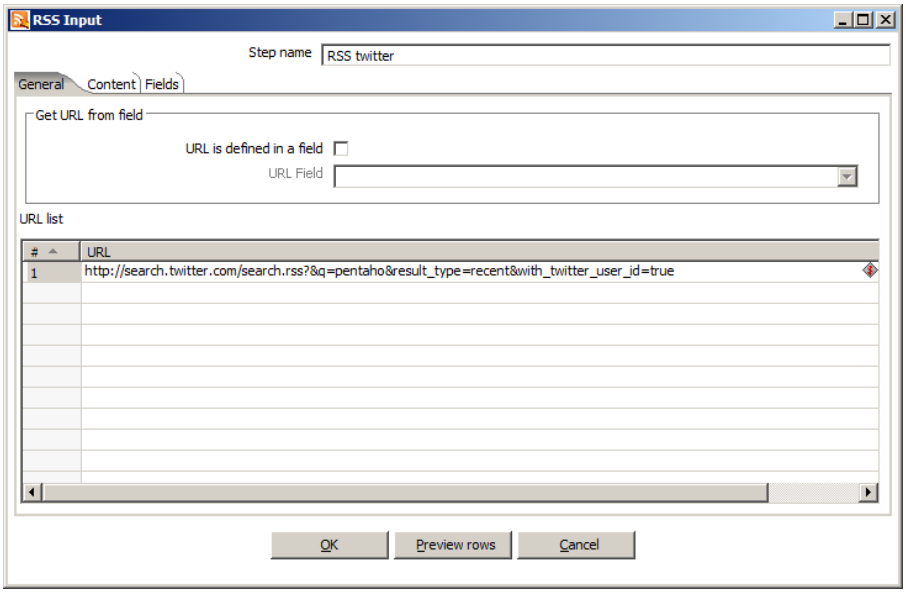
Con el procedimiento que hemos seguido podríamos analizar tanto marcas comerciales (ej: Coca Cola, Oracle) como por ejemplo para analizar el impacto del lanzamiento de un nuevo videojuego en el mercado o un estreno de cine. También resultaría adecuado su uso en el ámbito político para observar las opiniones que se tienen sobre los candidatos así como la monitorización de fuentes de información para detectar hostilidades

La técnica que hemos empleado para tal fin es la de Análisis de Sentimientos (conocida también como Minería de Opiniones, Clasificación de Sentimientos o Computación Afectiva), esta es una técnica que dados unos datos de origen con un formato de texto, en los que aparecen opiniones o sentimientos sobre distintas entidades u objetos, permite extraer las opiniones de los mismos y clasificarlas. Es decir, podríamos decir que se trata de un tratamiento computacional de las opiniones, sentimientos y fenómenos subjetivos en los textos.

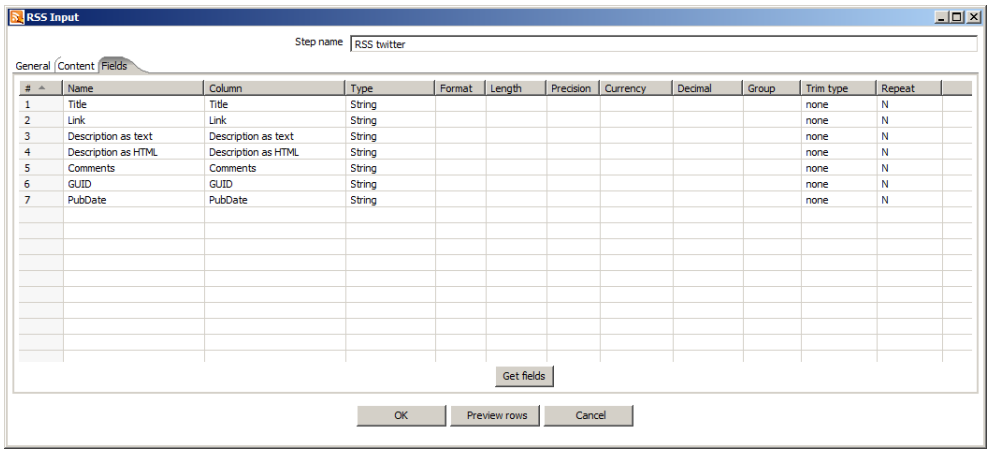
Esta técnica utiliza el lenguaje natural, ya que es el que utiliza el usuario, y tratar computacionalmente este lenguaje conlleva ciertos problemas como la ambigüedad de las palabras, ya que dependen fuertemente del contexto. Los retos a los que se enfrenta son la extracción de las características sobre las que se está opinando y la clasificación de dichas características.

2. Comienzo del estudio

Para el desarrollo de este estudio hemos realizado la búsqueda de la palabra Pentaho en la red de microblogging twitter. Hemos escogido este término puesto que recientemente la compañía dedicada al Business Intelligence, ha lanzado al mercado una nueva versión de su software. En una primera captura vemos como podemos obtener la información fuente que vamos a utilizar en nuestro estudio. Se trata de un paso de entrada RSS (versión de PDI utilizada 4.2.1 estable) en el que especificamos que los tweets que estamos buscando deben contener la palabra pentaho.

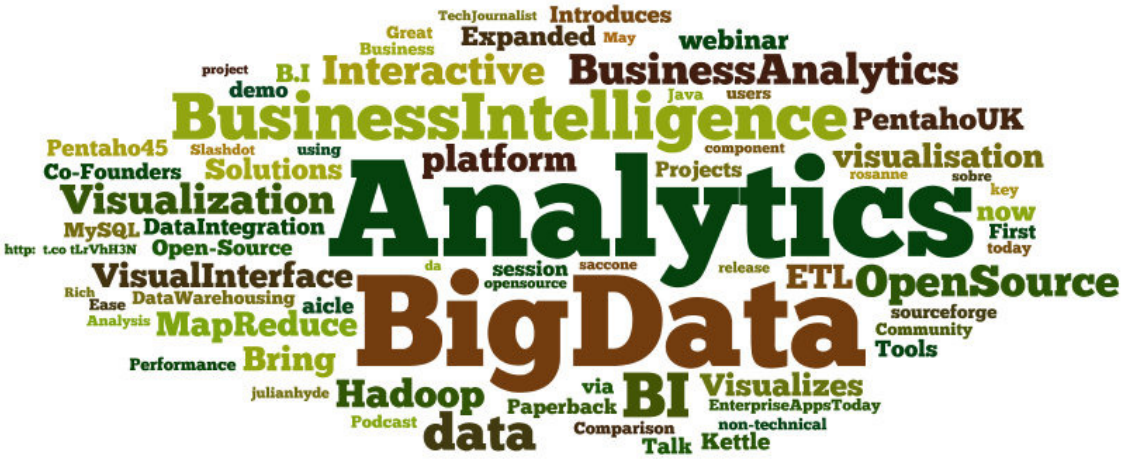


Si nos dirigimos a la pestaña Fields obtendremos de manera automática los campos que cada entidad que nos vamos a traer contiene. Es importante destacar que la API de búsqueda de twitter tiene unas limitaciones para evitar una sobrecarga en sus servidores y que solo nos permite obtener información de los últimos 10 días o un máximo de 1500 tweets.



Con todas las entidades procesadas con la herramienta de integración de datos que incluye el propio Pentaho, realizamos una nube de palabras con el texto de los tweets para darnos una ligera idea de que términos tienen relación con la palabra que estamos buscando.

Podríamos ver esto como un resumen que le podríamos dar a un extraterrestre que no tiene ni idea de que es ni que significa el término Pentaho. Pues bien con un simple golpe de vista y analizando las palabras que mas resaltan nos podemos dar cuenta que es un software de Business Intelligence y que además se trata de una herramienta Open Source, comentar que en esta nube de palabras se ha excluido el propio termino pentaho para que no se desvirtuaran los resultados.



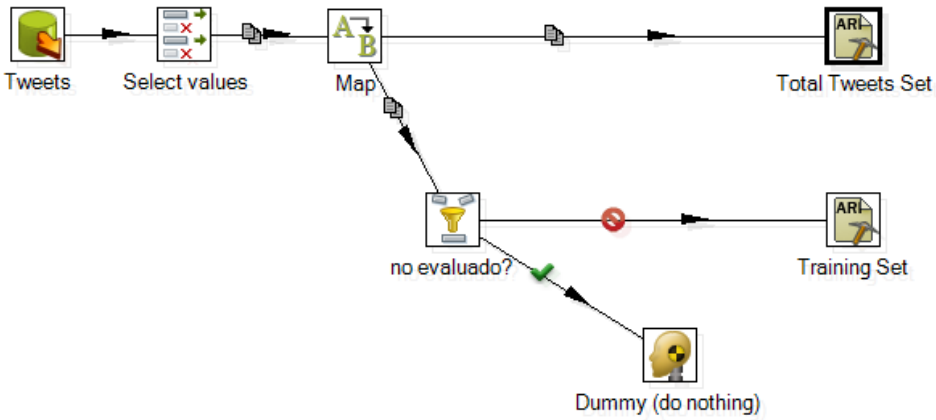
En una segunda pasamos a analizar los hashtags de todo el tráfico de tweets que nos hemos traído anteriormente, por si algún lector a estas alturas no tuviera claro lo que es un hashtag decir que se trata de una cadena de caracteres precedida de un almohadilla (#), de esta manera los usuarios remarcan un tema o una palabra dotándola de una mayor

importancia. Con un hashtag se suele expresar una opinión o sentimiento (ejemplos: #MeGustaPentaho, #BusinessIntelligence, #EleccionesFrancia).

En nuestra muestra vemos como el hasthag mayoritario es pentaho, y son muy importantes también los términos bigdata y pentaho45 lo que nos indica que la nueva versión 4.5 ha sido muy popular en los tweets recientes, además vemos que en sus últimas versiones pentaho incorpora conectividad y posibilidad de trabajar con las tecnologías emergentes de bigdata. Resulta sorprendente que con un simple análisis automático podamos obtener tanta información, pero aun queda lo mejor.

En la tercera parte de este estudio hemos tratado de realizar minería de datos con la muestra anterior pero nos ha resultado imposible puesto que partíamos de solo 750 tweets de los cuales tras una minuciosa evaluación humana 250 resultan expresar una opinión positiva mientras que solo 20 muestran rechazo o disgusto con pentaho, por lo que se desestima este caso de estudio puesto que no se disponen de entidades negativas suficientes para que los algoritmos de minería aprendan con garantías.

Sondeando un poco la red social nos decantamos por la búsqueda Firefox como tema alternativo para realizar Data Mining. Con esta búsqueda ya tenemos un total de aproximadamente 1500 tweets (total de tweets que la API de twitter permite obtener), de estas entidades se han evaluado manualmente un total de 419 tweets, siendo los restantes analizado mediante scripts SQL de búsqueda de palabras o smileys que expresen sentimientos (Ejemplos: like, love, nice,cool, great, hate, crash, crazy, fail, error, ☺ , ☹). Con todos los tweets evaluados ya sea de manera manual o automática nos toca generar dos ficheros ARFF que nos servirán de fuente para la herramienta de DM Weka, se genera un fichero con el total de los datos (1500 tweets) y otro con el subconjunto de tweets que hemos analizado de forma manual, a este subconjunto lo pasamos a llamar conjunto de entrenamiento pues en el futuro es lo que va a servir para entrenar a los diferentes algoritmos utilizados.



K medias

El primer algoritmo que aplicaremos es el denominado K medias.
Se trata de un algoritmo voraz para partir los datos en k clústers, este procedimiento utiliza la distancia euclídea de cada uno de los puntos al centro de cada clúster para su posterior agrupamiento. Es muy útil como primera aproximación por ser uno de los más veloces y eficientes. Debido a la naturaleza de nuestros datos vamos a escoger como valor de K el 3 puesto que nos es fácil ver que existirán 3 conjuntos bien diferenciados de tweets (positivos, negativos y neutros).

Para ello abrimos el WEKA explorer y le pasamos como fuente el ARFF con los datos del conjunto de entrenamiento (419 tweets), después nos vamos a la pestaña Cluster y elegimos el algoritmo SimpleKMeans al que manualmente le ponemos el valor 3 en el campo numClusters. El siguiente paso es pasarle el fichero ARFF con el total de los tweets como conjunto para realizar los test. Una vez tengamos configurado lo anterior le damos a Start, y lo que WEKA va a realizar es lo siguiente: Tomar los datos de entrenamiento como datos maestros para realizar un aprendizaje y en segunda instancia aplicar esos conocimientos que ha adquirido al conjunto total de los datos, realizando una predicción del conjunto final en el que se encontraría cada uno de los tweets. El proceso de aprendizaje, denominado modelo puede guardarse en un archivo con extensión .model para su reutilización dentro de Weka o dentro de Pentaho Data Integration a través del paso Weka Scoring.

Datos de la conjunto de entrenamiento (evaluación humana):

Valoración de Tweet	Número de Tweets
Positivo	228
Negativo	98
Neutro	93
TOTAL	419

En los resultados que se muestran a continuación podemos ver que el clúster número 0 es el destinado a los tweets con sentimientos negativos, dado que los atributos negativos mayor valor medio en el. El clúster 1 por otra parte es el dedicado a las entidades en las que los usuarios de twitter han proporcionado una opinión positiva puesto que vemos que en él se

alojan los 228 tweets que manualmente señalamos como positivos. En el tercer conjunto etiquetado como clúster 2 es en el que se guardan los objetos neutros.

```
Cluster centroids:
                                Cluster#
Attribute          Full Data          0          1          2
                     (419)       (98)      (228)       (93)
============================================================
positive              0.6062     0.2551     1.0044          0
                     +/-0.6563  +/-0.5974  +/-0.5268       +/-0

positive_smiley       0.0453          0     0.0833          0
                     +/-0.2083       +/-0   +/-0.277       +/-0

negativo              0.3031     1.1224     0.0746          0
                     +/-0.5961  +/-0.6464  +/-0.3094       +/-0

negative_smiley       0.0143     0.0612          0          0
                     +/-0.1189   +/-0.241       +/-0       +/-0

h_eval                    Good        Bad       Good    Neutral
  Bad                98 ( 23%)   98 (100%)   0 (  0%)   0 (  0%)
  Good              228 ( 54%)    0 (  0%) 228 (100%)   0 (  0%)
  Neutral            93 ( 22%)    0 (  0%)   0 (  0%)  93 (100%)
  Not_Classified      0 (  0%)    0 (  0%)   0 (  0%)   0 (  0%)

=== Evaluation on test set ===
Clustered Instances

0       188 ( 12%)
1       395 ( 25%)
2       967 ( 62%)
```

El algoritmo de K medias tras realizar el aprendizaje nos clasifica al total de los tweets de la siguiente manera: Cluster 0 : 188 tweets, Cluster 1: 395 tweets y Clúster 2.

Datos del conjunto total (evaluación automática WEKA algoritmo 3 medias):

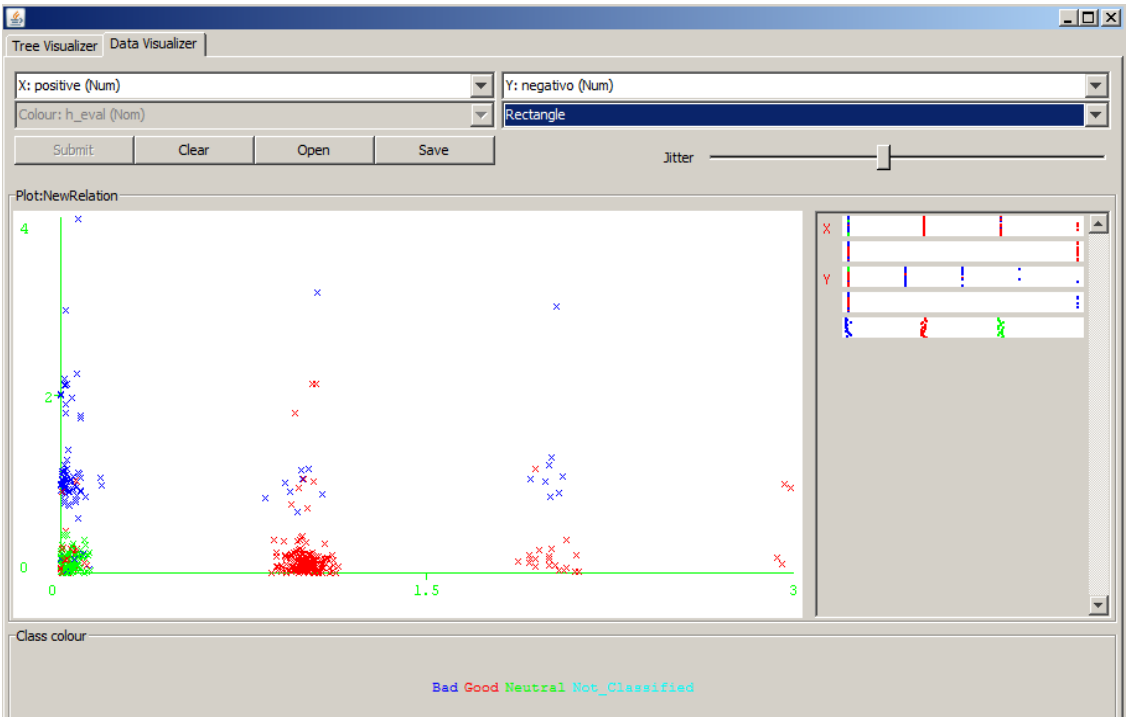
Valoración de Tweet	Número de Tweets
Positivo	395
Negativo	188
Neutro	967
TOTAL	1550

Estos resultados son un resumen de la ejecución en un último paso generaremos una hoja de cálculo con los tweets y su predicción asociada.

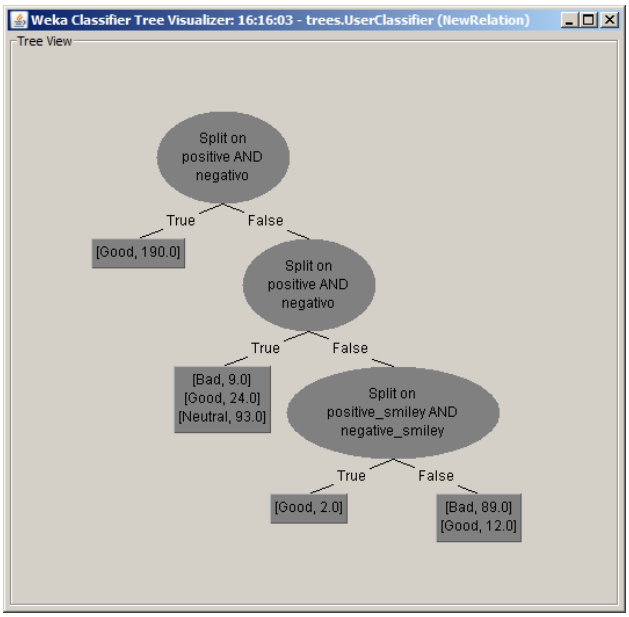
Árbol creado por el usuario

El segundo método que vamos a emplear es el de generar nosotros manualmente un árbol que nos va servir como modelo. Para su elaboración vamos a la pestaña superior de Weka y

elegimos User Classifier como algoritmo, le damos a comenzar y nos mostrara una ventana con una caja con el total de los datos sobre la que debemos de pulsar para ir a un formato de visualización de eje de coordenadas XY, a través de este eje podemos contrastar el valor de las diferentes variables y formar grupos de valores con un mismo valor.



Una vez clasificados los distintos valores de la manera más homogénea posible cambiamos a la pestaña para visualizar el árbol y veremos algo como lo siguiente:



Al cerrar la ventana de edición del árbol nos ejecutará el procedimiento que hemos creado con el árbol sobre el conjunto de datos total, recordar especificarle el conjunto total como Supplied Test Set. Los resultados obtenidos son los siguientes

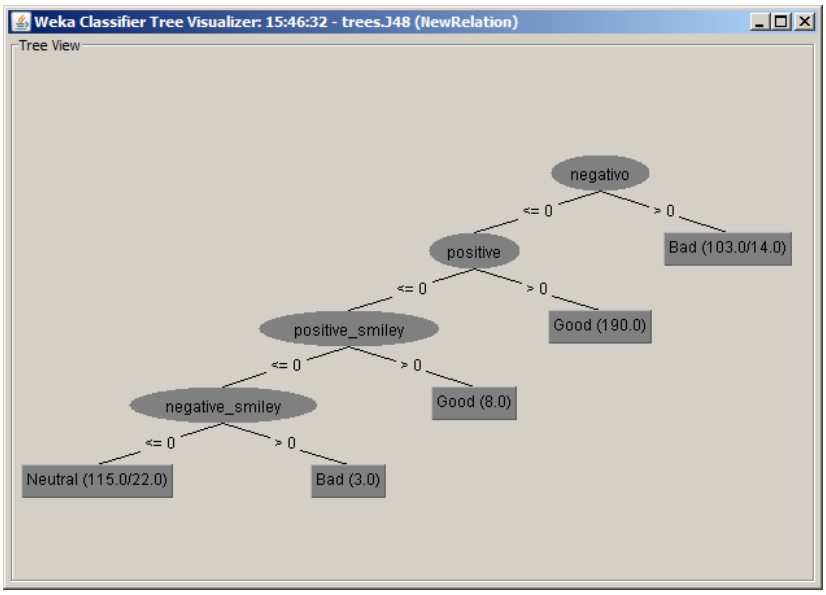
```
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  89   0   9   0 |   a = Bad
  12 192  24   0 |   b = Good
   0   0  93   0 |   c = Neutral
 100 138 893   0 |   d = Not_Classified
```

En ella podemos ver como de los 1131 tweets que no estaban en el conjunto de entrenamiento, es decir los que estaban clasificados no clasificados los ha distribuido de la siguiente forma: 100 como negativos, 138 positivos y 893 como neutros. Estos resultados son un resumen de la ejecución global, en un último paso generaremos una hoja de cálculo con los tweets y su predicción asociada.

Árbol J 48

El algoritmo J48 de WEKA es una implementación del algoritmo C 4.5, uno de los algoritmos de minería de datos más utilizado. Se trata de un refinamiento del modelo generado con OneR (regla mayoritaria sobre un solo atributo). El parámetro más importante que deberemos tener en cuenta es el factor de confianza para la poda, confidence level , que influye en el tamaño y capacidad de predicción del árbol construido. Para cada operación de poda, define la probabilidad de error que se permite a la hipótesis de que el empeoramiento debido a esta operación es significativo. A probabilidad menor, se exigirá que la diferencia en los errores de predicción antes y después de podar sea más significativa para no podar. El valor por defecto es del 25%. Según baje este valor, se permiten más operaciones de poda. El árbol que este algoritmo genera automáticamente es el siguiente (recordar pasar como conjunto de test el total de los datos).



Vemos en ahora el pseudocódigo del árbol.

negativo <= 0

| positive <= 0

| | positive_smiley <= 0

| | | negative_smiley <= 0: Neutral (115.0/22.0)

| | | negative_smiley > 0: Bad (3.0)

| | positive_smiley > 0: Good (8.0)

| positive > 0: Good (190.0)

negativo > 0: Bad (103.0/14.0)

Los resultados que obtenemos con este algoritmo son los siguientes

```
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  92   0   6   0 |   a = Bad
  14 198  16   0 |   b = Good
   0   0  93   0 |   c = Neutral
 110 147 874   0 |   d = Not_Classified
```

En ella podemos ver como de los 1131 tweets que no estaban en el conjunto de entrenamiento, es decir los que estaban clasificados como Not_Classified los ha distribuido de la siguiente forma: 110 como negativos, 147 positivos y 874 como neutros.

Con la ejecución de los dos algoritmos que utilizan una forma de árbol podemos realizar una primera comparativa directa, simplemente viendo cómo operan sobre el conjunto de datos de entrenamiento. Los resultados que arroja esta comparativa son que el algoritmo J48 es ligeramente superior a el correspondiente al árbol que nosotros hemos creado puesto que tiene un mayor porcentaje de aciertos.

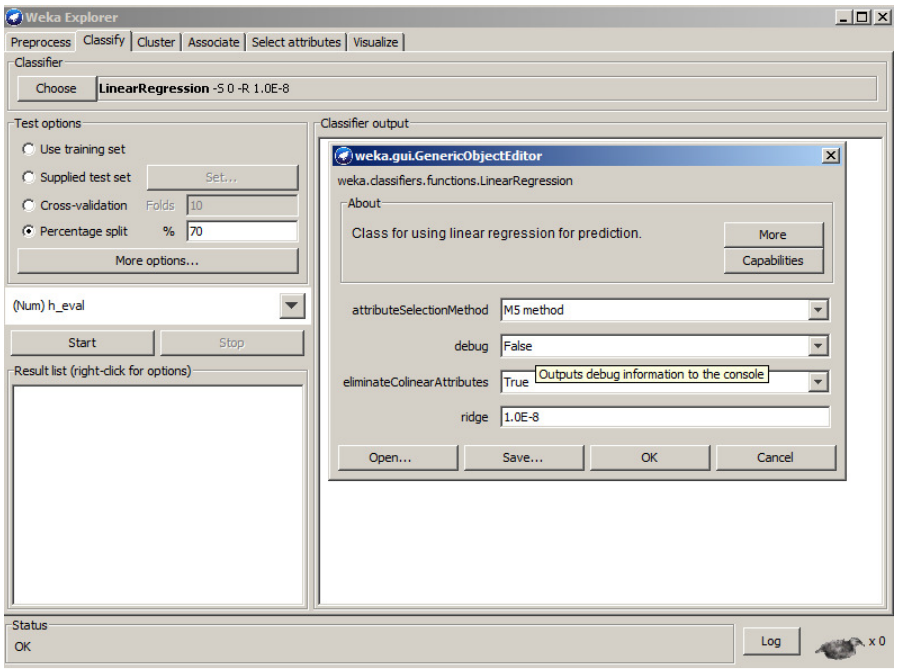
User Tree						
Bad	**Good**	**Neutral**	**Not_classified**	**<--Classified as**		
89	0	9	0	**Bad**	Bien clasificados:	374
12	192	24	0	**Good**	mal clasificados	45
0	0	93	0	**Neutral**	% ✘	10,74%
100	138	893	0	**Not_Classified**	% ✓	89,26%

J48						
Bad	**Good**	**Neutral**	**Not_classified**	**<--Classified as**		
92	0	6	0	**Bad**	Bien clasificados:	383
14	198	16	0	**Good**	Mal clasificados	36
0	0	93	0	**Neutral**	% ✘	8,59%
110	147	874	0	**Not_Classified**	% ✓	91,41%

Algoritmo de Regresión Lineal

El cuarto algoritmo que vamos a utilizar consiste en aplicar una regresión lineal a nuestro caso de estudio (método con el que se intenta modelar a través de una recta, la relación entre una variable dependiente Y, las variables independientes X_i y una constante aleatoria k, ecuación $Y= a_1X_1+ a_2X_2 + a_3X_2+\ldots +a_nX_n+ k$).

El primer paso que debemos de hacer previamente de realizar la regresión propiamente dicha es el de normalizar las variables, desde la ventana de preprocesamiento escogemos los atributos y escogemos les aplicamos el filtro normalizar, con esto vamos a lograr que nuestros valores originales se distribuyan en el intervalo (0,1) de esta manera evitamos que atributos con valores elevados metan ruido a nuestra recta. En el segundo paso debemos seleccionar en la pestaña Clasificar el algoritmo Linear Regression, en la captura de pantalla bajo estas líneas podemos observar que el propio algoritmo tiene marcada la opción de eliminar los atributos colineales (vectores colineales son aquellos paralelos en el plano) para que se reproduzca la mejor manera posible la recta de predicción.

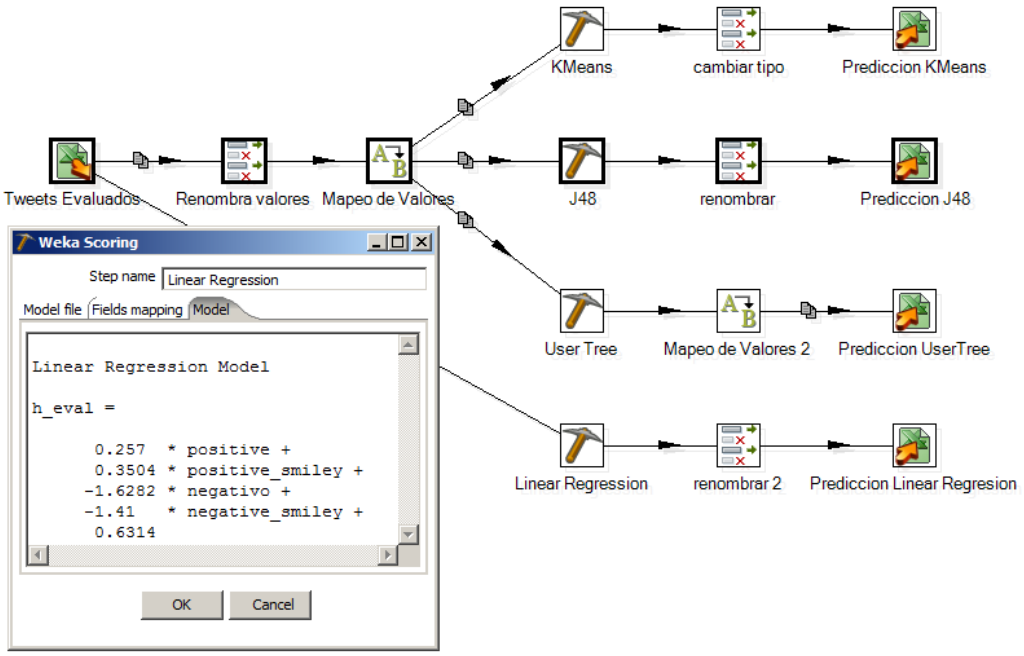


La recta que este algoritmo nos genera es la siguiente:

h_eval = 0.257 * positive + 0.3504 * positive_smiley -1.6282 * negative -1.41 * negative_smiley+ 0.6314

Nos guardamos de la misma manera que con los anteriores el modelo de este algoritmo. Con los 4 modelos guardados nos disponemos a realizar la última parte de este estudio, a lo largo de las diferentes ejecuciones de los distintos procedimientos hemos visto de una forma u otra como mediante el aprendizaje WEKA realizaba predicciones sobre los tweets que no estaban en el conjunto de entrenamiento, pero estos procedimientos nos devolvían datos estadísticos y resúmenes del numero de tweets que supuestamente serían de cada tipo.

En esta última transformación de Kettle lo que vamos a recuperar son los tweets propiamente dichos junto con su predicción asociada. El formato de las predicciones varía en función del algoritmo empleado: en K medias será un número de clúster, en los de tipo árbol el pronóstico devuelto será un término que describe el tweet. Por último en la regresión lineal lo que se nos devuelve es un valor entero perteneciente siguiente conjunto {-6,-4,-3,-2,-1, 0, 1} que indica el sentimiento del tweet, los valores más altos (1 y 0) del conjunto indican las entidades claramente positivas mientras que los valores iguales o menores a -1 indican aquellos tweets que expresan un sentimiento negativo.



En esta tabla vemos como los diferentes procedimientos han realizado sus predicciones sobre los tweets, destacar su alta fiabilidad que podemos observar con un simple golpe de vista.

El único error que vemos es el perteneciente al algoritmo que nosotros mismos hemos realizado puesto que esta categorizando como neutro un tweet que claramente es negativo. También debemos fijarnos en como la regresión lineal asigna el valor -3 al primer tweet expresando que contiene más de un término negativo lo que denota un mayor grado de insatisfacción.

Tweet	Clúster	User Tree	J48	Linear Regression
Is it just I or is Firefox the browser that hangs and crashes the most? :-(	0 (Bad)	Bad	Bad	-3
FastestFox - Browse Faster :: Add-ons for Firefox https://t.co/Khn21Wzo	1 (Good)	Good	Good	1
Icant believe that firefox has a better ftp client than android os has.	1 (Good)	Good	Good	1
FastestFox - Browse Faster :: Add-ons for Firefox https://t.co/uWQ0Mx5F	1 (Good)	Good	Good	1
@gregsidelnikov nice but its dont work on firefox ...	1 (Good)	Bad	Bad	-1
@danmasso Closed out Firefox and started over. :(	0 (Bad)	Neutral	Bad	-1
Honestly Firefox is annoying me now. #memorybloat	0 (Bad)	Bad	Bad	-1
#noscript is the most annoying #firefox addon	0 (Bad)	Bad	Bad	-1
RT @Three_Ninjas: Firefox crashes once a day.	0 (Bad)	Bad	Bad	-1
@misterjaydee thanks for the Firefox <3 ^WR	1 (Good)	Good	Good	1
I hate the new Mozilla Firefox.	0 (Bad)	Bad	Bad	-1
Oh.. #Firefox is cool too #JustSaying	1 (Good)	Good	Good	1
Firefox is getting slower day by day	0 (Bad)	Bad	Bad	-1
Long story short, thanks @firefox	1 (Good)	Good	Good	1

@tfaiso firefox and good show	1 (Good)	Good	Good	1
i hate firefox.	0 (Bad)	Bad	Bad	-1

En este trabajo se ha podido demostrar la gran utilidad que tiene la minería de datos al aplicarla a un caso real. Hemos experimentado lo sencillo que es mediante WEKA el análisis y estudio estos datos, y su posterior interpretación. Hemos decidido varias de las posibilidades que nos ofrece esta herramienta para hacer un estudio más completo.

El preprocesado, la clasificación, el agrupamiento, la asociación y la visualización previos de los datos de entrada nos han permitido obtener, con más facilidad, mejores resultados. ¿Te ha parecido atractivo nuestro ejemplo? Si estás interesado en realizarlo con tus propios datos en tu organización, no lo dudes y contacta con nosotros (info@stratebi.com) 91.788.34.10