



TRABAJO FIN DE MASTER ANALISIS DE SENTIMIENTO DE TWITTER

Máster Data Science
Curso 2016/2017

Autor:
Javier González Méndez

Tutor:
Felipe Ortega

AGRADECIMIENTOS

Agradecer a Ana la paciencia y el apoyo que me ha prestado para realizar el máster.

A Jorge y Emma por permitirme que les robase su tiempo.

A Felipe por sus enseñanzas y su tutorización de este trabajo.

A todos los profesores del máster, con especial hincapié en Alberto, Isaac, Jesús y Carlos.

A Javi que tanto me ha enseñado y me ha ayudado a acabar el máster.

Por último, agradecer a todos los compañeros de clase por hacer que las clases fueran tan divertidas e instructivas a la vez.

INDICE

Contenido

1. INTRODUCCION	4
2. OBJETIVOS Y DIAGRAMA DEL PROCESO	5
3. TECNOLOGIAS USADAS	8
3.1 TWITTER.....	8
3.2 TWITTER API	9
3.3 PYTHON.....	11
3.3.1 GetOldTweets-python	11
3.3.2 Nltk.....	12
3.3.3 Sklearn	13
3.4 R	13
3.5 AFINN -- Finn Årup Nielsen.....	15
3.6 Algoritmos utilizados.....	15
3.6.1 Regresión Lineal.....	16
3.6.2 Regresión Logística	16
3.6.3 SVM o Máquinas de vectores de soporte	17
3.6.4 Naive Bayes.....	17
4. METODOLOGIA DE TRABAJO Y RESULTADOS	18
4.1 La descarga de datos	18
4.2 Primer tratamiento.....	19
4.3 Análisis exploratorio de datos	21
4.4 Modelización Estadística	29
5. Evaluación de modelos predictivos con datos reales.....	37
5.1 Los Tweets neutros.....	39
6. CONCLUSIONES.....	42
7. Bibliografía.....	44

1. INTRODUCCION

VERTI Seguros es una compañía de venta directa de seguros especializada en el segmento de Automóviles y Hogar, que opera básicamente a través de Internet y por el canal telefónico. La empresa, cuyo color de referencia es el naranja, inició su actividad en el mercado español el 10 de enero de 2011 ofreciendo productos innovadores con precios competitivos.

Con este lanzamiento, MAPFRE se adentró en el segmento de venta directa de seguros siguiendo un modelo que está funcionando en otros países de Europa, donde las principales aseguradoras que venden sus productos a través de canales presenciales también cuentan con filiales de venta por Internet y teléfono.

Esta aseguradora, que se dirige a un segmento de clientes diferente al tradicional de MAPFRE, compite sobre todo con las compañías on-line y telefónicas que operan en España. Ambas empresas, MAPFRE y VERTI comparten la misma filosofía de atención al cliente y máxima calidad en el servicio.

VERTI comercializa seguros de Automóviles, Motos, Hogar y Mascotas con precios competitivos pero con un alto nivel de servicio y calidad. La venta on-line complementa el modelo de negocio tradicional de MAPFRE, que es líder en el segmento de Automóviles y de Hogar, con unas cuotas de mercado a septiembre de este año del 20 y del 16,3 por ciento, respectivamente.

2. OBJETIVOS Y DIAGRAMA DEL PROCESO

Como hemos explicado, Verti es una compañía de seguro directo enfocada eminentemente al canal online. Por esa razón, mucha parte de la publicidad y los esfuerzos en marketing se centran en estos canales, como puede ser Facebook, Twitter, YouTube,..., canales online en los que se trata de captar y comunicar con los clientes.

En el presente documento, nos vamos a centrar en una de estos canales online de comunicación con los clientes: Twitter. Este canal es usado por la compañía tanto para el lanzamiento de publicidad como un canal de comunicación directo con los clientes o potenciales usuarios. Por eso, vamos a realizar un análisis de sentimientos de los tweets que son emitidos a la cuenta corporativa de la compañía [@vertiseguros](#).

A lo largo del documento vamos a presentar el trabajo, el cual se ha basado en las siguientes partes

- Descarga, limpieza y tratamiento de la información
- Análisis exploratorio de los datos
- Creación de los ficheros de entrenamiento para los modelos estadísticos a través del uso del fichero AFINN-111
- Entrenamiento de los diferentes modelos estadísticos (hasta 7 distintos) con varias metodologías:
 - Uso del fichero total frente a la validación cruzada
 - Uso de la totalidad del texto frente a la eliminación de las stop words
 - Uso de palabras simples frente a bi-gramas (tuplas de palabras consecutivas)
- Vamos a analizar estos modelos desde 4 diferentes parámetros obtenidos a través de los modelos:
 - Accuracy
 - Precision
 - Recall
 - F-measure

Cada cual de estos parámetros nos da una información distinta y muy valiosa de cada modelo

- Para cerrar, intentaremos extrapolar los resultados obtenidos para clasificar los tweets de manera online y realizar acciones concretas con ellos

Como vamos a poder comprobar a lo largo del documento, no ha sido un trabajo fácil. Hemos tenido muchos problemas a la hora de recuperar la información por parte de twitter; hemos tenido que depurar mucho la información obtenida; hemos tenido que cambiar las escalas para calcular la polaridad de los tweets;... Pero al final, vamos a mostrar cómo se distribuyen en el tiempo dichos tweets con su polaridad calculada, diferenciados por una tipología que hemos tenido que considerar para clarificar los datos. También mostraremos cuales han sido las palabras más comentadas en los tweets, para poder sacar así conclusiones más oportunas sobre los comentarios de la gente.

A continuación, hemos estado trabajando con varios modelos estadísticos (Naive Bayes, SVM, Regresión Lineal) para ver cuál de ellos tiene mejor funcionamiento para detectar, a futuro, un tratamiento de los tweets emitidos por los clientes y detectar cuales si o si deben ser contestados por el CM de la cuenta. Hemos usado diferentes técnicas para afrontar el problema como es el uso del texto puro y el texto sin stop words para ver si hay mejoras y luego hemos usado los bi-gramas, para comprobar si el uso de tuplas de palabras consecutivas mejoran los resultados.

Una vez que tengamos un modelo ganador, recuperaremos información real y muy reciente sobre la cuenta para pasarle el modelo que consideremos como ganador y analizaremos los resultados.

Terminaremos el documento con las conclusiones obtenidas en el análisis así como futuros pasos a seguir o trabajos pendientes para ampliar dicho estudio.

Este proceso se puede ver en siguiente diagrama:

Obtención de datos de Twitter

Tecnologías: Python, Twitter

Objetivos: recuperar los datos iniciales para el trabajo

Limpieza y depuración de datos

Tecnologías: Python, Afinn

Objetivos: una vez obtenidos los datos, los tratamos y los depuramos para dejarlos óptimos

Análisis exploratorio de datos

Tecnologías: Python, R

Objetivos: realizamos un análisis descriptivo y exploratorio de los datos para seguir depurando y creamos los ficheros de entrenamiento

Modelización estadística

Tecnologías: Python (NLTK, Sklearn)

Objetivos: entrenamiento y validación de modelos estadísticos. Decisión de tomar el mejor o el que más nos interesa

Presentación de resultados

Tecnologías: Python

Objetivos: aplicación del modelo ganador a datos actualizados. Conclusiones y pasos futuros.

Todos los programas utilizados para la realización de este estudio se pueden encontrar en mi página personal de github:

https://github.com/javiollo/TFM_JaviGlez

3. TECNOLOGIAS USADAS

3.1 TWITTER

Twitter se creó en 2006 como una red social en base a contenido con forma de SMS. Desde entonces ha crecido rápidamente hasta convertirse en la red social de microblogging que es hoy en día. Millones de usuarios acceden cada día para compartir experiencias y opiniones convirtiéndose así en una herramienta ideal para la realización de encuestas y sondeos. No obstante para ello se hace necesario aplicaciones como la que aquí se desarrolla que compacte y de formato a la información haciéndola útil y clara.

Twitter se basa en una red de usuarios los cuales pueden leer y escribir contenido en unidades de hasta 140 caracteres. Cada una de estas unidades se denomina tweet. El conjunto de tweets que ha publicado un usuario aparecen en su perfil por orden cronológico inverso. Sobre un tweet pueden realizarse diversas acciones, a saber:

- **Favorito:** Un tweet es marcado como favorito por otro usuario, esto no tiene ningún efecto más que como contador de usuarios a los que les ha gustado el tweet.
- **Retweet:** Por otro lado, un usuario puede retuitear un tweet, lo cual añadirá ese tweet a su propia línea de publicaciones indicando siempre que se trata de una publicación de otro usuario (del que se retuiteó).
- **Respuestas:** Un tweet puede generarse como respuesta a otro tweet, lo que puede llegar a generar conversaciones o debates.

Cualquiera de estos tres factores puede emplearse como indicativo de la relevancia de un tweet.

Un tweet que ha sido retuiteado por muchos usuarios aparecerá en los perfiles de más personas (y por lo tanto en la página principal o feed), con lo que será leído más veces y tendrá un potencial mayor de llegar a un número elevado de usuarios. Del mismo modo, un tweet muy marcado como favorito suele indicar que el tweet ha gustado a mucha gente, generalmente por concordancia del usuario con lo expresado en el tweet.

Los usuarios se relacionan mediante un sistema de suscripciones, en el que un usuario puede suscribirse al contenido que otro usuario publica, convirtiéndose así en su seguidor.

El conjunto de usuarios a los que sigue un usuario concreto se denominan amigos.

La página principal de Twitter de un usuario (que no su perfil) contiene una mezcla de los tweets y retweets recientes (por orden cronológico inverso) publicados por todos sus amigos.

Un usuario que tenga un número elevado de seguidores logrará que su contenido llegue directamente a muchos más usuarios.

Por lo tanto los tweets y retweets realizados por personas con muchos seguidores aparecerán en la página principal de un conjunto elevado de usuarios, logrando que esos tweets sean leídos por un número grande de usuarios, haciendo que estos tweets tengan un potencial mayor de convertirse en populares.

Por otro lado los tweets pueden contener, además de texto, una serie de “entidades” que se listan a continuación:

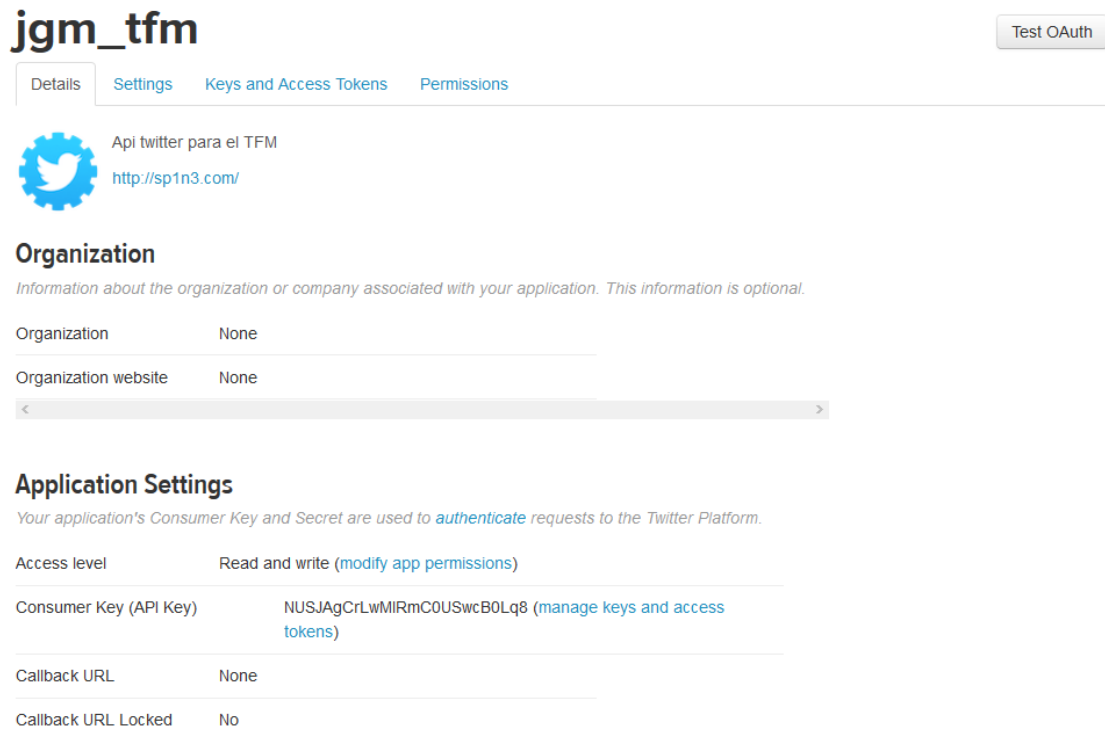
- **Menciones:** Una mención representa una citación a un usuario concreto en un tweet. Se representan mediante el símbolo “@” seguido del nombre del usuario que se quiere citar. Cuando un tweet contiene una mención el usuario citado recibe una notificación informándole de la existencia de dicho tweet.
- **Enlaces:** Un tweet puede contener también URLs o enlaces a cualquier contenido de la red, como imágenes o páginas web.
- **Hashtags:** Los hashtags son una especie de etiquetas que se emplean para indicar o incluso complementar el contenido de un tweet. Se emplean generalmente para agrupar tweets por temáticas, de modo que si se visita la página de un hashtag concreto, podrán verse todos los tweets que se hayan publicado y lo contengan.

3.2 TWITTER API

Twitter pone a disposición de los desarrolladores una gran variedad de documentación, herramientas y APIs. Se puede encontrar desde widgets¹⁰ para incorporar en páginas web, o información sobre Twitter Cards¹¹, hasta soporte sobre Apps para móvil, la API de Streaming¹², o la API REST.

La API REST o API proporciona acceso de lectura sobre los datos públicos en Twitter, y por lo tanto no requiere disponer datos de ingreso de las cuentas a las que se pretende acceder, por lo que se pueden extraer datos como los tweets de un usuario, información de perfiles usuario y datos de los seguidores entre otras opciones de una forma muy sencilla y para cualquier cuenta.

Para configurar la API, primero hay que registrarse en Twitter y crear una App en Twitter Apps¹³, luego generar las credenciales o el OAuth con el cual la API identifica las peticiones y así crear las respuestas JSON.



The screenshot shows the Twitter Developer Portal interface for an application named 'jgm_tfm'. At the top right is a 'Test OAuth' button. Below the application name are tabs for 'Details', 'Settings', 'Keys and Access Tokens', and 'Permissions'. The 'Settings' tab is active, showing the application's icon (a blue Twitter bird) and the name 'Api twitter para el TFM' with the URL 'http://sp1n3.com/'. Below this is the 'Organization' section, which is currently set to 'None'. The 'Application Settings' section shows the 'Access level' as 'Read and write (modify app permissions)', the 'Consumer Key (API Key)' as 'NUSJAgCrLwMIRmC0USwcB0Lq8 (manage keys and access tokens)', the 'Callback URL' as 'None', and the 'Callback URL Locked' status as 'No'.

(figura 3.1)

Algunas de las peticiones o métodos más comunes que se pueden realizar mediante la API son GET followers/ids, GET friends/ids, GET followers/list, GET users/show, GET users/lookup o GET statuses/user_timeline, aunque hay muchos más. En el uso de las diferentes peticiones existentes hay que tener en cuenta algunos factores que limitan las posibilidades de estas. Los dos factores más importantes son:

1. Para aquellos métodos que devuelven una línea de tiempo de tweets, hay que tener en cuenta que la paginación de la información puede generar problemas debido a la naturaleza de tiempo real de los propios tweets. Por lo que lo más común será utilizar el parámetro max_id a partir del cual se cargaran los tweets, evitando así duplicidades en caso de haber actualizaciones entre petición y petición.
2. Otro factor a tener en cuenta es el límite en la tasa de peticiones. La tasa límite de la API se define en intervalos (o ventanas) de 15 minutos en las que según el método utilizado se pueden realizar un número determinado de peticiones u otro. La tabla siguiente muestra los límites para algunos de los métodos.

3.3 PYTHON

La motivación por la que usar el lenguaje de programación Python es debida a la evolución que este ha experimentado en los últimos años, y aunque el lenguaje es de propósito general, este es ampliamente utilizado en ciencias e ingeniería y dispone de una gran cantidad de bibliotecas para multitud de aplicaciones relacionadas con los campos de Machine Learning y Natural Language Processing, al igual que una librería con la que usar la API de Twitter. Además dispone de una gran cantidad de documentación online.

Así pues, se trata de un lenguaje de programación relativamente moderno, creado en el año 1991 por el Holandés Guido Rossum al conseguir unir las mejores ideas de otros lenguajes de una forma sencilla, intuitiva y fácil de aprender. Como se ha comentado, en los últimos años ha conseguido aumentar su popularidad gracias al auge de las aplicaciones web, hasta tal punto que se ha convertido en uno de los lenguajes de programación oficiales de Google.

Python es un lenguaje dinámico, es decir, no necesita declarar las variables previamente a la asignación del valor, si no que cada variable toma directamente el tipo del valor que se le esté asignando. Así pues la mayoría de las asignaciones resuelven el tipo en tiempo de ejecución y no en tiempo de compilación.

Por tanto, se puede añadir que también es un lenguaje interpretado, lo que quiere decir que el código no llega a traducirse a algo que el sistema operativo entienda, si no que este es interpretado por una máquina virtual que es capaz de entender y ejecutar el código, consiguiendo así tiempos de trabajo mucho más rápidos al no tener que realizar el ciclo de compilación, ejecución y depuración.

Python es un lenguaje de programación orientado a objetos, de hecho en Python prácticamente todo son objetos. Los objetos son entidades o instancias de una clase. Las clases están formadas por atributos, las variables que definen el objeto, y por métodos, las funciones que operan con estas variables.

Python dispone de muchísimos entornos de desarrollo diferentes, algunos libres, otros comerciales y otros enfocados al contexto científico. En este sentido, uno de los entornos más completos y el utilizado en el desarrollo del proyecto es Anaconda.

3.3.1 GetOldTweets-python

La API oficial de Twitter tiene la limitación de tiempo en la que no se puede obtener tweets más viejos de una semana. Algunas herramientas proporcionan acceso a los tweets más antiguos, pero en la mayoría de ellos tienes que gastar algo de dinero antes. Básicamente,

cuando se introduce en la página de Twitter, un scroll se inicia y si se desplaza hacia abajo se comienza a obtener más y más tweets, todas las llamadas a un proveedor JSON. Una vez que se cargan los datos, los podemos recuperar a través de los paquetes Python Lxml y Pyquery.

El mayor problema de esta herramienta es que no podemos descargar la totalidad de información que ofrece la API de oficial de Twitter en formato json. Los únicos datos que podemos obtener son los siguientes:

- Id (str)
- Permalink (str)
- Nombre de usuario (str)
- Texto (str)
- Fecha (fecha)
- Retweets (int)
- Favoritos (int)
- Menciones (str)
- Hashtags (str)
- Geo (str)

Con esta herramienta podemos recuperar todos los tweets de un usuario, podemos filtrar por fechas, podemos realizar una consulta de algún término en el texto del tweet,...

Podemos encontrar esta herramienta en el siguiente enlace:

<https://github.com/Jefferson-Henrique/GetOldTweets-python>

3.3.2 Nltk

Según la Wikipedia el procesamiento del lenguaje natural (PLN) es: Es una subdisciplina de la Inteligencia Artificial. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales.

El análisis automático de sentimiento es un paso más en el intento de traducir las emociones humanas a datos. Pero la espontaneidad y la inmediatez de la opinión en medios sociales hacen que estos sentimientos sean más auténticos y preserven su contenido emocional.

Para conseguir procesar el lenguaje natural existen muchas herramientas, una de ellas es NLTK que posee una colección de paquetes y objetos Python muy adaptados para tareas de PLN.

El Natural Language Toolkit (NLTK) es una plataforma usada para construir programas para análisis de texto. La plataforma fue liberada originalmente para análisis de texto. La plataforma fue liberada originalmente por Steven Bird y Edward Loper en conjunto con un curso de lingüística computacional en la Universidad de Pennsylvania en 2001. Hay un libro de acompañamiento para la plataforma llamado Procesamiento de Lenguaje Natural con Python.

3.3.3 Sklearn

Scikit-learn es la principal librería que existe para trabajar con Machine Learning, incluye la implementación de un gran número de algoritmos de aprendizaje. La podemos utilizar para clasificaciones, extracción de características, regresiones, agrupaciones, reducción de dimensiones, selección de modelos, o pre procesamiento. Posee una API que es consistente en todos los modelos y se integra muy bien con el resto de los paquetes científicos que ofrece Python. Esta librería también nos facilita las tareas de evaluación, diagnóstico y validaciones cruzadas ya que nos proporciona varios métodos de fábrica para poder realizar estas tareas en forma muy simple.

3.4 R

R es un lenguaje y entorno de programación, creado en 1993 por Ross Ihaka y Robert Gentleman del Departamento de Estadística de la Universidad de Auckland, cuya característica principal es que forma un entorno de análisis estadístico para la manipulación de datos, su cálculo y la creación de gráficos. En su aspecto R puede considerarse como otra implementación del lenguaje de programación S, con la particularidad de que es un software GNU, General Public License (conjunto de programas desarrollados por la Free Software Foundation), es decir, de uso libre.

La página principal del proyecto “R-project” es <http://www.r-project.org>, en ella podremos conseguir gratuitamente el programa en su última versión, o cualquiera de las anteriores (para el caso de utilizar paquetes no implementados para las últimas versiones), además de manuales, librerías o package y demás elementos que forman la gran familia que es R.

Hay que tener en cuenta que R es un proyecto vivo y sus capacidades no coinciden totalmente con las de S. A menudo el lenguaje S es el vínculo escogido por

investigadores que utilizan la metodología estadística, y R les proporciona una ruta de código abierto para la participación en esa actividad, los usuarios pueden contribuir al proyecto implementando cualquiera de el-las, creando modificaciones de datos y funciones, librerías (packages),... Ningún otro programa en la actualidad reúne las condiciones de madurez, cantidad de recursos y manejabilidad que posee R, además de ser el que en los últimos años ha tenido una mayor implantación en la comunidad científica.

Entre otras características dispone de:

- Almacenamiento y manipulación de datos.
- Operadores para cálculo sobre variables indexadas (Arrays), en particular matrices.
- Herramientas para análisis de datos.
- Posibilidades gráficas para análisis de datos.

El término entorno lo caracteriza como un sistema completamente diseñado y coherente de análisis de datos. Como tal es muy dinámico y las diferentes versiones no siempre son totalmente compatibles con las anteriores. En la introducción a R no se hace mención explícitamente a la palabra estadística, sin embargo mayoritariamente se utiliza R como un sistema estadístico, aunque la descripción más precisa sería la de un entorno en el que se han implementado muchas técnicas estadísticas. Algunas están incluidas en el entorno base de R y otras se acompañan en forma de bibliotecas (packages).

Una diferencia fundamental de la filosofía de R, y también de la de S, con el resto del software estadístico es el uso del “objetos” (variables, variables indexadas, cadenas de caracteres, funciones, etc.) como entidad básica. Cualquier expresión evaluada por R se realiza en una serie de pasos, con unos resultados intermedios que se van almacenando en objetos, para ser observados o analizados posteriormente, de tal manera que se puede hacer un análisis sin necesidad de mostrar su resultado inmediatamente produciendo unas salidas mínimas.

Cada objeto pertenece a una clase, de forma que las funciones pueden tener comportamientos diferentes en función de la clase a la que pertenece su objeto argumento. Por ejemplo no se comporta igual una función cuando su argumento es un vector que cuando es un fichero de datos u otra función.

R está disponible en varios formatos: en código fuente está escrito principalmente en C (y algunas rutinas en Fortran), esencialmente para máquinas Unix y Linux, o como archivos binarios precompilados para Windows, Linux (Debian, Mandrake, RedHat, SuSe), Macintosh y Alpha Unix.

Junto con R se incluyen ocho bibliotecas o paquetes (llamadas bibliotecas estándar) pero otros muchos están disponibles a través de Internet en (<http://www.r-project.org>). Actualmente se encuentran disponibles 10.000 librerías (packages) desarrollados en R, que cubren multitud de campos desde aplicaciones Bayesianas, financieras, graficación de mapas, wavelets, análisis de datos espaciales, etc. Esto es lo que define R como un entorno vivo, que se actualiza con frecuencia y que está abierto a la mejora continua.

Podemos ver o modificar la lista de bibliotecas disponibles mediante la función `“.libPaths”` y conocer el camino a la biblioteca predeterminada del sistema La variable `“.Library”`. Estas bibliotecas se pueden clasificar en tres grupos: las que forman parte del sistema base y estarán en cualquier instalación, los paquetes recomendados (aunque no forman parte del sistema base se aconseja su instalación) y otros paquetes desarrollados por investigadores de todo el mundo para tareas o métodos de lo más diverso. Destacando áreas nuevas como ciencias de la salud, epidemiología, bioinformática, geoestadística, métodos gráficos, etc.

Una característica del lenguaje R es que permite al usuario combinar en un solo programa diferentes funciones estadísticas para realizar análisis más complejos. Además los usuarios de R tienen a su disponibilidad un gran número de programas escritos para S y disponibles en la red la mayoría de los cuales pueden ser utilizados directamente con R.

3.5 AFINN -- Finn Årup Nielsen

AFINN es una lista de palabras inglesas calificadas para valencia con un entero entre menos cinco (negativo) y más cinco (positivo). Las palabras han sido etiquetadas manualmente por Finn Årup Nielsen en 2009-2011

Hay dos versiones:

- AFINN-111: La versión más reciente con 2.477 palabras y frases.
- AFINN-96: 1.468 palabras y frases únicas en 1.480 líneas.

3.6 Algoritmos utilizados

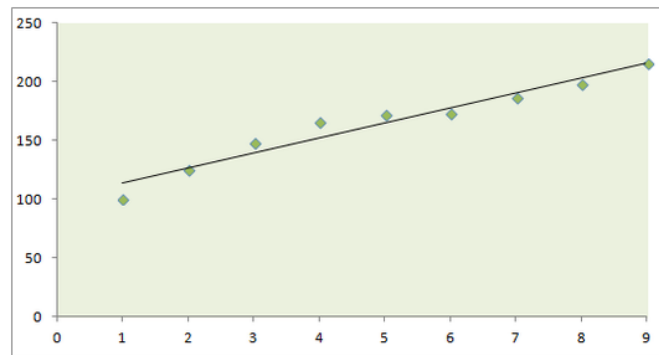
Los algoritmos de Machine Learning que hemos utilizado en este trabajo han sido, fundamentalmente: regresiones lineales, SVM y Naive Bayes, todos ellos son algoritmos supervisados. Esta supervisión se consigue a través de unos datos de entrenamiento los cuales sirven como guía a los algoritmos para llegar a resultados óptimos.

Los vamos a explicar un poco por encima.

3.6.1 Regresión Lineal.

Se utiliza para estimar los valores reales (costo de las viviendas, el número de llamadas, ventas totales, etc.) basados en variables continuas. La idea es tratar de establecer la relación entre las variables independientes y dependientes por medio de ajustar una mejor línea recta con respecto a los puntos. Esta línea de mejor ajuste se conoce como línea de regresión y está representada por la siguiente ecuación lineal:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

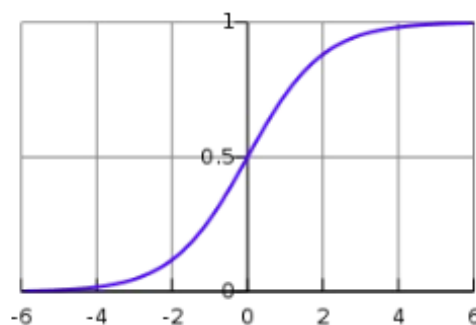


(figura 3.2)

3.6.2 Regresión Logística

Los modelos lineales, también pueden ser utilizados para clasificaciones; es decir, que primero ajustamos el modelo lineal a la probabilidad de que una cierta clase o categoría ocurra y, a luego, utilizamos una función para crear un umbral en el cual especificamos el resultado de una de estas clases o categorías. La función que utiliza este modelo, no es ni más ni menos que la función logística:

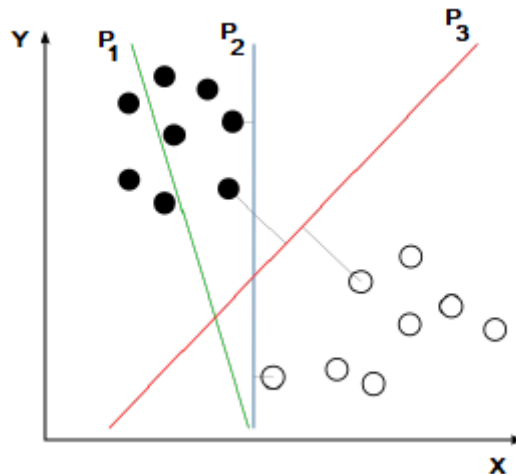
$$f(t) = \frac{e^t}{1 + e^t}$$



(figura 3.3)

3.6.3 SVM o Máquinas de vectores de soporte

Dado un conjunto de ejemplos de entrenamiento, una SVM los representa como puntos en el espacio, separando las clases por la mayor distancia posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, según su proximidad serían asignadas a una u otra clase. Es decir, si un punto nuevo pertenece a una categoría o la otra.



(figura 3.4)

En la figura 3.3 podemos ver como el plano P_1 no separa las clases, mientras que P_2 y P_3 lo hace, con la salvedad de que P_3 lo hace con separación máxima. Esos planos se conocen como vectores soporte.

3.6.4 Naive Bayes

El algoritmo de clasificación Naive Bayes es un clasificador probabilístico. Se basa en modelos de probabilidades que incorporan fuertes suposiciones de independencia.

Las suposiciones de independencia a menos no tienen ningún efecto sobre la realidad. Por lo tanto, se consideran ingenuas (naive en inglés).

Es posible derivar modelos de probabilidades utilizando el teorema de Bayes (atribuido a Thomas Bayes). En función de la naturaleza del modelo de probabilidades, el algoritmo Naive Bayes puede prepararse en un entorno de aprendizaje supervisado.

4. METODOLOGIA DE TRABAJO Y RESULTADOS

4.1 La descarga de datos

La descarga de datos ha sido mucho más difícil de lo que esperaba. El principal problema que he tenido es que al usar la API de Twitter, solamente te devuelve los datos de una determinada cuenta hasta 7 días atrás desde la fecha de consulta, con lo que me ha sido imposible recuperar los datos de la cuenta de la empresa. Me puse en contacto con el CM de la compañía para ver si me los podía recuperar él, con resultados bastante decepcionantes.

Al final, tras buscar en muchos sitios y preguntar a mucha gente, encontré una herramienta en github que me han permitido descargar los datos necesarios para el desarrollo del análisis. Esta herramienta se llama GetOldTweets-python. Desde un principio trabajé con la descarga de todos los tweets que contuvieran solamente la palabra “Verti” y fue un error. No solamente descargué una cantidad ingente de datos (y de manera complicada, debido al gran volumen) sino que la gran mayoría de ellos eran totalmente innecesarios a vista de este análisis. Por ejemplo, se descargaba todos los tweets de cuentas que contuvieran dicha palabra y que no aportaban nada al estudio; se descargaban multitud de tweets del tipo “Vertí una lágrima....” (nótese la tilde en Vertí), los cuales tampoco aportan nada y la hora de limpiar las tildes en los textos se perdían y ensombrecían los datos; al trabajar solamente con los datos que te ofrecía la herramienta, entre los que no se encuentra el lenguaje en el que se ha escrito el tweet, se incluían tweets en otros idiomas, con lo que volvíamos a embarrar los datos de inicio.

Tras hacer muchas pruebas con esta herramienta y viendo las ventajas y desventajas que me ofrecían todas las posibilidades disponibles, opté por realizar una descarga de tweets a través de una consulta que contuvieran una combinación de palabras que pudieran aportar al estudio, como por ejemplo las palabras “verti” y “seguros” a la vez. Con esta consulta, garantizaba descargar los tweets emitidos desde de la cuenta de la que realizamos el análisis, así como los replys de la misma y las menciones que le hacen a la cuenta. A la vez, he intentado también capturar aquellos tweets que hacen referencia a la cuenta aunque no le hagan directamente a través de @vertiseguros. Luego estuve investigando la cuenta en Twitter y vi que perdíamos, por ejemplo, los siguientes tweets, los cuales son, a parte de malsonantes, tweets muy negativos que nos valen como ejemplos:



(figura 4.1)

Viendo más tweets de este tipo, las combinaciones de palabras introducidas han sido: “verti seguro”, “verti spotify”, “verti bien”, “verti mal”, “verti peor”, “verti siniestro”,... Esto ha generado multitud de ficheros y multitud de registro duplicados que han sido eliminados con unas sencillas sentencias en Linux.

Con todo este proceso, hemos conseguido partir de un fichero con 22.049 tweets recopilados desde inicios de 2011 (cuando la compañía salió al mercado) hasta el 31 de julio de 2017.

4.2 Primer tratamiento

En esta parte, lo que hemos hecho ha sido asignarle una nota positiva o negativa según la polaridad de los textos de los tweets. Esa asignación se le ha hecho a través del

fichero explicado ms arriba, AFINN-165, el cual viene de un estudio realizado por Finn Årup Nielsen, el cual asigna manualmente una nota a ciertas palabras que van desde el -5 hasta el 5.

El primer problema es que ese fichero viene con palabras inglesas y no hay correspondencia con palabras castellanas, con lo que he tenido que traducirlas y limpiar el fichero resultante. Por ejemplo, la palabra 'loveless' ha sido traducida literalmente por 'sin amor'. Como son dos palabras y nosotros queremos construir un diccionario, esta palabra ha sido eliminada.

Otro problema solventado ha sido '*lematizar*' dicho diccionario. La '*lematización*' es un proceso lingüístico que consiste en, dada una forma flexionada (es decir, en plural, en femenino, conjugada, etc.), hallar el lema correspondiente. El lema es la forma que por convenio se acepta como representante de todas las formas flexionadas de una palabra. Es decir, por ejemplo, la palabra abandona esta puntuada con -2. ¿Pero y si aparece la palabra abandonar? ¿O abandonado? ¿Restamos algo? Lo que he hecho ha sido cruzar este fichero con el fichero 'lemmatization-es.txt' obtenido en internet (<http://www.lexiconista.com/datasets/lemmatization/>).

Una vez solventados estos problemas, el desarrollo del programa, de forma rápida, hace lo siguiente:

- Para cada tweet, lo disecciona en los campos que hemos descargado
- Para el campo de texto:
 - o Elimina urls
 - o Elimina caracteres extraños (tildes, *, ..., /, -, ...)
- Creamos unos campos con la fecha (año, mes, día, hora, minuto)
- Asignamos el valor de la polaridad del tweet según las palabras que tiene el texto y calculamos otros campos que nos van a ayudar a entender el análisis:
 - o Contador de palabras positivas
 - o Contador de palabras negativas
 - o Si el tweet es positivo, negativo o neutro (según la nota total obtenida)

Después de un primer análisis de los resultados obtenidos, hemos tenido que diferenciar los tweets en 4 categorías, a saber:

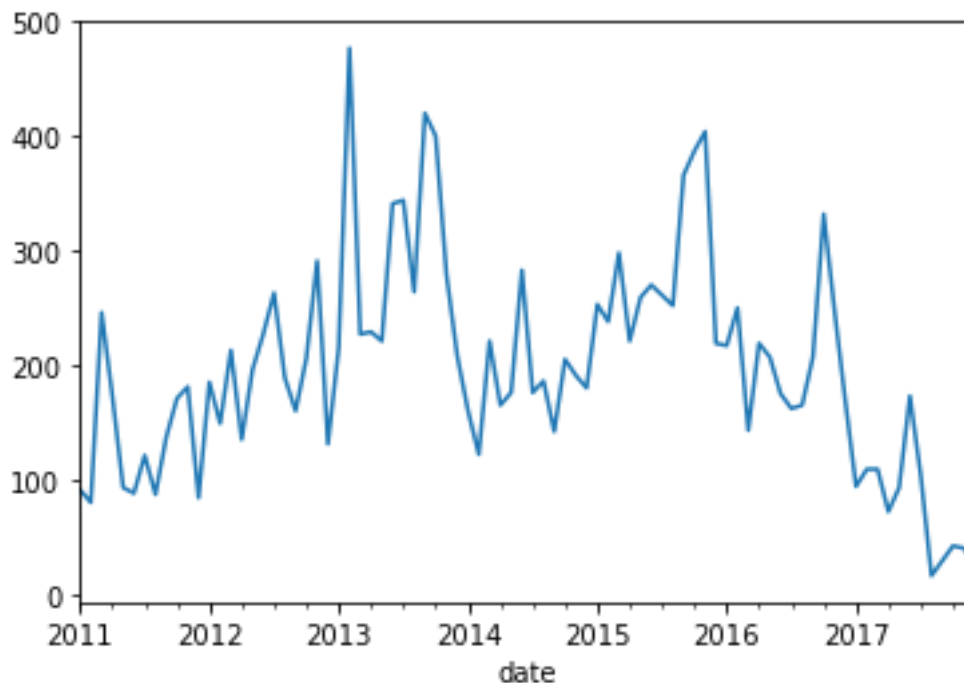
- Publicidad: son tweet emitidos desde la cuenta de Vertiseguros con propósitos exclusivamente comerciales

- Respuestas: son tweets emitidos desde la cuenta de Vertiseguros para responder a los clientes
- Clientes: son tweets de clientes o potenciales clientes en los que se refieren a la compañía (pueden ser emitidos a la cuenta @vertiseguros o un tweet genérico que contenga la palabra Verti)
- Excluidos: estos son tweets a excluir debido a quien los ha escrito: el antiguo CEO, la actual CEO, el antiguo CM de la compañía, cuentas de revistas o publicitarias,... Creo que no aportan nada al estudio y hay que eliminarlas.

4.3 Análisis exploratorio de datos

Una vez que hemos visto como hemos descargado los datos y los hemos tratado, mostramos ahora un análisis exploratorio de los datos obtenidos para ver como son y unos primeros resultados que nos muestran la polaridad de los tweets analizados, la cual usaremos como entrada para lanzar los modelos estadísticos posteriores.

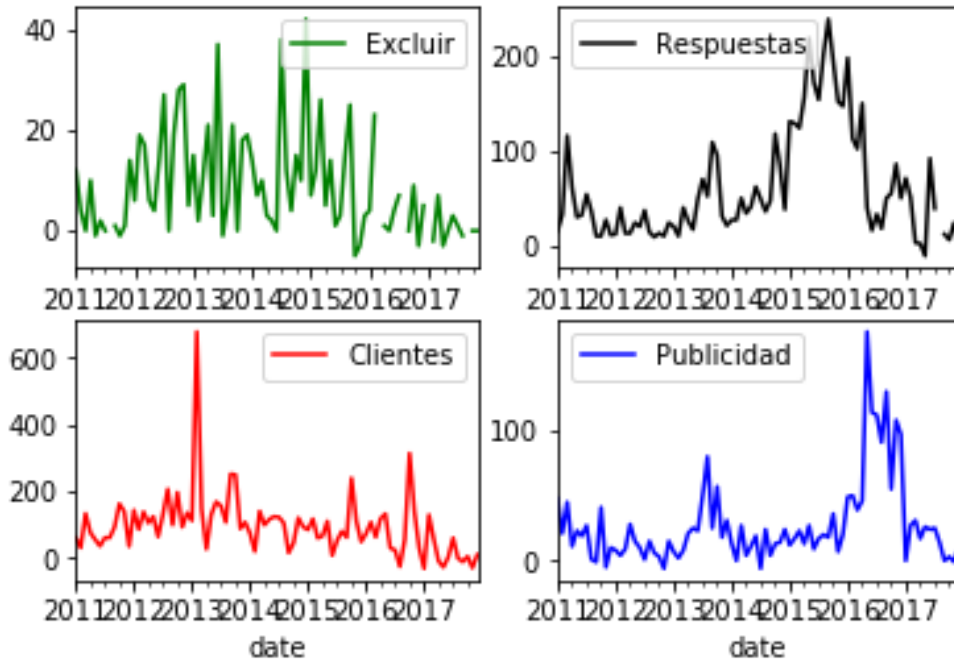
Lo primero que nos ha llamado la atención es la disparidad de la línea temporal del número de tweets recuperados por mes:



(figura 4.2)

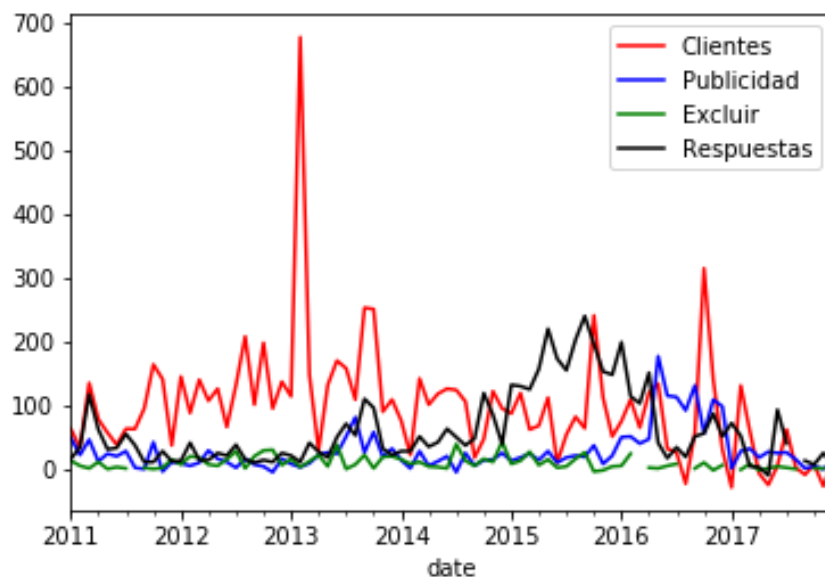
Esto nos ha hecho pensar que o no hemos recuperado correctamente los tweets o hay un funcionamiento muy atípico de la cuenta. Sobre todo, es muy preocupante el descenso del número de tweets desde 2016.

Separando los tweets por los tipos creados, obtenemos lo siguiente:



(figura 4.3)

Que si las juntamos, obtenemos la siguiente gráfica:

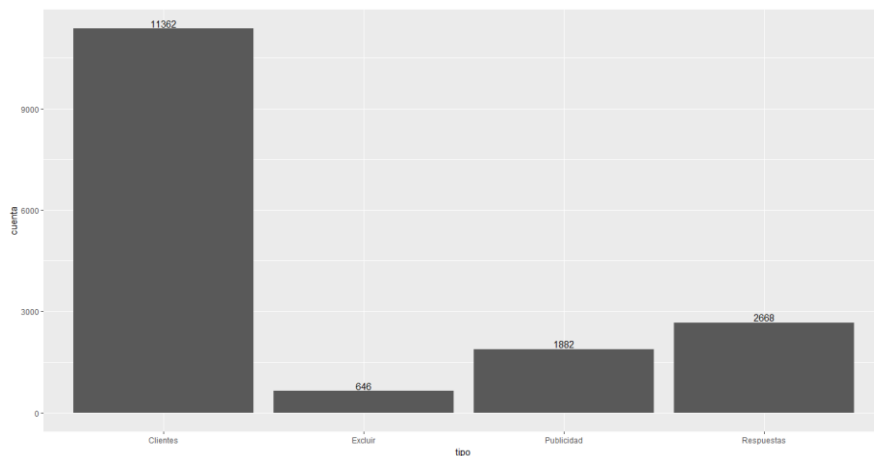


(figura 4.4)

En estos gráficos destacan varias cosas, a saber:

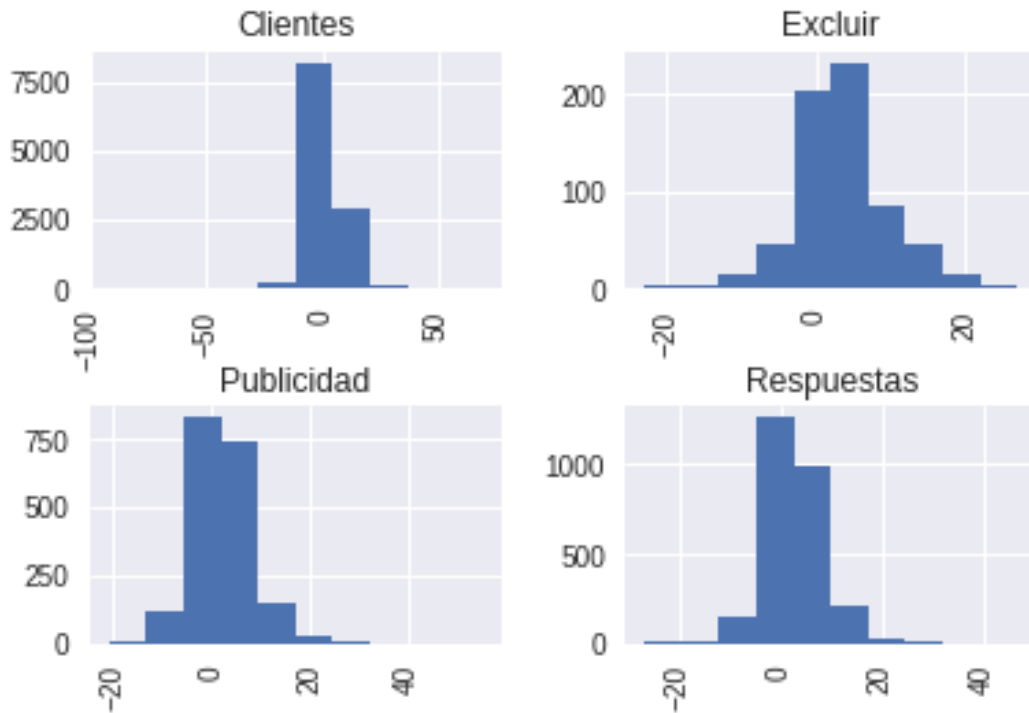
- En febrero de 2013 hay un pico de tweets de Clientes. Se deben una campaña de publicidad que hizo la compañía en la que se destacaba el concepto Despelotados, el cual aparece muchas veces en los hashtags del mes
- Aparece otro pico significativo en Octubre de 2016. Este se debe a que la compañía promocionó un concierto, Por Ellas, junto con Cadena100. También se ve reflejado en la repetición del hashtag #Porellas
- Los tweets a excluir son intermitentes en el tiempo
- Los tweets de respuestas eran casi inexistentes hasta mediados de 2013 y tuvieron su máximo esplendor a mediados de 2015, para luego bajar otra vez hasta niveles muy bajos durante 2017
- Los tweets publicitarios han sido muy bajos, hasta mediados de 2016 que subieron debido a unas campañas de publicidad de la compañía. En 2017 volvieron a bajar.

Recordamos que hemos creado 4 tipologías de los tweets recogidos con la siguiente distribución:



(figura 4.5)

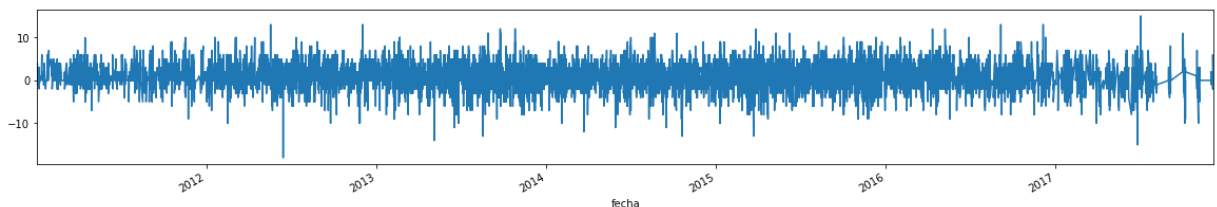
Como vemos en el gráfico, hay un total de 646 tweets catalogados como excluir y otros 1.882 como publicidad. Si vemos las notas asociadas a la totalidad de los tweets separados por tipos:



(figura 4.6)

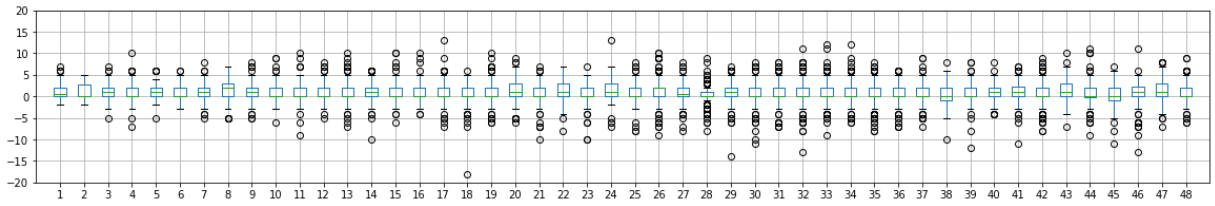
Comprobamos que tanto los tweets `excluir` como los de `publicidad` son bastante positivos. A partir de ahora, esos tweets los eliminamos del posterior análisis.

La evolución temporal de la polaridad de los tweets, la podemos ver con la siguiente gráfica:

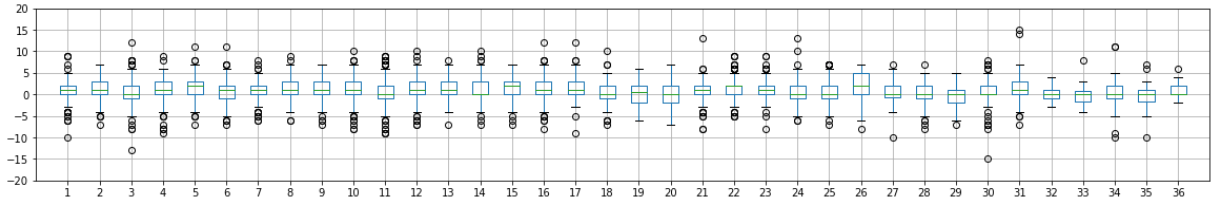


(figura 4.7)

La cual no parece aportar mucho: el valor medio de la polaridad varía alrededor del 0 pero no se ve muy bien. Las dos siguientes gráficas nos van a dar más información de una manera parecida. Se trata de una serie temporal agregada por meses / años donde los valores mostrados son unos boxplot, los cuales nos dan mucha información extra:

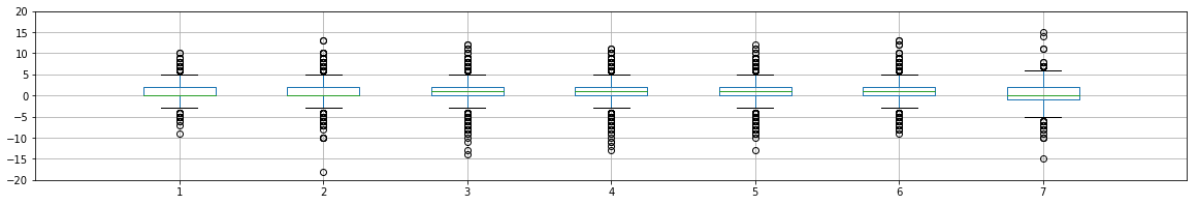


(figura 4.8.1: evolución mensual con boxplot de la polaridad de los tweets 2011 - 2014)



(figura 4.8.2: evolución mensual con boxplot de la polaridad de los tweets 2015 - 2017)

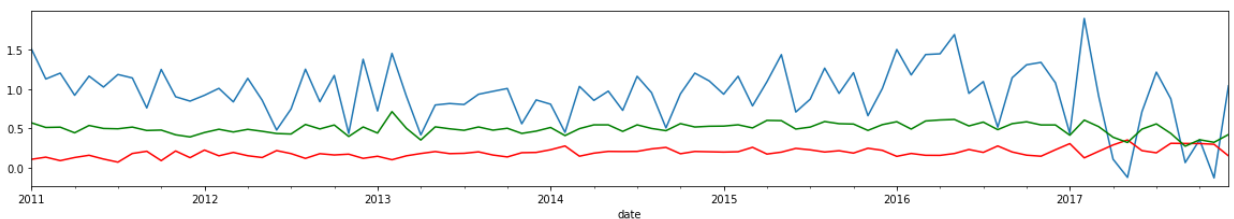
Si lo graficamos de manera anual en vez de mensual, obtenemos:



(figura 4.9: evolución anual con boxplot de la polaridad de los tweets)

En este último gráfico podemos observar como la polaridad del año 2017 (7) es eminentemente negativa, como indica el boxplot de ese año, así como los distintos outliers negativos que hay.

Esto también se puede ver con la siguiente gráfica, que muestra la evolución en el tiempo tanto de la polaridad (azul) como el número de tweets catalogados como negativos (rojo) y positivos (verde):



(figura 4.10)

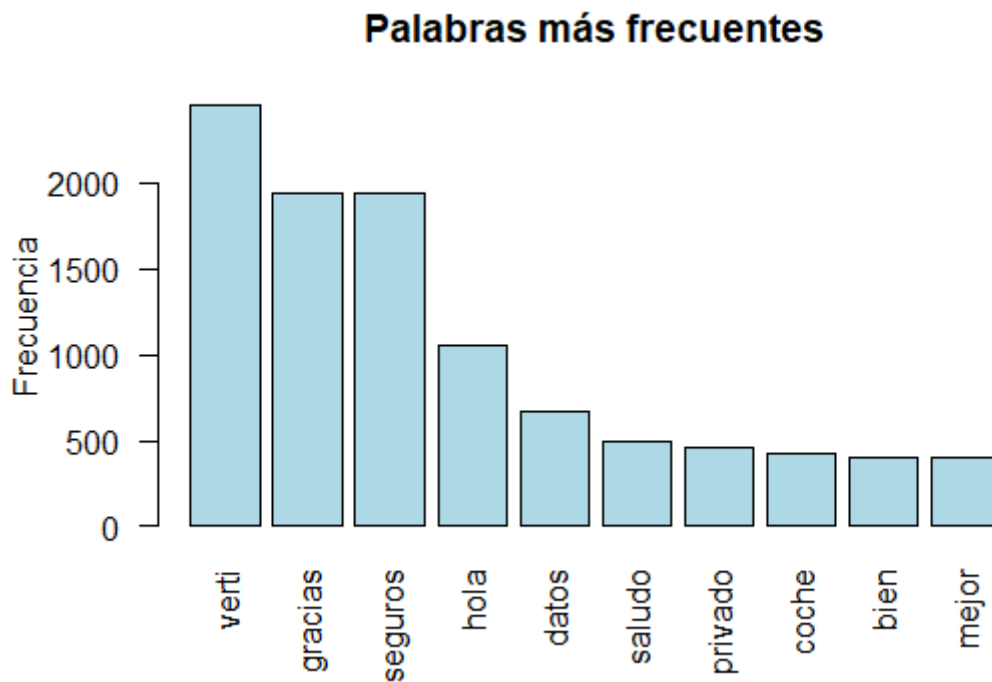
Se puede apreciar como la polaridad en 2017 es, de media, más baja que en los años anteriores así como ha aumentado el número de tweets negativos y disminuye el de positivos.

Otra de las cosas que he realizado ha sido una nube de palabras para los tweets categorizados como positivos y otra con los negativos. Estos son los resultados obtenidos para los tweets positivos con frecuencias superiores a 100:



(figura 4.11)

Y el top 10 de palabras más usadas es el siguiente:



(figura 4.12)

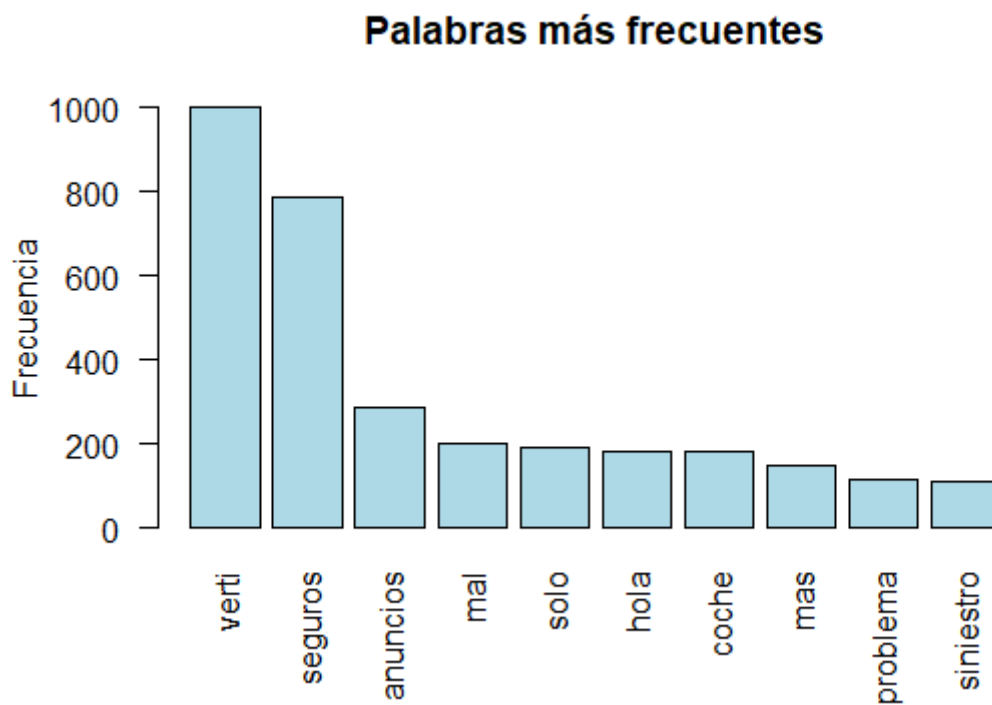
Estas palabras parecen indicar que la educación (hola, saludos, gracias) parecen indicar positividad en la polaridad de los tweets.

En cuanto a los negativos:



(figura 4.13)

Y su top 10:



(figura 4.14)

Aquí podemos ver que las principales palabras negativas indican ya de por si negatividad: mal, problema, solo. Y hay una que deberíamos tener en cuenta como compañía: aparece la palabra anuncio en esta lista. Es una tarea a revisar.

4.4 Modelización Estadística

Como hemos contado en la introducción, hemos usado hasta 7 modelos estadísticos distintos obtenidos de las librerías nltk y sklearn de Python. Estos son los modelos usados y sus librerías correspondientes:

- nltk.classify: NaiveBayesClassifier y MaxentClassifier
- sklearn.naive_bayes: MultinomialNB y BernoulliNB
- sklearn.svm: LinearSVC
- sklearn.linear_model: LogisticRegression y SGDClassifier

De cada uno de estos modelos, hemos calculado 4 diferentes estadísticos para compararlos, que son:

- Accuracy
- Precision
- Recall
- F-measure

Vamos a explicar brevemente estos 4 estadísticos usados. Partimos de lo que se conoce como matriz de confusión, que es lo siguiente:

	Clas. Verdadero	Clas. Falso
Verdadero	Verdaderos positivos	Falsos negativos
Falso	Falsos positivos	Falsos negativos

(figura 4.15)

Dónde:

Verdadero positivo: caso positivo etiquetados como positivo.

Verdadero negativo: caso negativo etiquetados como negativo.

Falso positivo: caso negativo etiquetado como positivo.

Falso negativo: caso positivo etiquetado como negativo.

Una vez construida la matriz de confusión, denominamos:

- sensitivity o recall al cociente entre los verdaderos positivos y el total de positivos.
- specificity al cociente entre los verdaderos negativos y el total de negativos.
- precision al cociente entre los verdaderos positivos y la suma de los verdaderos positivos y los falsos positivos.

	Clas. Verdadero	Clas. Falso	
Verdadero	Verdaderos positivos	Falsos negativos	$recall = \frac{truepositive}{totalpositive}$
Falso	Falsos positivos	Falsos negativos	$specificity = \frac{truenegative}{totalnegative}$
	$precision = \frac{truenegative}{truepositive + falsepositive}$		

(figura 4.16)

Una medida adicional de la precision es el concepto de accuracy, el cual definimos a través de la siguiente formula:

$$accuracy = sensitivity \cdot \frac{pos}{pos + neg} + specificity \cdot \frac{neg}{pos + neg}$$

Para terminar estos estadísticos, se define el F, como:

$$f - measure = 2 * \frac{precision * recall}{precision + recall}$$

El porqué de calcular tantos estadísticos es porque cada uno tiene sus ventajas y sus inconvenientes. Por ejemplo, el accuracy asume un coste igual para los errores de tipo b (falsos negativos) y los errores de tipo c (falsos positivos) y depende del problema, un % de accuracy puede ser pobre, bueno o excelente. Como ejemplo, me remito a las pruebas de cáncer. Un falso positivo es mucho más asumible que un falso negativo, por razones obvias.

En la recuperación de información se suele usar mas tanto el recall (cuantos de los positivos devuelve el modelo) como el precision (cuántos de los documentos devueltos son correctos), así como la medida F (f-measure) que es una ponderación de estos dos últimos estadísticos.

Hemos enfrentado también todos los modelos a la ejecución con todo el fichero de datos y a una validación cruzada con el mismo fichero pero con 5, 10, 15, 20 y 25 partes del mismo. Los datos óptimos nos han llevado a usar una validación cruzada de 10 ficheros.

Por último, hemos realizado el mismo análisis pero usando los ‘N-gramas’. Un ‘N-grama’ no es más que una sub secuencia de n elementos consecutivos en una secuencia dada. Para nuestro caso, hemos usado los ‘N-gramas’ de orden 1 y 2 (llamados bi-gramas). El uso de los bi-gramas nos va a dar la posibilidad de que el sentido de la frase pueda incrementarse, como por ejemplo:

“Esta casa es **grande**”

“Esta casa es **muy grande**”

En este ejemplo vemos que se incrementa el valor de la frase. Pero en otros casos le puede dar un cambio total al sentido de la frase, como por ejemplo:

“Esta casa no es grande”

Como vemos, ahora el sentido de la frase ha cambiado totalmente.

Vamos a usar tanto los ‘N-gramas’ de orden 1 y 2 para la totalidad de los textos as como eliminando los stop words de los mismos.

Una vez vistas las definiciones, vamos con los resultados obtenidos. El programa ejecuta una salida como pueda ser la siguiente:

```

-----
RESULTADO INDIVIDUAL (Naive Bayes)
-----
accuracy: 0.741732804233
precision 0.694131948981
recall 0.770972062666
f-measure 0.698685157819
-----
RESULTADO INDIVIDUAL (BernoulliNB)
-----
accuracy: 0.791666666667
precision 0.6962526963
recall 0.578853328996
f-measure 0.586533428132
-----
RESULTADO INDIVIDUAL (MultinomialNB)
-----
accuracy: 0.807208994709
precision 0.728529532877
recall 0.630121336275
f-measure 0.651034125375
-----
RESULTADO INDIVIDUAL (Maximum Entropy)
-----
accuracy: 0.779100529101
precision 0.6400132714
recall 0.502681394427
f-measure 0.445177238345
-----
RESULTADO INDIVIDUAL (SVM)
-----
accuracy: 0.808862433862
precision 0.72595821226
recall 0.64941593892
f-measure 0.670397904875
-----
RESULTADO INDIVIDUAL (Logistic Regresion)
-----
accuracy: 0.808862433862
precision 0.730516080778
recall 0.637081524557
f-measure 0.658654751257
-----
RESULTADO INDIVIDUAL (SGDClassifier)
-----
accuracy: 0.805224867725
precision 0.722139993833
recall 0.630456828279
f-measure 0.650788802175

```

(figura 4.17)

(ejemplo de salida correspondiente a la ejecución normal sin validación cruzada).

Una vez analizadas y juntadas todas las salidas, hemos decidido quedarnos con las salidas para la totalidad de los ficheros y la validación cruzada de 10 partes, para los modelos ejecutados con los ‘uni-gramas’ sin más y eliminando las stop words, y con ‘bi-gramas’ con y sin stop words, obteniendo los siguientes resultados:

accuracy	NORMAL		NORMALSW		BIGRAMS		BIGRAMASSW		
Modelo	Total	V.C.	Total	V.C.	Total	V.C.	Total	V.C.	RK
SVM	80,9%	93,3%	81,0%	93,3%	79,6%	93,3%	79,9%	93,3%	1
Logistic Regresion	80,9%	92,4%	80,8%	92,5%	79,4%	93,2%	79,4%	93,2%	2
MultinomialNB	80,7%	89,7%	81,0%	90,1%	79,0%	92,0%	80,1%	92,0%	3
SGDClassifier	80,5%	90,4%	80,9%	92,3%	78,3%	92,0%	79,3%	92,4%	4
BernoulliNB	79,2%	87,2%	77,7%	87,6%	76,1%	87,6%	76,1%	87,5%	5
Maximum Entropy	77,9%	78,7%	78,1%	81,0%	77,4%	80,9%	77,1%	85,2%	6
Naive Bayes	74,2%	76,3%	79,0%	83,2%	70,4%	73,1%	73,0%	76,7%	7

(figura 4.18)

precision	NORMAL		NORMALSW		BIGRAMS		BIGRAMASSW		
Modelo	Total	V.C.	Total	V.C.	Total	V.C.	Total	V.C.	RK
Logistic Regresion	73,1%	90,3%	73,3%	90,3%	69,5%	91,2%	69,6%	91,3%	1
SVM	72,6%	90,6%	72,9%	90,7%	69,9%	90,8%	70,6%	91,0%	2
MultinomialNB	72,9%	84,8%	73,4%	85,4%	68,6%	87,4%	71,2%	87,3%	3
SGDClassifier	72,2%	86,8%	73,3%	89,0%	66,6%	88,8%	69,3%	89,2%	4
Maximum Entropy	64,0%	86,6%	68,8%	88,6%	51,0%	88,6%	54,1%	91,2%	5
Naive Bayes	69,4%	73,6%	71,5%	77,9%	67,6%	72,3%	68,3%	74,1%	6
BernoulliNB	69,6%	82,7%	63,7%	84,5%	43,3%	86,6%	43,4%	87,3%	7

(figura 4.19)

recall	NORMAL		NORMALSW		BIGRAMS		BIGRAMASSW		
Modelo	Total	V.C.	Total	V.C.	Total	V.C.	Total	V.C.	RK
SVM	64,90%	89,60%	64,70%	89,50%	62,30%	89,40%	62,40%	89,10%	1
Naive Bayes	77,10%	83,80%	76,60%	87,90%	75,30%	82,40%	75,60%	84,80%	2
MultinomialNB	63,00%	85,60%	63,90%	86,10%	59,10%	90,60%	62,30%	91,10%	3
Logistic Regresion	63,70%	87,20%	62,80%	87,20%	60,60%	88,70%	60,40%	88,50%	4
SGDClassifier	63,00%	85,30%	63,00%	88,60%	57,60%	87,60%	60,10%	88,70%	5
BernoulliNB	57,90%	78,30%	53,40%	77,10%	49,20%	75,00%	49,20%	74,40%	6
Maximum Entropy	50,30%	51,70%	51,10%	57,10%	50,10%	56,80%	50,50%	66,60%	7

(figura 4.20)

f-measure	NORMAL		NORMALSW		BIGRAMS		BIGRAMASSW		
Modelo	Total	V.C.	Total	V.C.	Total	V.C.	Total	V.C.	RK
SVM	67,00%	90,10%	66,80%	90,10%	64,10%	90,10%	64,20%	90,00%	1
Logistic Regresion	65,90%	88,60%	64,90%	88,60%	62,20%	89,80%	61,90%	89,80%	2
MultinomialNB	65,10%	85,20%	66,10%	85,70%	60,30%	88,80%	64,20%	89,00%	3
Naive Bayes	69,90%	73,40%	73,10%	79,80%	66,70%	70,60%	68,60%	73,80%	3
SGDClassifier	65,10%	85,80%	65,20%	88,70%	58,30%	88,10%	61,60%	88,90%	5
BernoulliNB	58,70%	80,10%	51,80%	79,80%	44,00%	78,70%	44,00%	78,40%	6
Maximum Entropy	44,50%	47,40%	46,50%	57,00%	44,60%	56,60%	46,10%	70,60%	7

(figura 4.21)

La columna final de RK (ranking) está calculando las posiciones relativas de cada valor dentro de su columna, luego se suman todos los valores y se vuelve a calcular la posición relativa dentro de la columna.

Con estos valores, ya podemos ver como el modelo de las SVM es el modelo que siempre sale en primera posición, salvo para el estadístico de la precisión, en el que queda segundo. Es reseñable como suben los porcentajes de los estadísticos cuando se lanzan con validación cruzada, en casi todos los modelos, pero especialmente entre los modelos que quedan en las primeras posiciones (aumentos de hasta 20-25% en algunos casos).

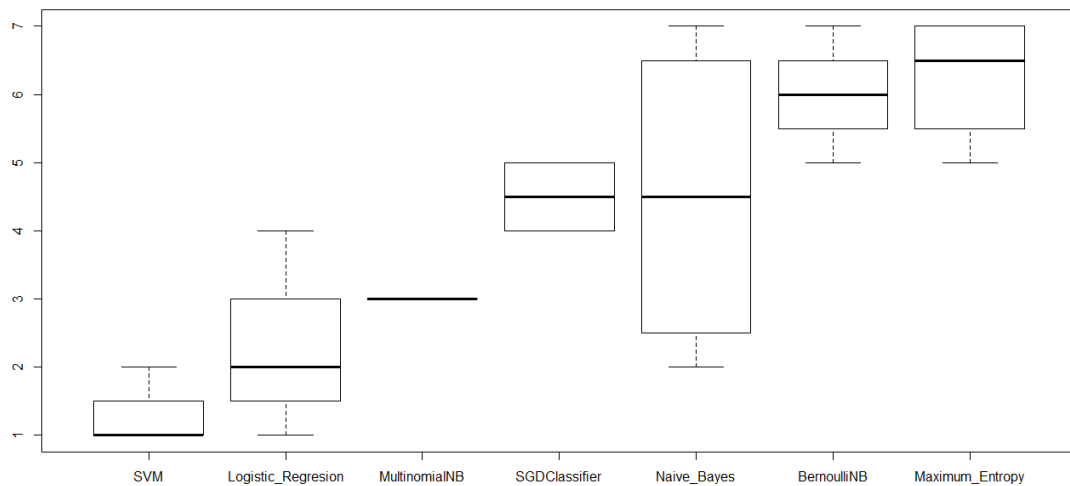
También me gustaría destacar que, en contra de nuestra primera impresión, pensábamos que los bi-gramas funcionarían mejor que los unigramas, pero vemos como no es así, es, de promedio, un 7% inferior en la suma de todos los estadísticos.

Si volvemos a *'rankinear'* todos los modelos y sus estadísticos, obtenemos la siguiente tabla:

Modelo	accuracy	precision	recall	f-measure	Ranking	Media	DesvEst
SVM	1	2	1	1	1	1,25	0,5
Logistic Regresion	2	1	4	2	2	2,25	1,26
MultinomialNB	3	3	3	3	3	3	0
SGDClassifier	4	4	5	5	5	4,5	0,58
Naive Bayes	7	6	2	3	3	4,5	2,38
BernoulliNB	5	7	6	6	6	6	0,82
Maximum Entropy	6	5	7	7	7	6,25	0,96

(figura 4.22)

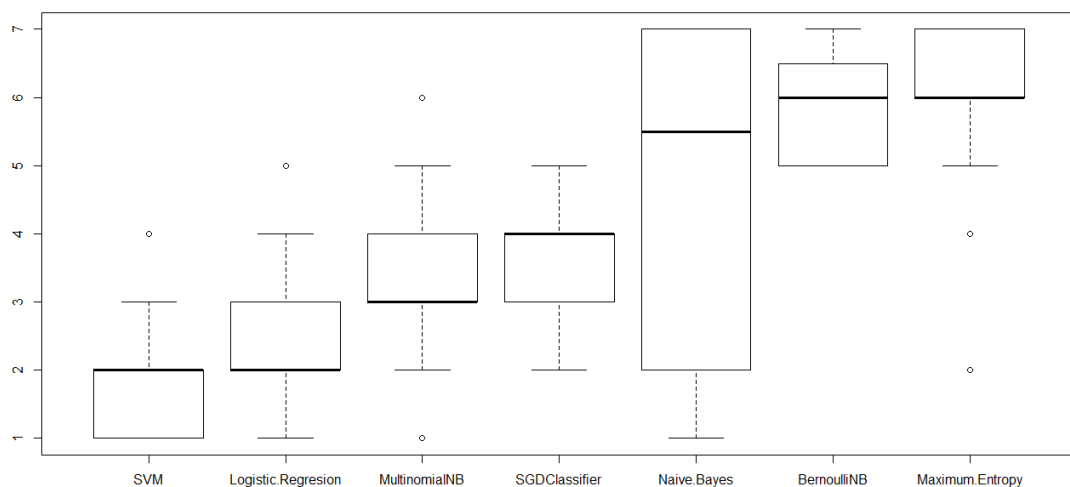
Volvemos a calcular la posición relativa dentro de cada columna, sumamos las posiciones y volvemos a *'rankinear'* con el resultado. Vemos un orden por % de acierto de cada uno de los modelos evaluados. La media es la posición media obtenida dentro de los distintos estadísticos calculados, así como la desviación estándar. Con estos datos, podemos mostrarlos en el siguiente gráfico:



(figura 4.23)

En este gráfico podemos ver como varían las posiciones según los diferentes estadísticos calculados en el trabajo y su variación dentro de las posiciones relativas que ocupan. Están ordenados por el orden que nosotros seleccionaríamos los distintos modelos según los resultados obtenidos. El modelo de Naive Bayes lo hemos puesto en quinto lugar en vez del SGDClassifier sobre todo por tener más dispersión.

Yendo más al detalle, si cogemos todos los resultados obtenidos, es decir, las 32 posibilidades dadas por los 4 estadísticos, las 4 posibilidades de los modelos ('uni-grama', 'bi-grama' con y sin stop words) y las dos posibilidades de validación cruzada o no, repetimos el mismo cuadro del boxplot, obteniendo:



(figura 4.24)

Esta vez comprobamos que, a pesar de tener el mismo orden, los datos tienen ahora más dispersión entre los datos, cosa normal pues pasamos de tener solamente 4 datos a tener 32.

Otro resultado interesante ha sido el calcular cuales han sido las 10 palabras más relevantes, aquellas con las características más informativas que hemos obtenido en un modelo de Naive Bayes. Estas palabras son las siguientes:

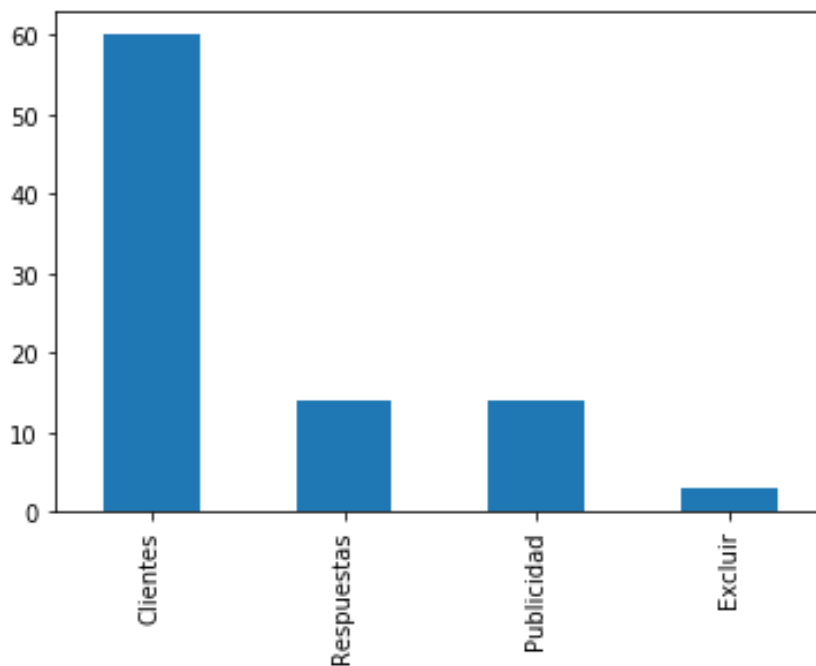
trate = True	neg : pos	=	71.8 : 1.0
puta = True	neg : pos	=	64.7 : 1.0
juro = True	neg : pos	=	57.7 : 1.0
baje = True	neg : pos	=	48.3 : 1.0
odio = True	neg : pos	=	45.9 : 1.0
guardia = True	neg : pos	=	43.5 : 1.0
enfermedad = True	neg : pos	=	31.8 : 1.0
perros = True	neg : pos	=	29.4 : 1.0
saludo = True	pos : neg	=	27.4 : 1.0
duele = True	neg : pos	=	27.1 : 1.0

Esta tabla nos indica la relación de que si un tweet contenga una de estas palabras el tweet tenga la posibilidad marcada a la derecha del todo de ser positivo o negativo, según indiquen los datos de la segunda columna. Por ejemplo, si un tweet tiene la palabra 'trate' dentro del texto, la posibilidad de que el tweet sea negativo es de 71.8 a 1 de que sea positivo. En estas 10 palabras, solamente se ha colado una palabra cuya probabilidad de aparecer haga que el tweet sea catalogado como positivo con una proporción de 27.4 a 1. Esa palabra es saludo. Parece ser que la educación dentro de un tweet está relacionada con la positividad del mismo. A nivel negocio, es preocupante que aparezca la palabra perro, puesto que uno de los productos que tiene la compañía es un seguro de mascotas, dirigido a perros. Esto es una posible vía de investigación a futuro para comprobar si tenemos un problema con este producto en este canal.

5. Evaluación de modelos predictivos con datos reales

Una vez que hemos visto que hay un modelo que ha resultado mejor sobre el resto de los estudiados, lo hemos usado para calcular las predicciones para los tweets con datos reales. Para ello, hemos descargado los tweets al igual que explicamos en la descarga de los datos iniciales pero esta vez con las fechas del mes de agosto de 2017 (hasta el día 21, nada más).

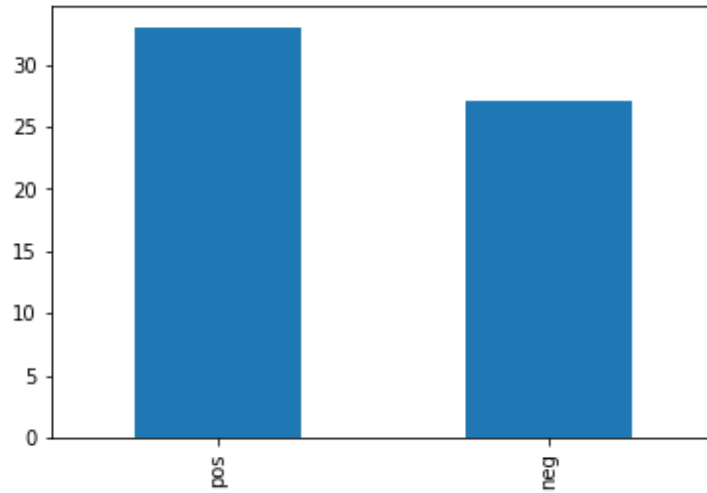
Una vez descargados los datos, este es el reparto de los mismos:



Cientes	61	Excluir	3
Respuestas	14	Publicidad	14

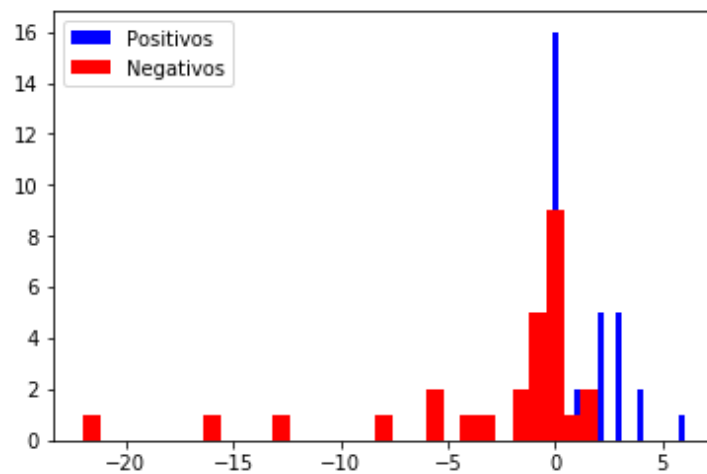
(figura 5.1)

Solamente vamos a tratar los tweets de los clientes. Para ello, hemos vuelto a calcular los parámetros de la SVM (linearSVC para ser más exactos) y hemos pasado el modelo a estos tweets. Los primeros resultados a comentar son las distribuciones de las polaridades de dichos tweets predichas:



(figura 5.2)

Y el reparto de las polaridades dadas anteriormente es (azul para los valores predichos como positivos y en rojo los negativos):



(figura 5.3)

Como podemos observar, hay unos casos que podríamos considerar como falsos positivos (2) y falsos negativos (3), que son:

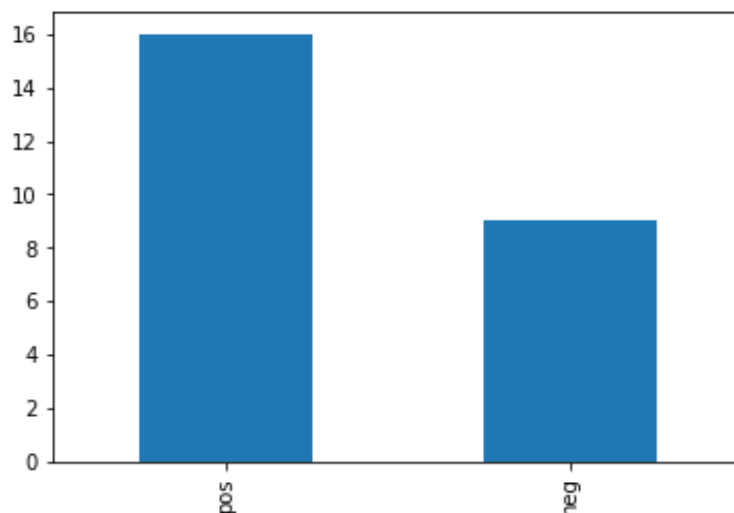
Predicción	Valor previo	Texto
NEGATIVO	2	A este paso, al metro de Granada no lo va a querer asegurar ni @vertiseguros . Granaínos, rojo=no pasar, que viene el scalextric!!!
NEGATIVO	2	@vertiseguros primer parte y primera sorpresa desagradable.mi póliza no incluye coche de sustitución.no renovare ni lo recomendaré.
NEGATIVO	1	Me han subido la póliza un 40% y sin dar un solo parte... jajaja adios @vertiseguros
POSITIVO	-2	A lo mejor si se quema la casa entera y muere alguien, entonces si que hubieran respondido que asco de gente.
POSITIVO	-1	Desconectando del trabajo... @NoemiGarGon @vertiseguros @beeva_es pic.twitter.com/2HdVg6hJQo

(figura 5.4)

En estos casos, hay alguno de ellos que podemos explicar. Por ejemplo, el último parece que se trata de un doble sentido que no hemos podido recoger correctamente. El usuario comenta que desconecta del trabajo, algo que es positivo mientras que la valoración previa del tweet ha sido negativa, por no poder detectar ese doble sentido.

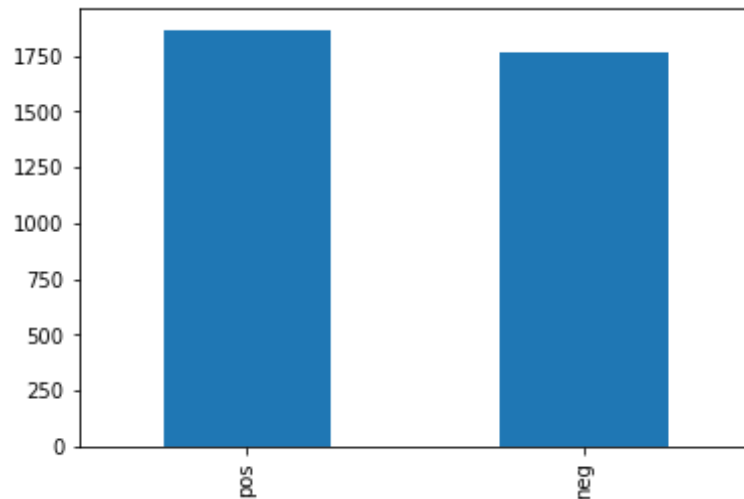
5.1 Los Tweets neutros

Para terminar, voy a analizar los tweets catalogados como neutros. Estos tweets no fueron introducidos en el entrenamiento de los modelos pero ahora somos capaces de darle una polaridad. En los casos reales hay un total de 25 casos, con la siguiente polaridad calculada:



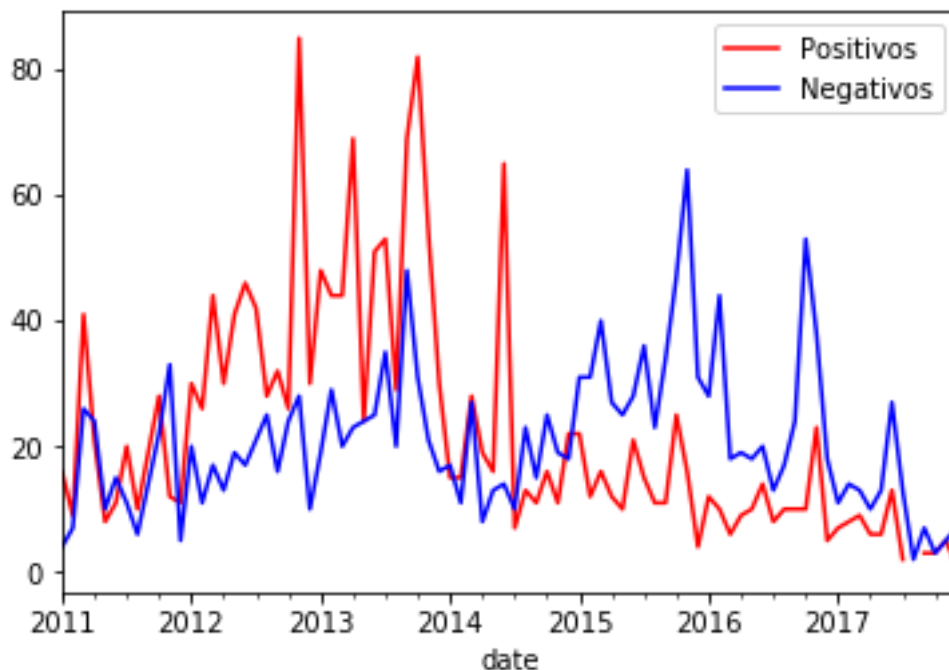
(figura 5.5)

En este caso, vemos que la mayoría de los tweets clasificados como neutros al principio son catalogados mayoritariamente como positivos. Es más, recuperando todos los tweets neutros que no entraron en el entrenamiento, al pasarles el modelo, obtenemos:



(figura 5.6)

Comprobamos como hay una porción un poco mayor de positivos pero no tan acrecentado como en lo datos reales. De hecho, si representamos estos tweets en el tiempo, obtenemos las siguientes series:



(figura 5.7)

Estas series muestran algo muy preocupante. Mientras al principio de la recolección de los datos, los tweets neutros se clasifican de manera positiva mayor que la negativa, a partir de mediados del año 2014 esta tendencia se invierte totalmente, predominando la mayoría de tweets negativos. Como compañía, tendrá que mirar con más detalle esto último.

6. CONCLUSIONES

Como hemos podido ver en el documento, el análisis no ha sido sencillo, debido, sobre todo, a las dificultades encontradas a la hora de obtener los datos, que provocaran que el diseño del trabajo cambiara radicalmente. Al tener menos información con la que trabajar, ha sido más difícil poder realizarlo y, por otro lado, trabajar con información no todo lo completa que quería desde un principio.

A continuación, a la hora de calcular la polaridad inicial de los tweets recuperados, hemos tenido que trabajar con el fichero con las palabras y las notas hasta que lo hemos '*lematizado*' y así obtener una polaridad más fina.

En el análisis exploratorio de los datos hemos obtenido unas cuantas conclusiones a mirar:

- Hay una disparidad mensual con el número de tweets, bajando considerablemente a finales de 2015
- La evolución anual de la polaridad inicial de los tweets en los últimos tiempos está tendiendo hacia la negatividad
- En la nube de palabras negativas aparece una que hay que vigilar: anuncio. ¿Los anuncios de Verti son molestos? ¿Es algo puntual en el tiempo o es de manera general? Es un punto anotado para los próximos pasos del proyecto.

Una vez que hemos diferenciado los tweets con la polaridad, hemos estudiado hasta 7 modelos estadísticos diferentes y hemos comparado los resultados según varios parámetros: uni-gramas y bi-gramas; accuracy, recall, precision y f-measure; validación cruzada frente a totalidad del fichero y textos con y sin stop words. Al final, hemos comparado todos los resultados según cada modelo y hemos creado una jerarquía de asignación de valor en el que ha salido ganador el modelo LinearSVC, que es una SVM con kernel lineal. También es verdad que hemos lanzado todos los modelos con los parámetros por defecto y que si hubiéramos tuneado cada uno de los modelos es posible que los resultados hubieran cambiado, pero hemos creído que se salía del alcance del trabajo y lo podemos anotar también en las tareas futuras por hacer.

Otra de las cosas que han salido y que hay que vigilar, es que entre las palabras más relevantes del modelo estadístico de Naive Bayes ha sido la palabra perro, y de forma negativa con una proporción de 29.1:1. La compañía tiene un producto especial de mascotas (eminentemente perros), con lo que aparece otra línea de investigación futura para revisar, mirar los tweets que contienen palabras referentes a este producto (perro, mascota, can,...)

Para terminar, hemos recuperado los datos reales de Twitter con los mismos criterios que en la recopilación de los datos pero solamente los datos de agosto de 2017 (del 1 al 21). A esos datos les hemos pasado el modelo estadístico ganador en la comparación ya explicada (linearSVC) para ver los resultados reales. Los datos han vuelto a mostrar algo preocupante: hay mayoría de tweets negativos. Es más, al pasar la totalidad de los tweets definidos como neutros al principio y que no usamos como datos de entrenamiento de los modelos, muestran que las predicciones de los mismos han sufrido una inversión en la polaridad de los tweets, predominando claramente los tweets negativos. Esto es algo que también queda en las acciones futuras para revisar por parte de la compañía e investigar que ha pasado con la cuenta.

Como hemos podido ver en todo el documento, los modelos estadísticos han sido entrenados con los parámetros por defecto, con lo que tenemos un amplio margen de mejora si tuneamos dichos modelos y encontramos los parámetros más óptimos para cada uno de ellos. Ese trabajo, aparte de tiempo, llevaría tal vez en una mejora del software, puesto que con el portátil con el que he desarrollado los cálculos a veces se ha quedado un poco corto, sobre todo con los algoritmos de las SVM, la validación cruzada y los bi-gramas.

Otra mejora sería obtener los datos de manera más fina. Tal y como hemos visto, Twitter impone unas restricciones a la captura de sus datos y ha hecho muy difícil la obtención de los mismos. Hablando con el community manager de la cuenta seguramente nos podría orientar más sobre términos clave para la recuperación de dichos datos.

Para terminar, me gustaría resaltar la potencia de estos algoritmos para el sector donde trabajo actualmente (Seguros). Otra de las posibles aplicaciones que se me ocurren es la detección de fraudes con los datos de captura de los partes de accidentes. Sería interesante ver qué palabras o características de los textos posibilitarían detecciones tempranas de posibles fraudes a estudiar por parte de la compañía.

7. Bibliografía

- Introducción a la minería de datos (Hernandez, Ramirez y Ferri).
<http://users.dsic.upv.es/~flip/LibroMD/>
- Técnicas de análisis de datos en investigación de mercados (Teodoro Luque Martínez)
<https://www.edicionespiramide.es/libro.php?id=1734737>
- Twitter Sentiment Classification using Distant Supervision.
<http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf>.
- Minera de datos.
https://en.wikipedia.org/wiki/Data_mining.
- Aprendizaje supervisado.
https://en.wikipedia.org/wiki/Supervised_learning.
- SVM.
<http://scikit-learn.org/stable/modules/svm.html>.
- Linear SVC.
<http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>
- Naives Bayes.
http://scikit-learn.org/stable/modules/naive_bayes.html.
- Verti.
<https://www.salaprensa.mapfre.es/ficha-nota-prensa/441/mapfre-lanza-verti-compania-de-venta-directa-de-seguros-de-automoviles-y-hogar>
- Afinn.
<https://github.com/fnielsen/afinn/tree/master/afinn/data>
- R.
<http://www.ugr.es/~batanero/pages/ARTICULOS/libroR.pdf>
- Errores en modelos.
http://wwwae.ciemat.es/~cardenas/curso_MD/precisionerrores.pdf
- Pandas.
<http://pandas.pydata.org/>