CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL

Programa de Experto en Data Science

Memoria del proyecto final

Proyecto EDS3
Comentarios de restaurantes en Tripadvisor

Autor:
Alicia Aguirre de la Cruz

Director:
Carlos J. Gil Bellosta

Fecha: 22/07/2018

Índice

1. Introducción
   1.1 Descripción del problema……p.3
   1.2 Estructura de la memoria……p.3

2. Tecnologías utilizadas……p.3

3. Estudio de los datos
   3.1 Descripción del conjunto de datos……p.4
   3.2 Fuente de datos y Web Scraping…… p.4
   3.3 Problemas de los datos y limpieza……p.5
   3.4 Análisis exploratorio de los datos…… p.5

4. Análisis de sentimiento……p.6

5. Modelo
   5.1 Preparación de los datos……p.8
   5.2 Construcción del modelo……p.9
   5.3 Evaluación del modelo……p.9

6. Resultados
   6.1 Evaluación de resultados……p.10
   6.2 Interpretación de variables……p.10

7. Comentarios……p.12

8. Bibliografía……p.13

1. Introducción

1.1 Descripción del problema

Entender aquellos motivos que explican la puntuación que los clientes dan a restaurantes a partir de análisis de los comentarios, buscando características que tienden a mencionarse en los comentarios con notas altas y otras que sean destacables en comentarios con notas bajas.

1.2 Estructura de la memoria

Para el desarrollo del proyecto, pueden destacarse tres partes principales:

En primer lugar, para la obtención de la información necesaria se lleva a cabo un *scrapeo* de la página de Tripadvisor. Con esta técnica conseguimos los datos necesarios para el posterior procesamiento de texto.

A continuación, se realiza un análisis de sentimiento de los comentarios y la correspondiente creación de variables que facilitarán el entrenamiento del modelo en la siguiente fase.

Y, por último, empleo de modelos predictivos y posterior evaluación de resultados, así como explicación de variables importantes que intervienen en la clasificación de los comentarios.

2. Tecnologías utilizadas.

Para la realización de la parte práctica, se han utilizado los siguientes lenguajes de programación y librerías.

-Web Scraping: Se ha utilizado Python como lenguaje de programación, Pandas y Selenium como librerías.

-Análisis de sentimiento: R como lenguaje, y entre las librerías principales, data.table, dplyr, ggplot2, stringr, tm, textcat, tidytext, RTextTools y RYandexTranslate

-Machine learning: De nuevo R, y librerías como data.table, dplyr, caret, RTextTools, xgboost, ROCR y lime.

3. Estudio de los datos

3.1 Descripción del conjunto de datos

En el proceso de obtención del conjunto de datos, se pretende conseguir una muestra de comentarios e información correspondiente a restaurantes de los diferentes barrios de Madrid, entre ellos Argüelles, Zona Centro, Chamberí, Chueca, Huertas, La Latina, Lavapiés, Plaza Mayor, Paseo del Prado, Barrio de Salamanca y Sol.

El *dataset* con el que se trabajará, tendrá los siguientes campos: Nombre del restaurante, puntuación del restaurante, código postal, título del comentario, comentario completo y puntuación del comentario.

Cabe destacar, que tanto los comentarios como la nota correspondiente a cada uno de ellos serán en lo que principalmente se basará el posterior análisis y construcción del modelo predictivo.

3.2 Fuente de datos y Web Scraping

Para llevar a cabo la extracción de información, se ha elegido la página web de Tripadvisor, y para ello se ha empleado Web Scraping, que es una técnica utilizada para extraer información de sitios Web, normalmente simulando la navegación de un humano. Esto nos permite navegar por la web y extraer la información necesaria conociendo los XPath correspondientes a los campos donde se encuentra la información. XPath es un lenguaje que permite construir expresiones que recorren y procesan un documento XML, lo que facilita la búsqueda y selección datos teniendo en cuenta la estructura jerárquica del XML.

Uno de los principales riesgos que pueden sufrirse a la hora de emplear esta técnica, es que es posible ser vaneado con facilidad, por lo que el grueso del *scrapeo* deberá hacerse poco a poco.

Además, es necesario controlar bien los tiempos para dejar que el contenido de la página se muestre completamente y evitar que no se encuentren ciertos XPath, debido al tiempo de carga de la página.

Finalmente, logran extraerse 2739 comentarios.

3.3 Problemas de los datos y limpieza

A la hora de depurar los datos nos encontramos con comentarios demasiado largos. Estos serán eliminados para evitar que alteren el peso de una palabra que aparece repetidas veces en un mismo comentario y, por tanto, puedan dar información errónea.

Además, se eliminan aquellos que tienen una longitud menor que 5, pues no nos aportarán gran interés.

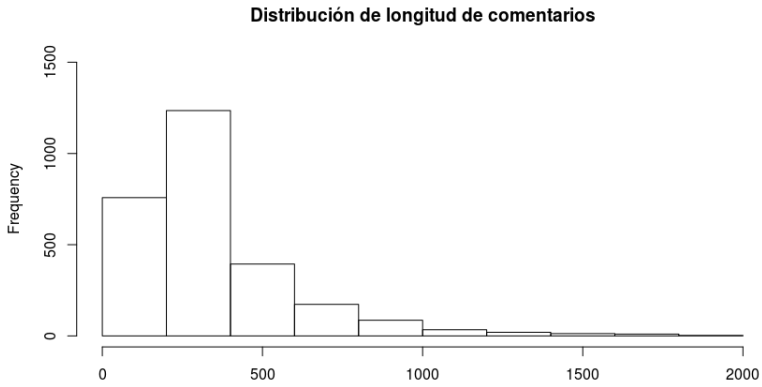
Por otro lado, nos encontramos con la necesidad de traducir los comentarios al idioma inglés, ya que muchas de las características asociadas al análisis de sentimiento serán atribuidas a los comentarios según un listado de referencia en inglés. Este listado contiene palabras a las que ya se han atribuido características positivas o negativas previamente, lo que facilitará en gran medida el trabajo.

Para la traducción se utiliza el paquete Yandex de R.

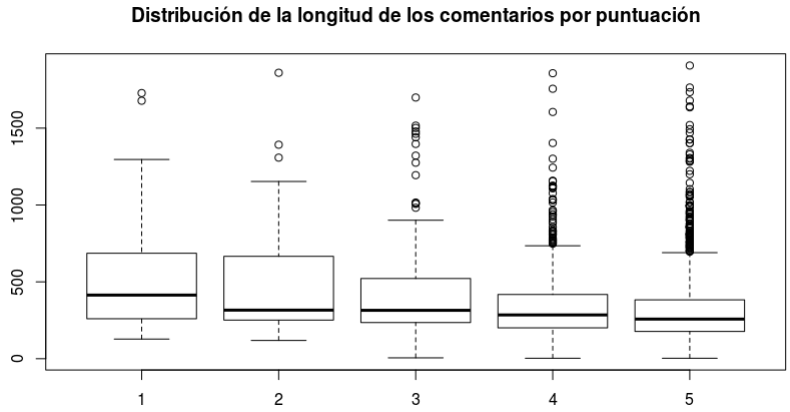
3.4 Análisis exploratorio de los datos

Haciendo un pequeño análisis de los datos, podemos ver cómo es su distribución según las puntuaciones y según la longitud de los comentarios:

Se puede observar que las puntuaciones están desbalanceadas, punto para tener en cuenta en futuros análisis.



Por otro lado, cabe destacar que la longitud de los *reviews* es mayor para los comentarios negativos que para los positivos. Esto tendrá un fuerte impacto a la hora de predecir resultados.



4. Análisis de sentimiento

Una vez hecha la limpieza de los datos se crea un *bag of words*, que consiste en la construcción de una matriz que contenga como columnas las palabras de las que está compuesto el texto, siendo las filas cada uno de los comentarios, y su valor la suma de las veces que aparece cada una de las palabras.

Posteriormente se crean nuevas variables que contienen la información en referencia a la afinidad positiva o negativa de los comentarios, para más tarde ver si este sentimiento está ligado con la puntuación. Estas nuevas *features* veremos que son de gran ayuda para el estudio y entrenamiento de los datos.

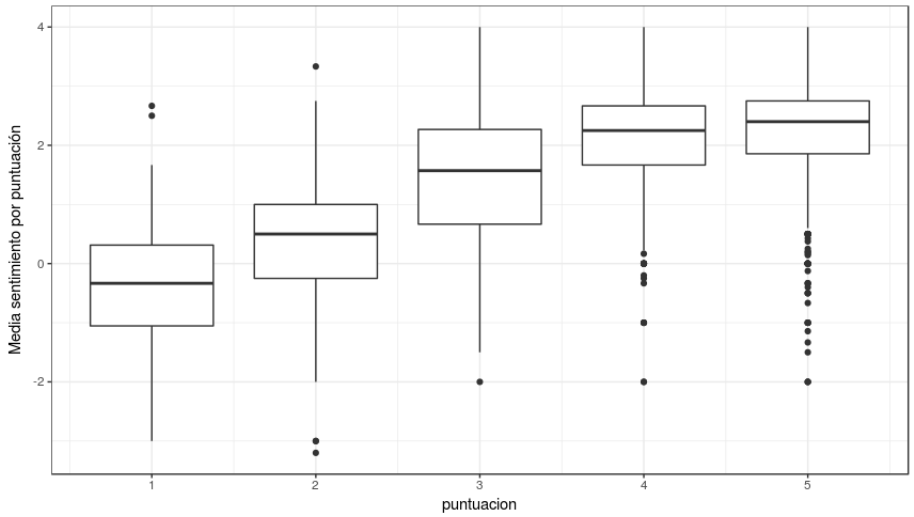
Una vez ordenado el *dataset,* poniéndolo en formato lago y habiendo añadido las puntuaciones según sentimiento para cada palabra, ahora los datos tienen este aspecto.

	puntuacion	review.id	word	afinn_score	bing_sentiment
1664	4	27	good	3	positive
1665	4	27	and	NA	<NA>
1666	4	27	creative	2	positive

Además, se ha asignado una puntuación positiva o negativa calculando la media de puntuación de todas las palabras por comentario, lo que da como resultado el *dataset* con el siguiente aspecto.

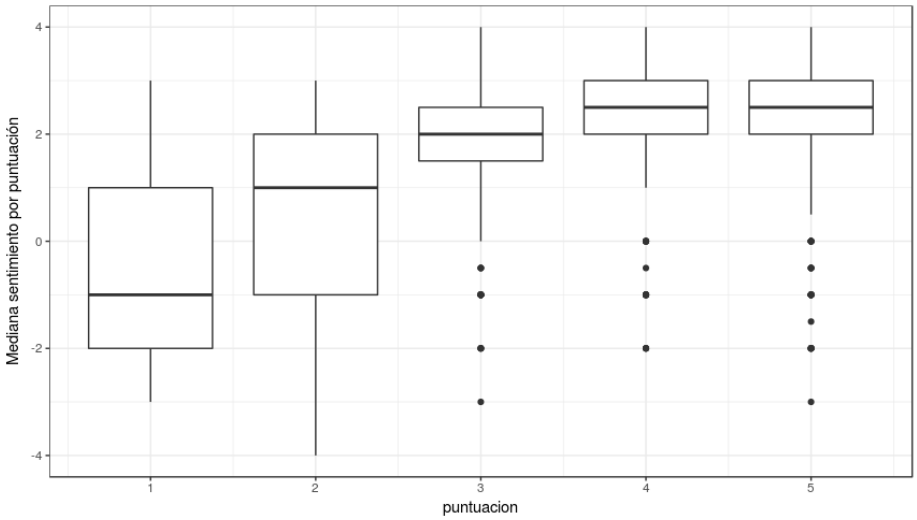
	review.id	puntuacion	media.sentimiento
	<int>	<int>	<dbl>
1	1	5	2.57
2	2	4	2.67
3	3	5	2.50
4	4	5	2.56
5	5	4	1.83
6	6	5	2.20
7	7	1	-0.333

Y el siguiente *boxplot*:



Visualmente podemos diferenciar a lo largo de las puntuaciones, que el sentimiento de los comentarios con 1 estrella es bastante negativo en comparación con los de 5 estrellas de puntuación, que muestran sentimiento positivo. Dicho esto, se añade esta variable al conjunto de datos inicial para utilizarla en los siguientes pasos.

Haremos lo mismo ahora con la mediana, comprobaremos resultados y añadiremos la misma igualmente al *dataset* inicial.



Además, se contará el número de palabras positivas y negativas, para generar nuevas variables que utilizaremos en la última parte de Machine Learning.

Las nuevas variables serán las siguientes: count_afinn_negative, count_afinn_positive, count_bing_negative y count_bing_positive.

Por último, cargaremos los resultados completos en un csv.

5. Modelo

5.1 Preparación de datos

Una vez hecho el análisis exploratorio y creadas las nuevas variables, vamos a proceder a la construcción y evaluación del modelo de predicción.

Para simplificar el análisis, en este caso, agruparemos las puntuaciones de 1 a 5 estrellas en una variable binaria, que se almacenará con el nombre de good.label. A los datos calificados como buenos se les asignará uno (1), agrupando las puntuaciones de 4 y 5 estrellas, mientras que los calificados como malos se les asignará cero (0) agrupando las puntuaciones de 1, 2 y 3 estrellas. Esto nos permitirá utilizar un algoritmo de clasificación y así tener menos categorías desbalanceadas.

También, deberemos tener en cuenta que al estar presente un mayor porcentaje de comentarios positivos que negativos, para separar el conjunto de entrenamiento y de test, tenemos que hacer el particionamiento de forma que se mantenga la proporción. Para ello utilizaremos la función *createDataPartition* de Caret.

Por otro lado, nuestro objetivo es ver la frecuencia de cada palabra, por lo que necesitaremos contar cuantas veces aparece cada una de ellas por comentario. Para realizarlo crearemos una matriz donde los comentarios ocuparán las filas y las palabras las columnas, y cada entrada de la matriz indicará el número de apariciones de cada palabra en cada comentario. Además, para evitar tener en cuenta todas las palabras,

de las cuales muchas no tendrán demasiado valor, solo nos quedaremos con aquellas palabras que aparezcan al menos un 1% (*sparsity* = 1 - 0.1 = 0.99).

Para poder quedarnos con las palabras que aparecen en los comentarios negativos, que son minoría, crearemos las matrices por separado para comentarios positivos y negativos, y después las uniremos. De esta manera conseguiremos las palabras más repetidas, tanto para comentarios calificados como positivos y como negativos.

Por último, una vez que hemos comprobado lo comentado anteriormente, además, se ha tenido en cuenta lo siguiente:
-Se han agregado al *dataset* las variables creadas durante el análisis de sentimiento.
-Los NA se han cambiado a 0.
-Se ha revisado que tanto el conjunto de entrenamiento como el de test tengan el mismo número de columnas.

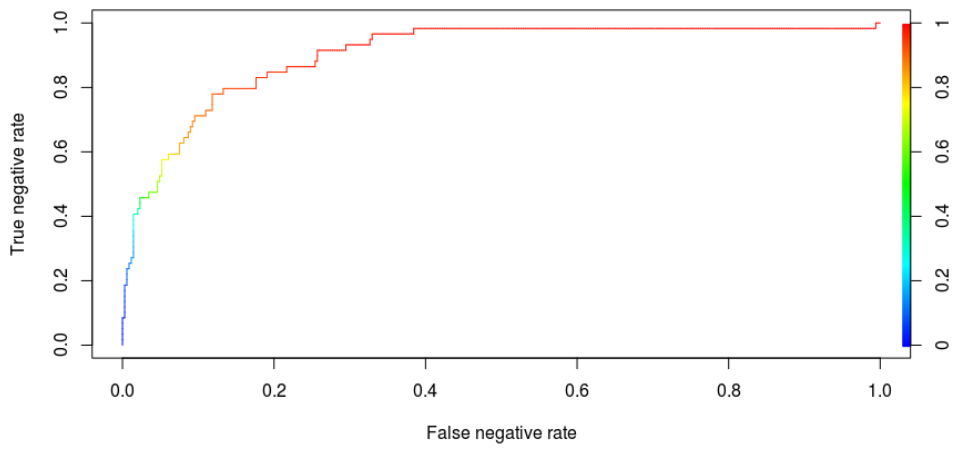
5.2 Construcción del modelo

Con los datos ya preparados, estamos en disposición de comenzar con la parte de construcción y entrenamiento del modelo, cuyo objetivo será clasificar los comentarios como positivos o negativos según la variable *good.label*.

En este caso, se ha utilizado XGboost, una implementación de GBT (*Gradient Boosted Trees*). GBT clasifica mediante el uso de un conjunto de árboles de decisión. Los árboles se construyen secuencialmente, añadiendo en cada iteración el árbol que mejor compense por los errores de los árboles ya existentes, y se le llama método de gradiente porque el modelo evoluciona en dirección al menor error, árbol a árbol. Proporciona alta precisión y velocidad de iteración.

5.3 Evaluación del modelo

El algoritmo XGBoost, nos dará una predicción probabilística, por lo que necesitamos establecer un límite a partir del cual clasificaremos los comentarios como positivos. Para ello dibujaremos la curva ROC como sigue.



Se han obtenido buenos resultados, y puede decirse que el límite a considerar puede ser sobre 0.8.

6. Resultados

6.1 Evaluación de resultados

Según la curva ROC, utilizando un límite de 0.8, podremos clasificar correctamente más de un 50% de los comentarios negativos, mientras clasificamos mal menos del 10% de los comentarios positivos.

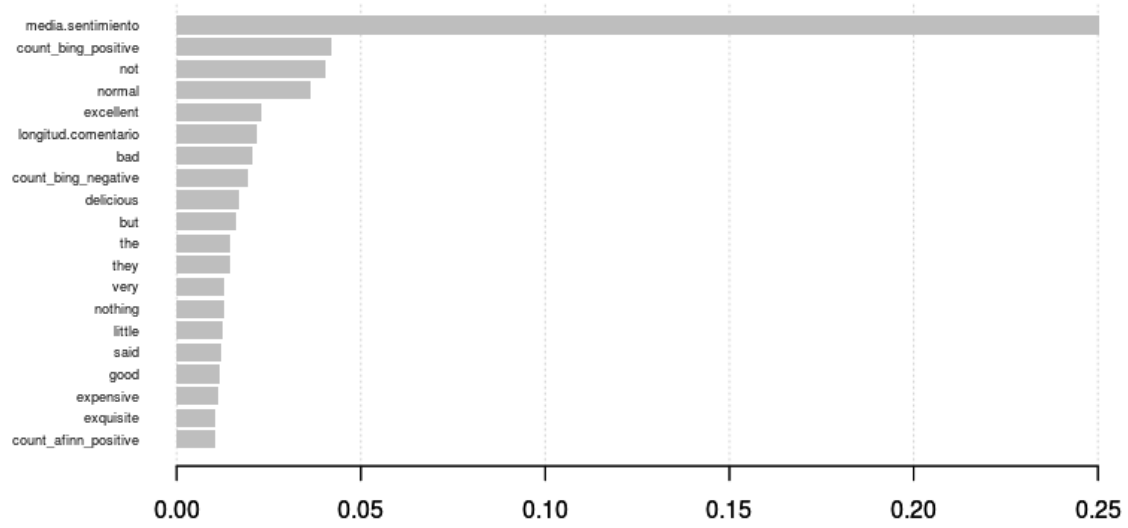
Véase la tabla de predicciones:

	pred	
true	0	1
0	35	24
1	24	332

Y dicho esto, la precisión global que queda es de un 88%.

6.2 Interpretación de las variables

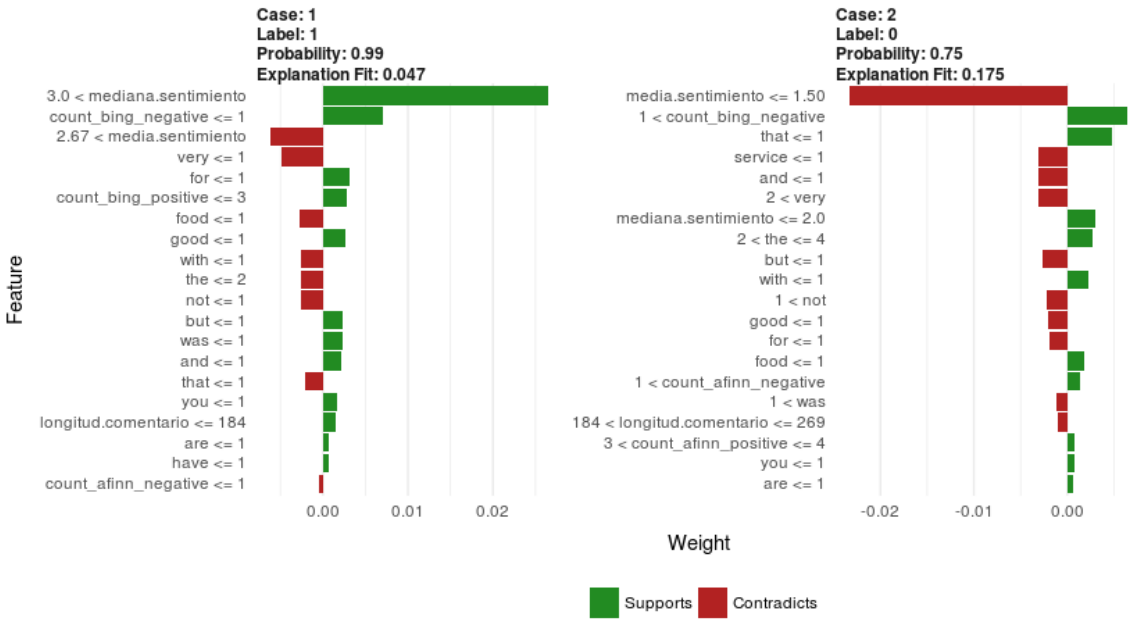
Si queremos mirar en profundidad, una buena tarea puede ser ver la importancia de las variables en el algoritmo XGBoost:



Como podemos ver, en general, encontramos que las palabras más predictivas son las de sentimiento negativo, lo cual es bastante coherente, ya que tienden a destacar para la predicción de comentarios negativos. Además, entre las variables más importantes, tenemos muchas de las que hemos creado durante el análisis de sentimiento.

Por otro lado, si nos centramos en los comentarios negativos y positivos por separado, utilizando el paquete Lime, que se emplea en la interpretación de modelos de caja negra, podremos ver qué características favorecen o contradicen las predicciones.

- Por ejemplo, para 2 comentarios negativos:

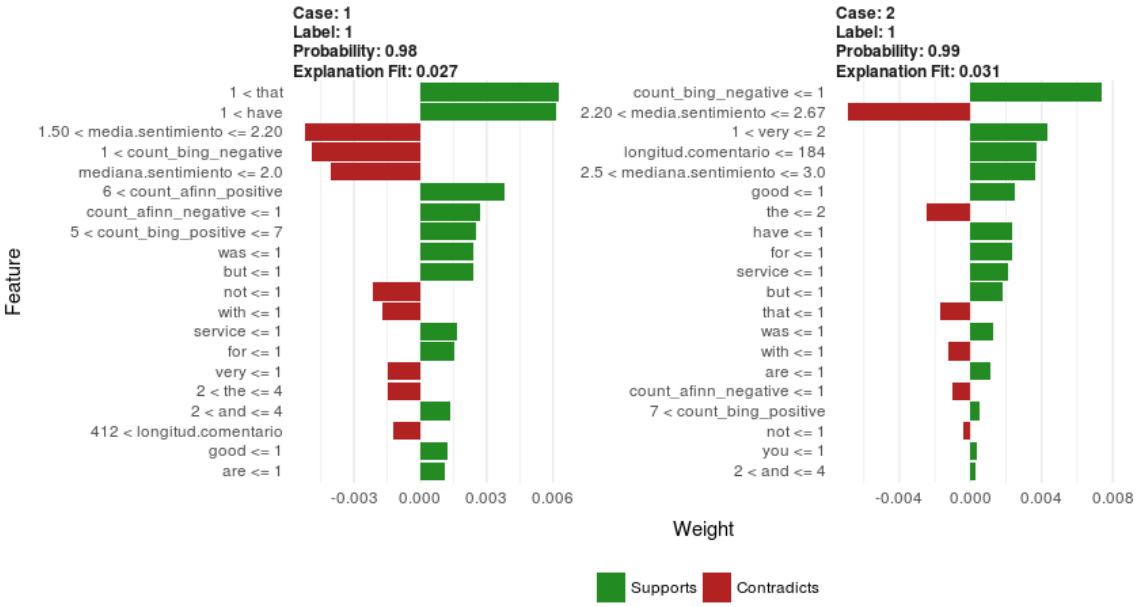


En este tipo de diagramas podemos diferenciar en verde aquellas variables que influyen positivamente en el resultado y en rojo las que no están correlacionadas con lo esperado (lo contradicen).

El caso 1, se trata de una predicción errónea (igual a 1, en lugar de 0). Los factores que han influido en la determinación del resultado son principalmente, la mediana de sentimiento y el número de palabras con sentimiento negativo, que se encuentran en las 2 primeras posiciones en color verde. La mediana de sentimiento por ser más alta de lo normal en comentarios negativos hace que el resultado se desvíe de lo esperado y que la calificación sea finalmente positiva. El número de palabras con sentimiento negativo <=1, hace que se interprete que el texto no tiene muchos matices negativos resaltados. Además, existen otras variables que, aunque en menor medida, han colaborado en la decisión.

En cuanto al caso 2, el resultado de la predicción es correcto. Dos variables que puede decirse que han afectado positivamente en la decisión han sido, el número de palabras con sentimiento negativo > 1 y la mediana de sentimiento <=2

- Para 2 comentarios positivos:



En el caso 1, las dos variables que sobre todo han favorecido al resultado son el número de palabras positivas > 6 y el número de palabras negativas <= 1.

En el caso 2, de nuevo el número de palabras negativas <= 1, la aparición de la palabra *very* en el rango indicado (1<very<=2) y la longitud del comentario <= 184. Lo cual tiene sentido, pues como decíamos en el análisis de sentimiento, los comentarios positivos tienden a ser mas cortos que los negativos.

Además, cabe destacar que la palabra *service* también es influyente en el modelo. Como podemos ver, si se habla del servicio <=1 veces, es favorable para determinar que el comentario sea positivo. Pero si se repite más de una vez, se clasifica como negativo. Por tanto, podemos concluir que uno de los aspectos que más resaltan los clientes como negativo en los restaurantes, es la atención recibida por parte del servicio.

7. Comentarios

En general, los resultados de las predicciones han sido bastante buenos, pero podríamos haber explicado mejor las puntuaciones con mayor cantidad de comentarios extraídos, ya que la cantidad de comentarios negativos es pequeña y es difícil obtener buena predicción de ellos.

También se podría profundizar en los comentarios negativos que se han clasificado mal y ver que variables añadir para lograr cierta mejora en la predicción.

Además, sería de gran utilidad incluir N-gramas (secuencias de palabras como *did not like*), además de las palabras por separado. Con esto conseguiríamos ver el impacto del conjunto, que es imposible de detectar por separado.

Y, por último, he de destacar que la parte con más dificultad ha sido el Web Scraping, puesto que se ha desarrollado para el caso particular de Tripadvisor y para cumplir con el propósito del presente proyecto.

8. Bibliografía

Selenium with Python — Selenium Python Bindings 2 documentation:
http://selenium-python.readthedocs.io/

Package 'RYandexTranslate' - CRAN-R:
https://cran.r-project.org/web/packages/RYandexTranslate/RYandexTranslate.pdf

Text Mining with R:
https://github.com/dgrtwo/tidy-text-mining

Xgboost presentation - CRAN-R:
https://cran.r-project.org/web/packages/xgboost/vignettes/xgboostPresentation.html

Understanding lime - CRAN-R:
https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html

Stack Overflow - Where Developers Learn, Share, & Build Careers:
https://stackoverflow.com/