FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2017/2018

Trabajo de Fin de Máster

TITULO: **Análisis y Predicción en Instagram**

***Alumna*: Marina García Solo de Zaldivar**

***Tutora*: Aida Calviño Martínez**

Junio de 2018

UNIVERSIDAD COMPLUTENSE
MADRID

Agradecimientos

Agradecer a Aida todo el esfuerzo y dedicación que ha puesto en ayudarme como tutora.

A mi familia, por apoyarme siempre. Y a Iván, por su ayuda y paciencia, porque sin él no hubiese sido posible.

Muchas gracias.

Resumen

La memoria que se presenta a continuación, es un proyecto que surge a partir del acelerado crecimiento que está experimentando Instagram. El objetivo base del trabajo es poder predecir qué porcentaje de seguidores de un usuario podrían dar *Likes* a una foto en Instagram, partiendo del análisis de diferentes factores relacionados con la publicación, así como la descomposición de los píxeles de la imagen.

Para el desarrollo del estudio se han recogido las 25 últimas publicaciones que las 500 cuentas con más seguidores del mundo tenían en Diciembre de 2017. Base a partir de la cual se elaborarán una serie de modelos predictivos, con la intención de encontrar el modelo que mejor explique nuestra variable objetivo.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo se va a hablar tanto del concepto y origen de Instagram, como de la relevancia del tema en la actualidad.

1.1. Concepto y origen

Instagram [1] es una red social en la que los usuarios pueden subir fotos y vídeos, a los que pueden aplicarles efectos fotográficos a través de la misma página; además de añadirles un título o texto en el que se suele escribir algo en relación a la imagen. La idea principal de esta red, al igual que el resto de redes, es la interactuación mediante comentarios y *Likes* en las publicaciones, generando un *feedback* entre usuarios.

Fue fundada por Kevin Systrom y Mike Krieger[1] en octubre de 2010. Esta página y aplicación, que cuenta con más de 800 millones de usuarios activos, fue creada con la idea de captar momentos en imágenes con un formato diferente. Es decir, la característica distintiva de la aplicación es que da una forma cuadrada a las fotografías en honor a la Kodak Instamatic y las cámaras Polaroid, contrastando con la relación de aspecto 16:9 y 4:3 que actualmente usan la mayoría de las cámaras de teléfonos móviles.

[1] Instagram about us: `https://www.instagram.com/about/us/`

1.2. Contexto

En sólo 8 años Instagram ha pasado a ser una aplicación con la que miles de personas no podrían pasar sin utilizar en su vida diaria. Y en la que las *celebrities* han cogido el relevo, utilizándola como foco importante para atraer a sus fans.

Según el Blog Marketingdirecto.com[2], "... aunque Instagram no cuente con las mismas cifras de usuarios que las reportadas por Facebook y YouTube, esta red social es la que mayor poder de influencia tiene. Hecho que no resulta extraño si tenemos en cuenta que se posiciona como el canal más efectivo a la hora de influir en las decisiones de compra e impulsar las ventas de las marcas."

De hecho, el 70 % de las principales marcas ya utiliza, como parte de sus campañas de marketing, estrategias con el objetivo de alcanzar audiencias mayores y maximizar sus ingresos a través de Instagram. Aquí es donde toman relevancia los llamados Influencers [2]; estos profesionales cuentan con audiencias (de tamaño variable) fidelizadas que confían en su palabra. Tanto, que el 72 % de los usuarios de Instagram afirma tomar decisiones de compras basadas en recomendaciones que se han encontrado en esta red social.

Por tanto, si una marca pudiese llegar a conocer cuántos usuarios se sentirían atraídos por una publicación en Intsagram podría utilizarlo para sacarle un mayor rendimiento a su estrategia comercial, así como saber hacia dónde enfocarla.

[2]Marketingdirecto.com webpage: `https://www.marketingdirecto.com/`

Capítulo 2

Fuente de los datos

En este capítulo se explica todo lo relacionado al origen y formato de los datos de los que parte el estudio.

2.1. Origen de los datos

Para este estudio se han recopilado las 25 últimas publicaciones que tenían las 500 cuentas con más seguidores del mundo, en Diciembre de 2017. En total, 12.500 observaciones.

Los datos se solicitaron a la página Picodash[1]. Esta web es, principalmente, un motor de búsqueda avanzado de Instagram, que funciona como herramienta de administración de redes sociales para ayudar a marcas, publicistas etc, a buscar y analizar el contenido de Instagram. Picodash permite buscar publicaciones de Instagram por rango de fecha / hora, ubicación, *Hashtags*, lugares o usuarios. También permite analizar los seguidores, visualizar publicaciones en el mapa y obtener un mapa de calor de la actividad de Instagram.

Al tratarse de un trabajo de fin de máster, la página proporcionó los datos de forma gratuita. El conjunto de datos estaba formado por 500 archivos en formato csv. Cada archivo tenía el nombre del usuario correspondiente a la cuenta.

[1]Picodash webpage: `https://www.picodash.com/`

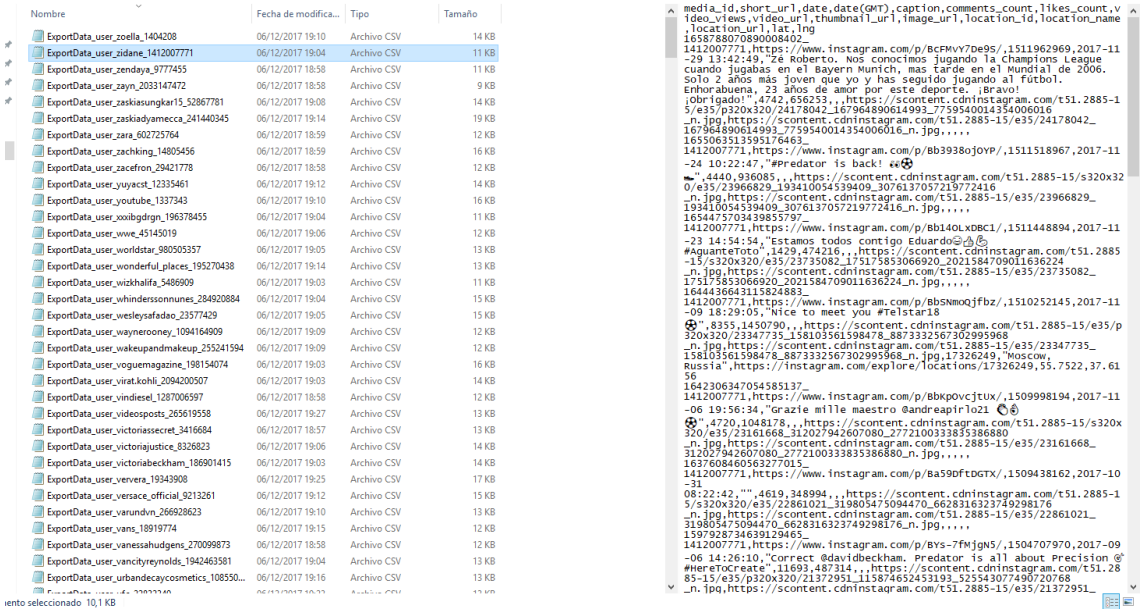

Figura 2.1: Archivos originales de cada cuenta de Instagram.

Una vez conseguidos los datos, se unieron en una misma tabla a través del Código A.1.1 de R. Este bucle consiste en:

- Leer cada archivo de la carpeta donde se encuentran.
- Coger el nombre del archivo correspondiente e introducirlo en una variable llamada Usuario.
- Unir todos los registros de cada archivo en una misma tabla llamada Total.

Como el análisis a realizar se centra en las imágenes, tras el proceso de unión se eliminaron todos los datos referentes a las publicaciones de vídeos, para proceder en segundo lugar a extraer y transformar las fotos de cada publicación.

2.2. Extracción y transformación de imágenes

Uno de los factores principales a la hora de realizar el estudio es el análisis de las imágenes. Es decir, comprobar si las características de la imagen, como el color o la luz,

influyen en la variable objetivo. Para ello, es necesario descomponer cada imagen en píxeles y leer la información de cada uno de ellos.

La forma de descomponer las imágenes está basada en el Modelo RGB (RED - GREEN - BLUE). Este espacio de color es el formado por los colores primarios (Rojo, Verde y Azul) y es el sistema más adecuado para representar imágenes que serán mostradas en monitores. Las imágenes RGB utilizan tres colores para reproducir en pantalla hasta 16,7 millones de colores. RGB [3] es el modo que, por lo general, viene en nuestras cámaras de fotos.

Previo al análisis, se efectuó la descarga de imágenes y descomposición de cada píxel en RGB, por lo que la información de cada píxel se guardaría en 3 columnas, una para RED, otra para GREEN y otra para BLUE. Al ser un número tan elevado de columnas, se decidió reducir la imagen a 10x10 píxeles, obteniendo como resultado 300 columnas (100 por cada color).

Aunque la base de datos original venía provista de una variable con la URL de la imagen, esos links se encontraban deshabilitados. Por tanto, antes de crear un código para la descarga de imágenes, se ve necesario crear un bucle a través del Código A.1.2 de R que haga lo siguiente:

- Leer el link de la publicación original de Instagram, dentro de la variable URL_PUBLICACION.
- Recorrer todo el código HyperText Markup Language (HTML) e identificar la URL de la imagen, buscando en el código la línea comprendida entre “content=” y “.jpg”.
- Guardar el link de la imagen en una nueva variable llamada URL_IMAGEN.

A continuación, se procede a crear otro bucle (A.1.3)que consiste en:

- Leer cada URL de cada imagen y acceder a ella a través de internet.
- Descargar la imagen en una carpeta determinada.
- Reducir la imagen a tamaño 10x10 px.
- Descomponer la información de los píxeles en RGB.

- Transformar la información de cada píxel en datos e introducirlas en 3 columnas, por colores RGB.

Además de las observaciones que se descartaron por ser vídeos, se eliminaron otro tanto de observaciones por errores al descargarlas. Por tanto, finalmente se obtiene la tabla con la que se va a trabajar, formada por 8.581 observaciones y 14 variables.

Capítulo 3

Objetivos y Metodología

En el siguiente capítulo se van a presentar los objetivos que nos han llevado a realizar el estudio, así como la metodología utilizada para alcanzarlos.

3.1. Objetivo Principal

El objetivo principal del estudio es la predicción del porcentaje de *likes* (Número de *likes*/Número de seguidores) que tendrá una foto de una cuenta de Instagram.

3.2. Objetivos Secundarios

A lo largo del análisis van surgiendo otros objetivos secundarios que se deben ir cumpliendo para poder alcanzar el objetivo final.

1. Conocer las variables y extraer información útil a partir de la información inicial de los datos.

2. Reducir la dimensionalidad de la gran cantidad de variables creadas a raíz de los pixeles de la imagen.

3. Eliminar las variables que no aporten información necesaria al modelo predictivo.

4. Realizar una comparativa de los diferentes modelos predictivos de forma que sea posible identificar el modelo que mejor se adapte a nuestros datos.

3.3. Metodología SEMMA

En cuanto a la forma de desarrollar el tratamiento y análisis de los datos, se llevará a cabo la metodología SEMMA [4]. Esta metodología, definida por SAS Institute, es el proceso de selección, exploración y modelado de grandes cantidades de datos para descubrir patrones desconocidos. El nombre de esta terminología es el acrónimo correspondiente a las cinco fases básicas del proceso.

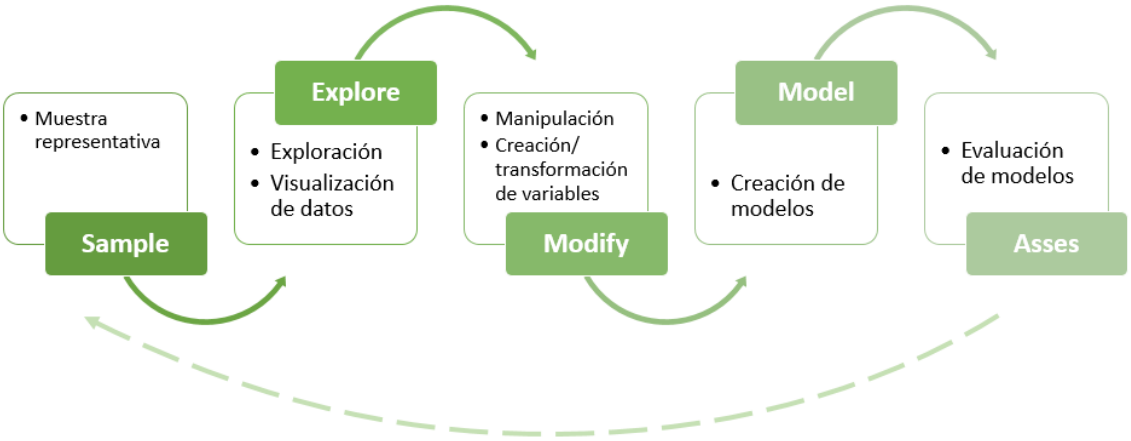

Figura 3.1: Esquema de fases de la metodología SEMMA

El proceso definido se inicia con la extracción de una muestra representativa sobre la que se va a aplicar el análisis. La forma más común de obtener una muestra es la selección al azar mediante el muestreo aleatorio simple, donde cada uno de los individuos de una población tiene la misma posibilidad de ser elegido. Una vez determinada una muestra, se debe proceder a una exploración de la información disponible con el fin de simplificar el problema y optimizar la eficiencia del modelo. Para lograr este objetivo se propone la utilización de herramientas de visualización o estadísticas que ayuden a poner de manifiesto relaciones entre variables. De esta forma se pretende determinar cuáles son

las variables explicativas que van a servir como entradas al modelo. La tercera fase consiste en la manipulación de los datos, de forma que se definan y tengan el formato adecuado los datos que serán introducidos en el modelo.

Una vez que se han definido, se procede al análisis y modelado de los datos. El objetivo de esta fase consiste en establecer una relación entre las variables explicativas y las variables objeto del estudio, que posibiliten inferir el valor de las mismas con un nivel de confianza determinado. Las técnicas utilizadas para el modelado de los datos incluyen métodos estadísticos tradicionales (tales como análisis discriminante, métodos de agrupamiento, y análisis de regresión), así como técnicas basadas en datos tales como redes neuronales, técnicas adaptativas, árboles de decisión.... Finalmente, la última fase del proceso consiste en la valoración de los resultados mediante el análisis de bondad del modelo o modelos, contrastado con otros métodos estadísticos o con nuevas poblaciones muestrales. Cabe decir que no siempre intervienen todas las fases del proceso, las fases pueden repetirse y el orden de las mismas puede modificarse. En este análisis en concreto no se va a realizar la fase de muestreo.

Una vez entendido el procedimiento de la metodología SEMMA, se procede a explicar cada uno de los modelos que se utilizarán en la fase de Modelado de datos del análisis.

3.3.1. Regresión Lineal

El análisis de regresión lineal [5] es una técnica estadística utilizada para estudiar la relación entre variables. Los métodos de regresión estudian la construcción de modelos para explicar o representar la dependencia entre una variable dependiente y las variables explicativas o independientes. Este modelo tiene lugar cuando la dependencia es de tipo lineal, y pretende dar respuesta cuestiones como ¿Es significativo el efecto que una variable X causa sobre otra Y? ¿Es significativa la dependencia lineal entre esas dos variables? De ser así, con este modelo se podría explicar y predecir la variable dependiente (Y) a partir de valores observados en las independientes (X).

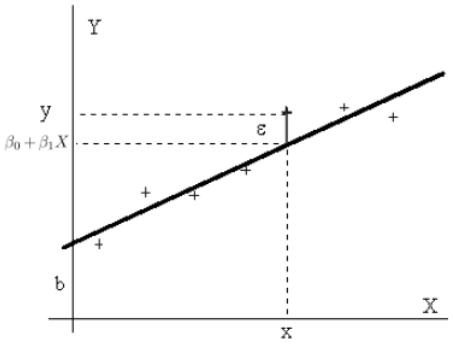

Figura 3.2: Representación de Regresión Lineal

La estructura del modelo de regresión lineal es la siguiente: $Y = \beta_0 + \beta_1 X + \varepsilon$. En esta expresión estamos admitiendo que todos los factores que influyen en la variable respuesta Y pueden dividirse en dos grupos: el primero contiene una variable explicativa X y el segundo incluye un conjunto amplio de factores no controlados que se engloban bajo el nombre de perturbación o error aleatorio, ε , y que provoca que la dependencia entre las variables dependiente e independiente no sea perfecta, sino que esté sujeta a incertidumbre.

Lo que sería deseable en un modelo de regresión es que los errores aleatorios fuesen de media cero para cualquier valor de X, es decir, $E[\varepsilon/X = x] = E[\varepsilon] = 0$, y por lo tanto:

$$E[Y/X = x] = \beta_0 + \beta_1 x + E[\varepsilon/X = x] = \beta_0 + \beta 1x$$

En dicha expresión se observa que:

- La media de Y, para un valor fijo x, varía linealmente con x.
- Para un valor x se predice un valor en Y dado por $y = E[Y/X = x] = \beta_0 + \beta_1 x$, por lo que el modelo de predicción puede expresarse también como $Y = \beta_0 + \beta_1 X$.
- El parámetro β_0 es la ordenada al origen del modelo y β_1 la pendiente, que puede interpretarse como el incremento de la variable dependiente por cada incremento en una unidad de la variable independiente. Estos parámetros son desconocidos y hay que estimarlos de cara a realizar predicciones.

Además de la hipótesis establecida sobre los errores de que la media ha de ser cero, se establecen las siguientes hipótesis:

- La varianza de ε es constante para cualquier valor de x.
- La distribución de ε es normal, de media 0 y desviación σ.
- Los errores asociados a los valores de Y son independientes unos de otros. En consecuencia, la distribución de Y para x fijo es normal, con varianza constante σ^2, y media que varía linealmente con x, dada por $\beta_0 + \beta_1 x$. Además los valores de Y son independientes entre sí.

3.3.2. Redes Neuronales

Los modelos de Redes Neuronales artificiales [6] podrían definirse como redes interconectadas masivamente en paralelo de elementos simples (normalmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Debido a su constitución y a sus fundamentos, las redes neuronales presentan un gran número de características semejantes a las del cerebro. Algunas de éstas son, por ejemplo, que sean capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc.

Principales ventajas

Entre las ventajas de este método estadístico encontramos:

1. Aprendizaje Adaptativo:

   Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial. Como las redes pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni necesidad de especificar funciones de distribución de probabilidad. Las redes neuronales son

sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema, haciéndolas capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje.

2. Auto-organización:

   Puede crear su propia organización o representación de la información que recibe mediante la etapa de aprendizaje. Las redes neuronales emplean esa capacidad de aprendizaje adaptativo para autoorganizar la información que reciben durante el aprendizaje, lo que consiste en la modificación de la red completa para llevar a cabo un objetivo específico. Esta autoorganización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a las que no había sido expuesta anteriormente. Esta característica es muy importante cuando se tiene que solucionar problemas en los cuales la información de entrada no es muy clara; además permite que el sistema dé una solución, incluso cuando la información de entrada está especificada de forma incompleta.

3. Tolerancia a fallos:

   En la red neuronal de un cerebro, si se produce un fallo en un número no muy grande de neuronas y aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina. Partiendo de ésto, hay dos aspectos distintos respecto a la tolerancia a fallos:

   - Las redes pueden aprender a reconocer patrones distorsionados o incompletos. Esta es una tolerancia a fallos respecto a los datos.
   - También pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

La razón por la que las redes neuronales son tolerantes a los fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. Almacenan información no localizada. Por tanto, la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.

Elementos Básicos

Una red [6] está constituida por neuronas interconectadas y distribuidas en tres capas, aunque esto último puede variar. Los datos ingresan por medio de la "capa de entrada", pasan a través de la "capa oculta" (que puede estar constituida, a su vez, por varias capas) y salen por la "capa de salida".

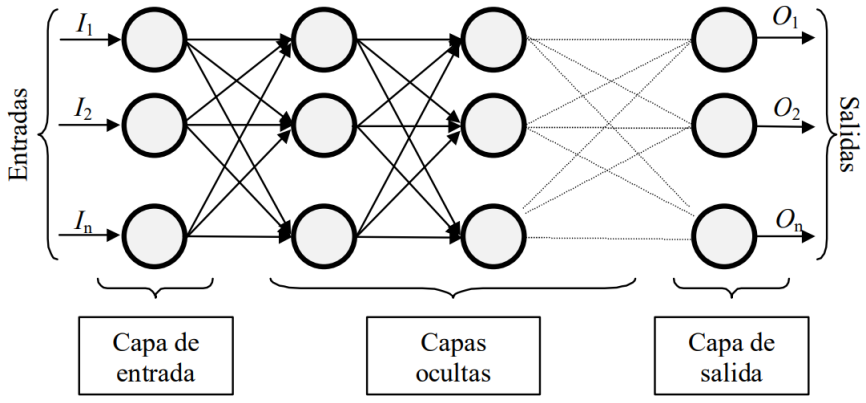

Figura 3.3: Representación de una Red Neuronal

- **Capa de entrada:** Es la capa que recibe directamente la información proveniente de las fuentes externas de la red.

- **Capas Ocultas:** Son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.

- **Capa de salida:** Transfieren información de la red hacia el exterior.

En la Figura 3.3 se puede ver el ejemplo de la estructura de una posible red multicapa, donde cada nodo o neurona únicamente está conectada con neuronas de un nivel superior. Una red es totalmente conectada si todas las salidas desde un nivel llegan a todos y cada uno de los nodos del nivel siguiente.

3.3.3. Random Forest

Random Forest [7] es uno de los algoritmos con mejores comportamientos en los problemas de aprendizaje, en particular en aquellos que cuentan con una cantidad importante de variables explicativas. Se trata de una técnica estadística basada en algoritmos donde se sortean M muestras del conjunto de datos originales sobre las que se construyen M árboles para los cuales, en cada nodo, se elige la mejor subdivisión hecha por un subconjunto de variables explicativas; buscando siempre la variable y el valor umbral de la misma que maximicen la homogeneidad de las particiones resultantes.

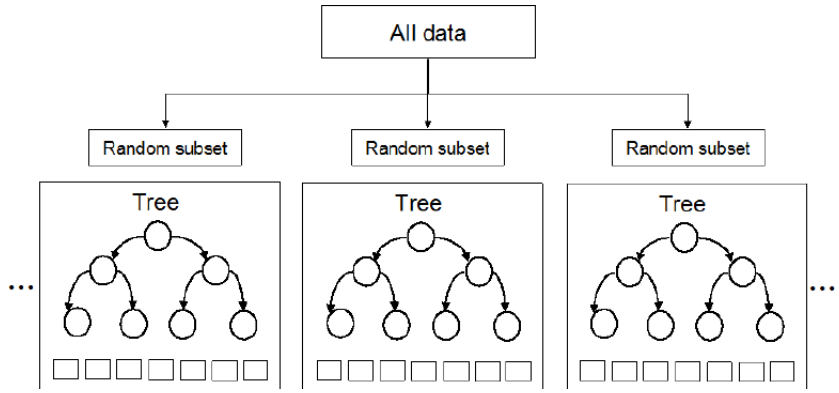

Figura 3.4: Representación de un árbol

Sintetizando el proceso, éste se basaría (para datos de tamaño N) en:

1. Repetir m veces las 3 fases:

   - Seleccionar N observaciones con reemplazamiento de los datos originales.

- Aplicar un árbol de la siguiente manera: En cada nodo, seleccionar p variables de las k originales y de las p elegidas, escoger la mejor variable para la partición del nodo.
- Obtener predicciones para todas las observaciones originales N.

2. Promediar las m predicciones obtenidas en el primer paso.

Teniendo ésto en cuenta, los principales parámetros a controlar serían:

- El tamaño de las muestras n y si se va a utilizar, o no, un reemplazamiento.
- El número de iteraciones m a promediar.
- El número de variables p a muestrear en cada nodo.
- El número de hojas final o la profundidad del árbol.
- El *Maxbranch* (número de divisiones máxima en cada nodo).
- El *p-valor* para las divisiones en cada nodo. Cuanto más alto es el *p-valor*, los árboles serán menos complejos; es decir, más sesgo, menos varianza.
- El número de observaciones mínimo en una rama-nodo. Se puede ampliar para evitar sobreajuste (reducir la varianza) o reducir para ajustar mejor (reducir el sesgo).

Las ventajas del Random Forest son:

- Es uno de los algoritmos de aprendizaje más certeros que hay disponibles.
- Rapidez de ejecución.
- Puede manejar cientos de variables de entrada sin excluir ninguna.
- Da estimaciones de qué variables son importantes en la clasificación.
- Computa las proximidades entre los pares de casos que pueden usarse en los grupos, ofreciendo así una localización de valores atípicos, y produciendo vistas interesantes de los datos.

- Ofrece un método experimental para detectar las interacciones de las variables.

Las desventajas del Random Forest son:

- La clasificación hecha por Random Forest es difícil de interpretar.
- Para los datos que incluyen variables categóricas con diferente número de niveles, el Random Forest puede parcializarse a favor de esos atributos con más niveles.
- Si los datos contienen grupos de atributos correlacionados con similar relevancia para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes.

3.3.4. Gradient Boosting

Boosting [8] es un enfoque de Machine Learning basado en la idea de crear una regla de predicción altamente precisa combinando muchas reglas relativamente débiles e imprecisas. El objetivo de Boosting es el de mejorar el rendimiento del algoritmo de aprendizaje al tratarlo como una "caja negra"que se puede llamar repetidamente, como una subrutina. La idea fundamental es elegir conjuntos de entrenamiento para el algoritmo de aprendizaje base, de tal manera que lo obligue a inferir algo nuevo sobre los datos cada vez que se lo llame.

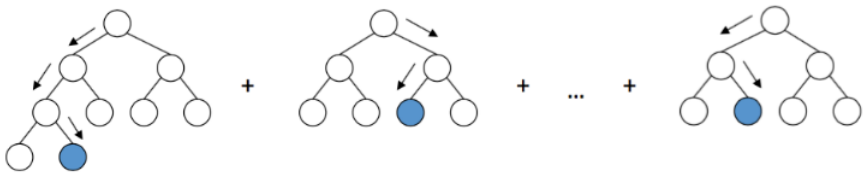

Figura 3.5: Representación de la idea inicial de Gradient Boosting

Es decir, este algoritmo consiste en repetir la construcción de árboles de regresión / clasificación, modificando ligeramente las predicciones iniciales cada vez, intentando ir minimizando los residuos en la dirección de decrecimiento. Al plantear diferentes árboles

de forma reiterada, el proceso va ajustando las predicciones cada vez más a los datos, y de alguna manera unos árboles corrigen a otros con lo que la flexibilidad y adaptación del método mejora respecto a la construcción de un único árbol.

Ventajas Principales

1. Invariante frente a transformaciones monótonas: no es necesario realizar transformaciones logarítmicas, etc.

2. Buen tratamiento de missing, variables categóricas, etc. Universalidad.

3. Muy fácil de implementar, relativamente pocos parámetros a monitorizar (número de hojas o profundidad del árbol, tamaño final de hojas, parámetro de regularización...).

4. Gran eficacia predictiva, algoritmo muy competitivo, superando normalmente al algoritmo Random Forest.

5. Robusto respecto a variables irrelevantes y a la colinealidad, detectando interacciones ocultas.

Desventajas Principales

En datos relativamente sencillos (pocas variables, sin missing, sin interacciones, linealidad (regresión) o separabilidad lineal (clasificación)), el gradient boosting (o random forest) pueden ser preferibles modelos sencillos (regresión, regresión logística, discriminante) o modelos ad-hoc que adapten aspectos concretos como la no linealidad.

3.3.5. Ensemble de Modelos

Los métodos Ensemble [9] consisten en la construcción de predicciones a partir de la combinación de varios modelos. Es decir, utilizan varios algoritmos de aprendizaje para obtener un rendimiento predictivo que mejore el que podría obtenerse por medio de cualquiera de los algoritmos de aprendizaje individuales que lo constituyen.

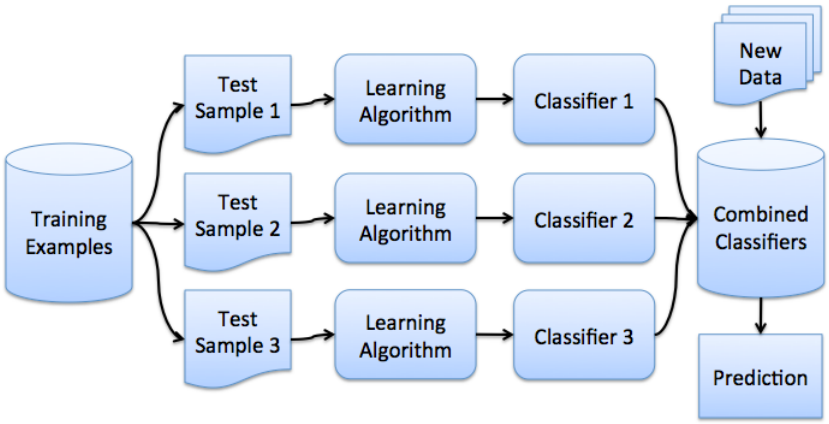

Figura 3.6: Representación de un ensamblado

Aunque los métodos de ensamblado más comunes son Bagging y Boosting, en el presente estudio se va a realizar la comparación mediante Stacking [10]. El cuál se trata de una forma de combinar varios modelos introduciendo el concepto de la meta de aprendizaje entre modelos. Es decir, dadas unas predicciones obtenidas por diferentes algoritmos, se combinan sus resultados.

Entre sus principales ventajas podríamos destacar que este tipo de modelado es bastante robusto, debido a que unos modelos se corrigen a otros; y que reducen la varianza del error en general.

Por otro lado, hay algunas desventajas [11] que presentan este tipo de metodología debido a su complejidad derivada de que cada modelo tiene sus errores de estimadores de parámetros, además de que los resultados no son interpretables.

Capítulo 4

Descripción de variables

Además del origen de los datos, es preciso conocer cuál es exactamente la estructura de éstos, por lo que se muestra a continuación una breve descripción de las variables con las que se llevó acabo el estudio.

4.1. Variables iniciales

En la presente sección se detallan las variables que contiene el archivo inicial del que parte el estudio.

Nombre variables	**Descripción**	**Nivel**
Caption	Texto que acompaña a la publicación	Nominal
Comments_Count	Número de comentarios	Intervalo
Date	Fecha de la publicación en formato número	Nominal
Date(GMT)	Fecha de la publicación en formato Fecha y Hora	Nominal
Image_Url	URL de la imagen	Nominal
Lat	Coordenadas Latitud	Nominal
Likes_Count	Número de Likes	Intervalo
Lng	Coordenadas Longitud	Nominal
Location_Name	Nombre de la localización (en caso de tener Ubicación)	Nominal
Location_Url	URL de la localización	Nominal
Media_Id	Identificativo de la publicación	Nominal
Short_Url	URL corto de la publicación	Nominal
Video_Url	URL del video (en caso de vídeo)	Nominal
Video_Views	Número de visualizaciones (en caso de vídeo)	Intervalo

Tabla 4.1: Lista de variables iniciales

4.2. Generación y transformación de variables

En esta sección veremos todas las variables que se añadieron a los datos con el fin de enriquecer el análisis.

Una parte de la información extra que se añadió, fue obtenida de la página Social-Blade[1]. Esta web es una plataforma de estadísticas que permite hacer un seguimiento, midiendo la evolución de las cuentas públicas de usuarios en múltiples redes sociales como YouTube, Twitter e Instagram. Esta información es relativa a los Usuarios de las cuentas de Instagram.

TOP 500 INSTAGRAM PROFILES - SORTED BY MOST FOLLOWED

RANK	GRADE	USERNAME	MEDIA	•FOLLOWERS•	FOLLOWING
1	A++	instagram	5,045	231,654,522	196
2	A++	selenagomez	1,406	133,066,764	37
3	A++	cristiano	2,163	119,979,887	388
4	A++	arianagrande	3,376	117,336,659	1,341
5	A++	beyonce	1,580	110,605,163	0
6	A++	kimkardashian	4,110	106,849,795	117
7	A++	taylorswift	101	106,226,121	0
8	A++	kyliejenner	5,345	101,021,835	125
9	A++	therock	3,333	99,331,902	197
10	A++	justinbieber	4,287	96,463,010	77
11	A++	neymarjr	4,071	88,275,035	715
12	A++	kendalljenner	2,876	86,698,965	186
13	A++	leomessi	285	86,628,991	198
14	A++	nickiminaj	5,190	85,577,789	1,028
15	A++	natgeo	16,177	85,282,416	121
16	A++	nike	907	75,819,651	136
17	A++	mileycyrus	6,704	73,987,830	568
18	A++	jlo	2,200	72,125,388	1,059
19	A++	khloekardashian	3,233	72,050,139	142
20	A++	katyperry	878	68,404,203	338
21	A+	ddlovato	1,981	65,318,192	358
22	A+	kourtneykardash	3,433	60,380,064	76
23	A+	badgalriri	4,212	59,613,918	1,296
24	A+	victoriassecret	5,715	58,638,430	507
25	A+	kevinhart4real	5,025	56,591,645	440

Figura 4.1: Ranking 25 primeras publicaciones en SocialBlade

[1]SocialBlade webpage: `https://socialblade.com/`

Las variables obtenidas son las que se muestran en la Tabla 4.2:

Nombre variables	**Descripción**	**Nivel**
Rank	El puesto en la lista de cuentas con más seguidores	Nominal
Media	Número de publicaciones que tiene la cuenta	Intervalo
Followers	Número de seguidores	Intervalo
Following	Número de cuentas a las que sigue el usuario	Intervalo

Tabla 4.2: Lista de variables extraídas de SocialBlade

Por otro lado, se añaden una serie de variables adicionales a raíz de las variables ya existentes que se muestran en la Tabla 4.3.

Nombre variables	**Descripción**	**Nivel**	**Proceso**	**Rol**
ByN	Indica si la imagen está en blanco y negro	Binomial	SAS	Input
Dia	Día de la semana (codificado de 1 a 7)	Intervalo	SAS	Input
Dia_Sem	Día de la semana	Nominal	SAS	Input
Dif	Diferencia entre esa publicación y la anterior	Intervalo	SAS	Input
Etiquetas	Número de @ del *Caption*	Intervalo	SAS	Input
FS	Indica si se publicó en Fin de Semana	Binomial	SAS	Input
Hashtag	Número de # del *Caption*	Intervalo	SAS	Input
Hora	Hora de la publicación	Nominal	SAS	Input
Idioma	Indica si el *Caption* está en Inglés o no	Binomial	Manual	Input
Letra	Número de letras que tiene el *Caption*	Intervalo	SAS	Input
Mes	Mes de la publicación	Nominal	SAS	Input
Palabra	Número de palabras que tiene el *Caption*	Intervalo	SAS	Input
Porlike	% de followers que dieron Like a la publicación	Intervalo	SAS	Objetivo
Sector	Sector del Usuario	Nominal	Manual	Input
Repost	Indica si la publicación es un Repost de otra	Binomial	SAS	Input
Ubi	Indica si la publicación tiene Ubicación	Binomial	SAS	Input
Cap	Indica si la publicación tiene *Caption*	Binomial	SAS	Input

Tabla 4.3: Lista de variables adicionales

4.2.1. Variable Objetivo

Aunque la idea inicial es predecir el número de *likes* qué tendrá una foto en Instagram, como partimos con datos de cuentas con información diferente las unas de las otras, se decidió crear una variable objetivo que se adaptase a cada una. Para ello se calculó el porcentaje de seguidores que han reaccionado con un *like* a la publicación, siendo la variable *Porlike* = (Número de *likes*/Número de seguidores) * 100 que tendrá una foto de la cuenta. De esta forma, la variable objetivo no se verá tan influenciada por el número bruto de seguidores de la cuenta.

En la Figura 4.2 se muestra el histograma de la variable objetivo *PORLIKE*, donde se puede observar cómo la mayoría de las observaciones están en un porcentaje de *likes* entre el 0 y el 11 %.

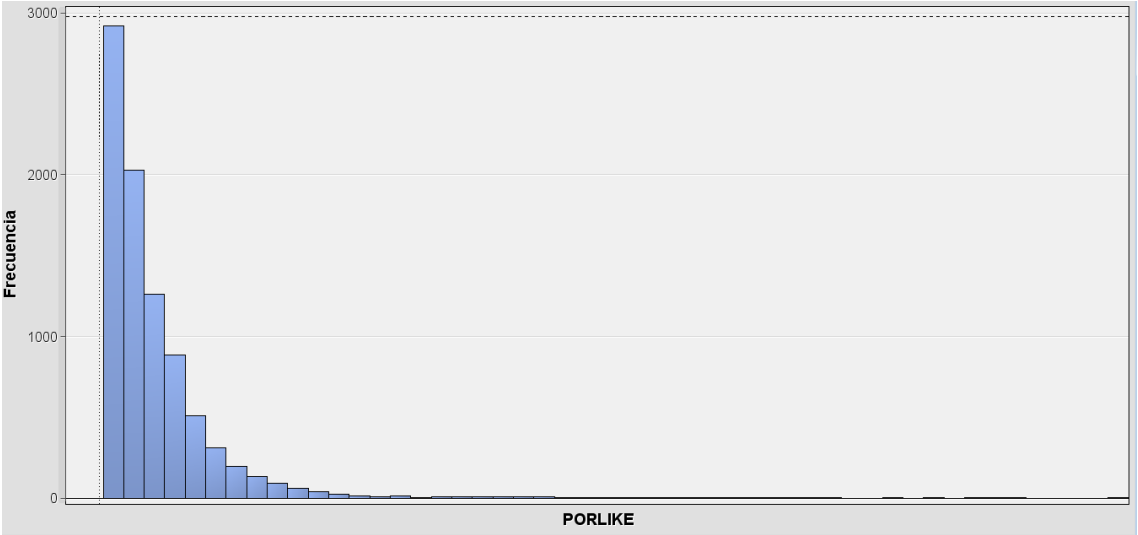

Figura 4.2: Histograma de la variable objetivo.

4.2.2. Hora de la publicación

En el caso de la variable "Hora de la publicación", se decidió agrupar las horas en varios tramos, de forma que pasase a ser una variable categórica con 7 distintos momentos del día:

- Mañana (7 a.m – 10 a.m)
- Media mañana (11 a.m – 12 a.m)
- Medio día (1 p.m – 2 p.m)
- Tarde (3 p.m – 7 p.m)
- Noche (8 p.m – 10 p.m)
- Media noche (11 p.m – 1 a.m)

- Madrugada (2 a.m – 6 a.m)

En base a esta clasificación se muestra en la Figura 4.3 la media de la variable objetivo según la hora de publicación, se muestra en media ya que al ser un porcentaje la suma carece de sentido. La mañana es la hora que más porcentaje de *likes* tiene siendo este porcentaje del 3,19 %.

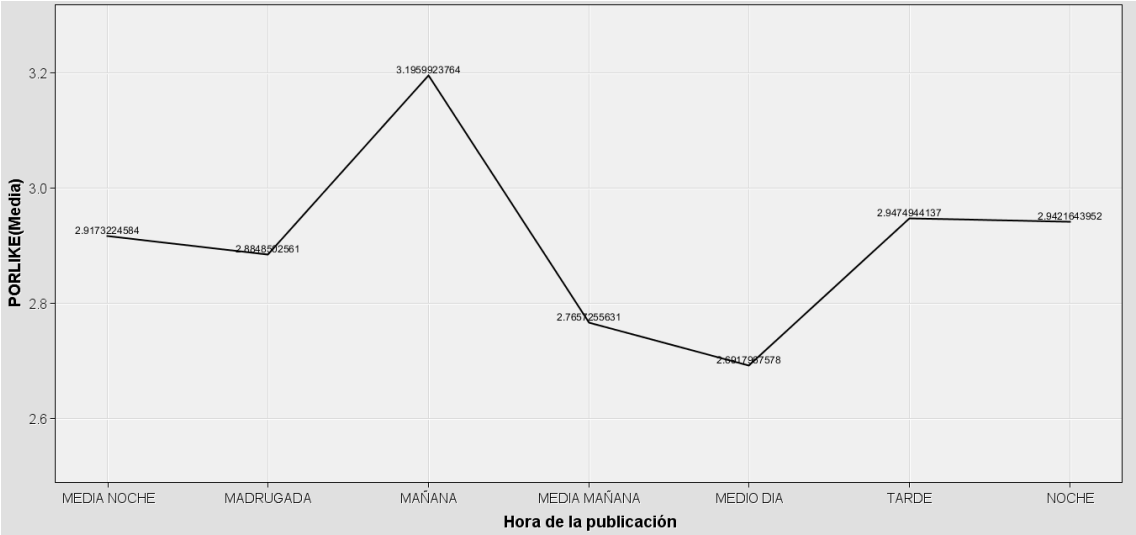

Figura 4.3: Variable hora de publicación frente a la media de la variable objetivo.

4.2.3. Diferencia entre publicaciones

Esta variable se genera cogiendo la fecha de la publicación más actual y restándole la anterior, así se consigue la diferencia en minutos entre dos publicaciones. Creemos que el tiempo que transcurre entre dos publicaciones es importante ya que publicar con muy poco tiempo de intervalo podría saturar a los *followers* y con tiempos entre publicaciones muy amplios podría descender la influencia que se crea en la red social.

Esta variable frente a la objetivo se muestra en la Figura 4.4 donde se puede observar un gráfico de dispersión que nos indica que tiempos muy largos entre publicaciones provocan porcentajes de *likes* más bajos, mientras que los porcentajes más altos de *likes* se concentran en los tiempos de publicación más cortos.

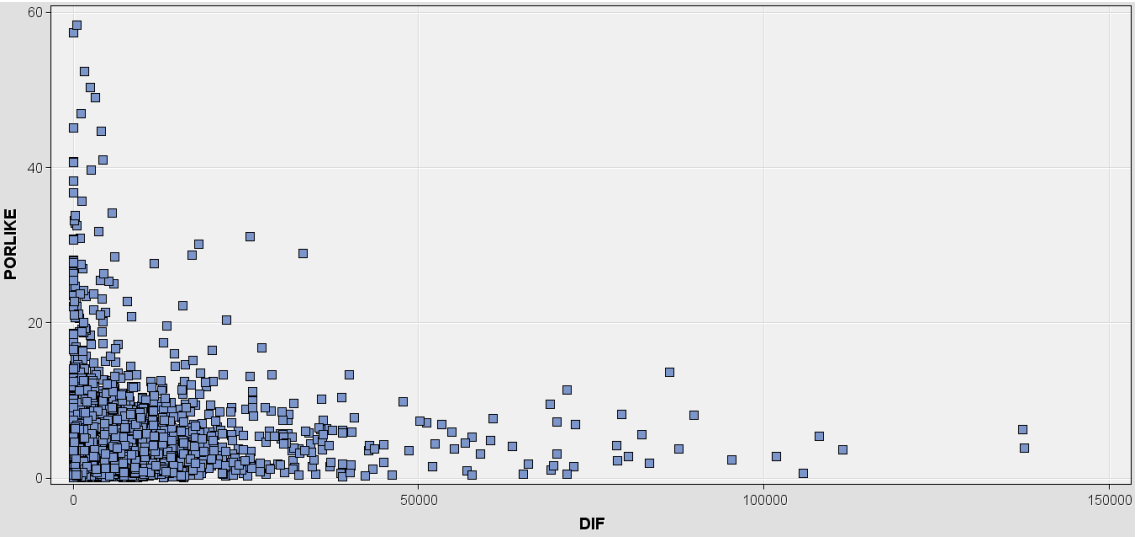

Figura 4.4: Gráfico de dispersión del tiempo entre publicaciones.

4.2.4. Día de la semana

Se da por hecho que los fines de semana la gente tiene más tiempo libre. Lo que se busca con esta variable es analizar si ese tiempo libre también influye en la utilización de la red social y por consiguiente en el porcentaje de *likes*, que será más alto cuanta más gente esté utilizando *Instagram*.

Esto lo vemos en la Figura 4.5, donde los días con más porcentaje de *likes* es el Domingo, el Viernes y el Sábado. Por el contrario el Miércoles es el día que claramente tiene menos valor la variable objetivo.

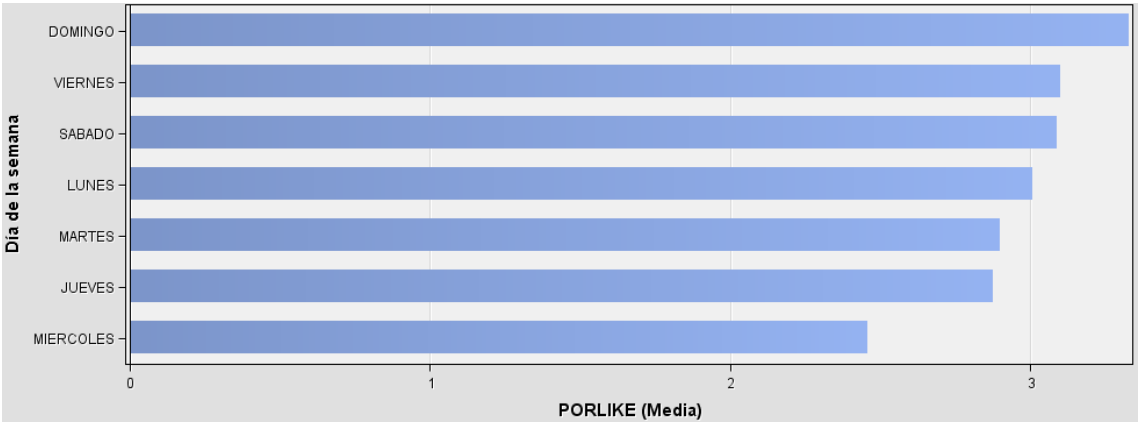

Figura 4.5: Porcentaje de *likes* por días de la semana.

4.2.5. *Caption*

El *Caption* es el título, pequeña explicación o comentario que acompaña a la imagen o vídeo subido. Es uno de los factores importantes que podría afectar al número de *likes* de una publicación ya que dependiendo del tema del que se hable o los *hashtag* que se mencionen en él, puede influir más o menos en los seguidores que vean la publicación.

Es por esto que se considera necesario extraer alguna referencia sobre qué tema podrían estar hablando los usuarios que escribieron el *Caption*. Para intentar generalizar de alguna forma el tema y poder agruparlos, se empleó la técnica de *Text Mining* sobre esta variable a través de SAS Miner.

Text Mining

La minería de textos [12] es una de las ramas de la lingüística computacional que trata de obtener información y conocimiento a partir de conjuntos de datos que en principio no tienen un orden o no están dispuestos en origen para transmitir esa información. La información que realmente interesa en la minería de textos es aquella contenida en los documentos pero de manera general, es decir, no en un texto en concreto sino que es la información global que tienen todos los registros, textos, documentos... de la colección. Por tanto, podemos decir que la Minería de Textos comprende cuatro actividades funda-

mentales:

1. Recuperación de información, es decir, seleccionar los textos pertinentes.

2. Descarte de palabras que no aporten información, como artículos, adverbios, conjunciones...

3. Extracción de la información incluida en esos textos: hechos, acontecimientos, datos clave, relaciones entre ellos, etc.

4. Encontrar asociaciones entre esos datos claves extraídos de los textos.

Una vez comprendido el concepto de esta técnica, se procede a explicar el procedimiento seguido para extraer información del *Caption*.

Primero, a través del Nodo de *Parsing*, se analizó sintácticamente el texto de la variable identificando nombres, verbos, adverbios... y realizando una limpieza de los artículos, preposiciones, etc.

En segundo lugar, se pasa a ejecutar el Nodo *Tema del texto*, el cual se encarga de analizar el texto de la variable *Caption* e intenta encontrar relaciones entre los términos que aparecen en dicha variable. Por cada agrupación de términos que crea, añade una variable nueva a la tabla de Entrenamiento, que adquiere para cada observación el peso que tiene ese tema en el *Caption* en cuestión. Finalmente, se obtienen 430 agrupaciones distintas, tal y como se ve en la Tabla 4.6.

ID tema	Tema	Número de términos	Nº docs
1	birthday,+happy birthday,+happy,+brother,yg	13	80
2	+day,christmas,good day,+w photo,7 days	3	192
3	de,dia,ano,todos,ontem	4	159
4	+night,good night,saturday,amazing night,+late	5	109
5	+photo,amivitale,fall,paulnicklen,+w photo	2	118
6	+love,+perform,chocolate,+brow,excited	1	93
7	love,+show,true,king,classic	4	81
8	+time,+great time,+well time,studio,+spend	7	91
9	today,lucky,important,training,second	3	80
10	repost,get_repost,regram,vi,dr	3	46
11	+good,good day,luck,good morning,good luck	13	97
12	beautiful,+brow,+hope,+lose,well	3	65
13	+present,allaboutyou,allaboutyoufromdeep,w-17,prada	5	32
14	+wear,burberry,+shirt,col,prada	5	69
15	thanksgiving,happy thanksgiving,+fill,paradise,+happy	8	43
16	+year,+start,+time,support,+bring	5	73
17	+look,novababe,+feeling,+hot,+lady	4	59
18	+life,7 days,+w photo,+favorite,learn	3	57
19	letter,comment,impossible,lmao,+favorite	4	26
20	+look,+brow,aur,different,happiness	4	58
21	tonight,cause,kehlani,lilyachty,+fake	7	59
22	+know,tha,well,statement,literally	4	49
23	always,+fast,+nice,+focus,+shot	5	51
24	en,el,todo,te,estar	9	43
25	regram,rg,+image,+tree,paulnicklen	5	30
26	+little,+catch,+lady,+pumpkin,cutie	8	48
27	+guy,+perform,hai,yeah,great	8	55
28	+friend,+good friend,+well friend,+good,japan	5	51
29	ready,+colour,+focus,+adventure,+unwrap	6	43
30	+mood,winter,stockholm,elissa,fight	3	30
31	para,casa,partido,pr,ver	9	44
32	+world,top,+dress,fashion,+lipstick	14	41
33	sarcasmonly,irm,seu,vi,+heart	1	22
34	por,por la,obrigado,obrigada,agradecer	21	68
35	+great,+great time,great win,+pleasure,team	17	61
36	+holiday,+favorite,party,line,cheer	12	49
37	+girl,+introduce,dance,baby,birthday	11	48

Figura 4.6: Lista de temas exportados por el Nodo Tema del Texto

Una vez realizado este análisis, se observó que los resultados no eran muy concluyentes, ya que producía una gran cantidad de variables poco definidas.

De modo que, utilizando esta lista como base orientativa, se creó una serie de variables de temas generales. Dichas variables dicotómicas toman el valor 1 cuando en el *Caption* aparecen una serie de términos relacionados con un tema determinado, y 0 cuando no. Las variables creadas son las que se muestran en la Tabla 4.4.

Nombre variables	**Descripción**
Deportes	Aparecen términos relacionados con el tema del deporte.
Felicitación	Aparecen términos relacionados con las felicitaciones.
Moda	Aparecen términos relacionados con la moda.
Música	Aparecen términos relacionados con la música.
Positivo	Aparecen términos relacionados con pensamientos positivos.

Tabla 4.4: Lista de variables de temas

La lista de términos utilizados para cada tema se encuentra en el Anexo A.2.

Además de lo mencionado anteriormente, se han conseguido extraer del *Caption* las siguientes variables:

- **Etiquetas:** Para mencionar a otros usuarios de Instagram en la publicación, se utiliza el carácter "@"seguido del nombre del usuario. Por tanto, contando los "@", se consigue extraer el número de menciones que aparecen en una publicación.
- **Hashtag:** Para categorizar con unos temas concretos se utiliza el carácter "#"seguido del tema en cuestión. Con esta variable se consigue extraer, contando los "#", la cantidad de temas que se han asignado a la publicación.
- **Repost:** Una de las posibilidades que da Instagram es la de publicar en tu propia cuenta una imagen de otra cuenta. Esto viene indicado por la palabra *"Repost"*, por lo que si esta palabra aparece en el *Caption* tomará valor 1, y en el caso contrario un 0.
- **Idioma:** Se ha generado un diccionario de palabras más utilizadas en inglés para compararlo con el contenido del *Caption*. Si hay un alto indice de coincidencia esta variable tomará el valor 1, si no el 0.
- **Sector:** Esta variable surge a raíz de la posibilidad de que este análisis pueda servir para cuantificar el valor que tiene una cuenta en términos comerciales. Por ejemplo, si un usuario va a anunciar algo, la empresa estará dispuesta a pagar más o menos según el éxito que tenga en la red. Aquí es donde puede influir el sector en el que se mueva. A continuación se detallan los sectores seleccionados, junto con algunos ejemplos que los definen:
  - *Belleza:* Maquillaje, peluquería.
  - *Celebrities:* Famosos por su vida personal.
  - *Cine:* Actores, actrices, directores.
  - *Deportes:* Deportistas, entrenadores, equipos de fútbol.
  - *Influencer:* Youtubers, Instagramers.

- *Marcas:* Empresas privadas.
- *Medios:* Periodistas, políticos.
- *Modelo:* Profesionales relacionados con la moda.
- *Música:* Cantantes, músicos.
- *Ocio:* Cuentas de viajes, restaurantes.

- **Palabra:** Es el número de palabras que aparecen en el *Caption*.
- **Letra:** Se ha contado el número de letras que aparecen en el *Caption*. Esta variable, así como la anterior, pretenden analizar el efecto de la longitud de los *Captions* en el porcentaje de *likes*, de manera que podamos determinar si son preferibles los *Captions* cortos o largos.

La variables que se obtienen del *Caption* tomarán valor 0 en caso de que la publicación no tenga un *Caption*.

4.2.6. Variables de los píxeles

Las imágenes se extraen de las publicaciones y se convierten al formato RGB en matrices de píxeles de 10x10, obteniendo de esta forma 300 variables que nos darán la información sobre la imagen. Con estas 300 variables se extrae también si la imagen está en blanco y negro o en color, pero tener este número de variables es una de las problemáticas que genera la descomposición, por eso es necesario llevar a cabo alguna técnica que permita manejar los datos con el menor número de variables posibles sin perder información. La técnica elegida que se aplicará en este trabajo es el Análisis de Componentes Principales (ACP).

Blanco y negro / Color

Para identificar si una imagen está en blanco y negro, se dividen las imágenes en cinco zonas y se escogen pixeles al azar de cada una. Si los tres componentes de RGB son iguales en todos los pixeles se puede concluir que la imagen está en blanco y negro. En este caso la

variable toma valor 1, por el contrario, si los pixeles son diferentes la variable toma valor 0 lo que indicará que está en color. Esta descomposición de los pixeles se realiza con el programa R.

Esta descomposición parte de que el color negro en RGB tiene las componentes (0,0,0) y el color blanco (255,255,255).

Análisis de componentes principales

El ACP [13] es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables). Es decir, ante un banco de datos con muchas variables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible. Los nuevos componentes principales o factores serán una combinación lineal de las variables originales, y además serán independientes entre sí. Un aspecto clave en ACP es la interpretación de los factores, ya que ésta no viene dada a priori, sino que será deducida tras observar la relación de los factores con las variables.

Un análisis de componentes principales tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que existe información redundante y, por tanto, pocos factores explicarán gran parte de la variabilidad total.

La elección de los factores se realiza de tal forma que el primero recoja la mayor proporción posible de la variabilidad original; el segundo factor debe recoger la máxima variabilidad posible no recogida por el primero, y así sucesivamente. Del total de factores se elegirán aquellos que recojan el porcentaje de variabilidad que se considere suficiente, en este caso tomaremos como suficiente el 75 % de la varianza.

Una vez realizado el análisis vemos que para poder explicar el 75 % de la varianza debíamos coger 33 componentes, los cuales explicarían el 75,3 %. En los gráficos comprobamos cuánto explican los 33 componentes. Observamos que, aunque el primer autovalor explique el 30,87 % y su autovalor valga 92.59, el resto de componentes explican bastante poco, y con la proporción acumulada llegaríamos al componente 33.

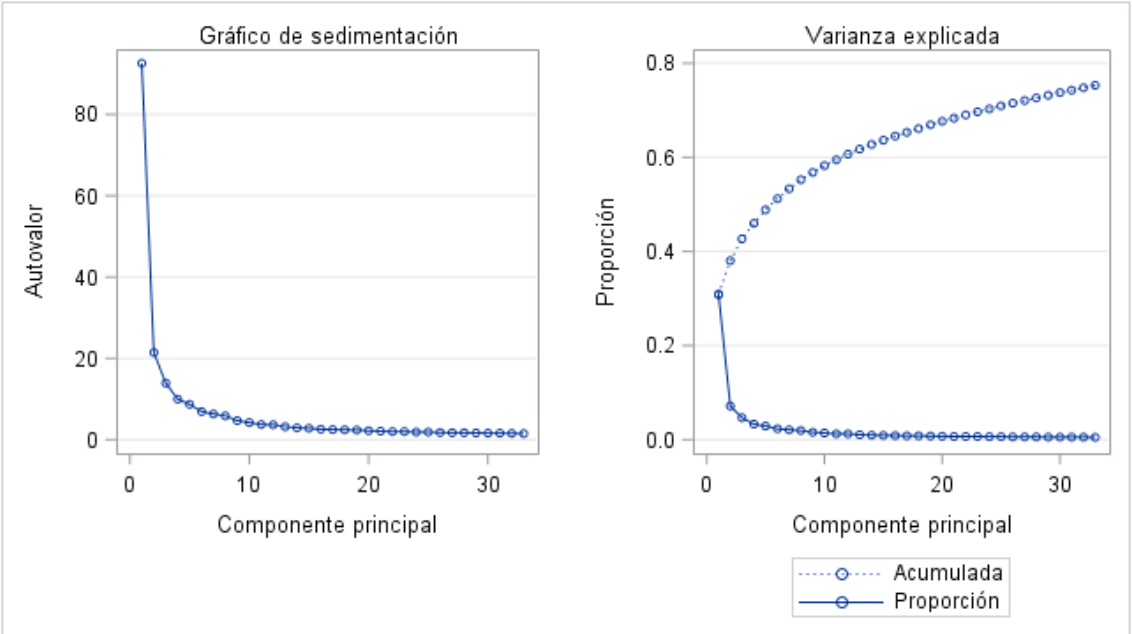

Figura 4.7: Gráfico de la varianza explicada por los Componentes

4.3. Variables Finales

Finalmente, se descartan las variables originales que toman una mera función identificativa como URL_imagen_nueva o URL_Ubicacion, además de variables de las que ya se había sacado información como FECHA, UBICACION o TEMA. Por otro lado se rechaza también la variable Comentarios, puesto que es un dato que no se puede predecir con anterioridad.

Así, el conjunto de datos estará formado por 57 variables, de las cuales 33 son los componentes principales, una es la variable dependiente *Porlike*, y el resto son nombradas a continuación, contenidas también en las tablas anteriores.

Variables Finales	
BYN	Media
CAP	Mes
Diasem	Sector
Dif	Ubi
Followers	Repost
Following	Moda
Hashtag	Deportes
Hora	Música
Idioma	Positivo
Palabra	Felicitación
Letra	día
Etiquetas	

Tabla 4.5: Variables Finales

Capítulo 5

Modelos de predicción

En el siguiente capítulo procederemos a presentar una serie de modelos estadísticos que se han llevado a cabo a lo largo del estudio, y que han sido previamente definidos en la sección 3.3.

5.1. Validación Cruzada

Para desarrollar los modelos se ha utilizado la técnica de validación cruzada [14], cuyo objetivo es evaluar los resultados de un análisis garantizando que son independientes de la partición entre datos de entrenamiento y prueba. Los pasos a seguir son los siguientes:

1. Dividir los datos aleatoriamente en k grupos.
2. Se realiza iterativamente la siguiente operación: Desde i=1 hasta k
   - Dejar aparte el grupo i
   - Construir el modelo con los grupos restantes
   - Estimar el error al predecir el grupo i.
3. El error de predicción se mide a partir del Average Squared Error (ASE) de los datos excluidos en cada iteración.

5.2. Comparación de Modelos

Para llevar a cabo la comparación entre los modelos que se van a explicar, se utilizará como indicador el ASE [15]. Se trata de un estimador que mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima. La diferencia se produce debido a la aleatoriedad o porque el estimador no tiene en cuenta la información que podría producir una estimación más precisa. En término matemáticos se calcula como:

$$ASE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}i - Yi)^2 \tag{5.1}$$

Donde $\hat{Y}_i$ es el valor predicho por el modelo y Y_i es el valor real. Al igual que la varianza, el ECM tiene las mismas unidades de medida que el cuadrado de la cantidad que se estima.

5.3. Regresión Lineal

Antes de indicar los diferentes modelos de regresión lineal que se han construido, debemos saber que se han utilizado una serie de métodos de selección de variables de cara a rechazar variables que realmente no aporten lo suficiente a la variable objetivo, y reducir la complejidad del modelo. Lo que también nos permite hacernos una idea de la estructura de los datos y sus relaciones. Los 3 métodos de selección de variables son:

- **Forward:** Este método consiste en, partiendo desde cero, ir introduciendo una a una las variables que mayor mejora produzcan en el modelo hasta que no haya ninguna variable más fuera del modelo que aporte información. La mejora del modelo se mide con el test de la F.

- **Backward:** Este método consiste en, partiendo del modelo que contiene todas las variables, ir eliminando una a una las variables que menos influyan en el modelo hasta que todas las variable del modelo sean significativas. Esta influencia se mide también con el test de la F.

- **Stepwise:** Este método es una mezcla de los anteriores. Es similar a forward, salvo por que se pueden eliminar las variables que han entrado en el modelo, ya que al entrar alguna, ésta pudiera hacer no significativo el aporte de otra. La eliminación de las variables se hace de acuerdo al método backward.

Otra modalidad de selección de variables que se ha utilizado es *Random Select*, proceso en que se realiza un método *stepwise* repetidas veces con diferentes archivos *train* (seleccionados como una muestra del conjunto de entrenamiento original) dándonos una lista de modelos seleccionados; donde el que más se repita será el que mejor se ajusta.

Por otra parte, para comparar cada modelo se utilizan 3 criterios, los cuales indican que cuanto menor sea el valor que tomen, mejor será el modelo que están evaluando. Estos criterios son los siguientes:

- **AIC** (Akaike information criterion): $n \ln\left(\frac{SSE}{n}\right) + 2p$
- **BIC** (Bayesian information criterion): $n \ln\left(\frac{SSE}{n}\right) + 2q(p + 2 - q)$, donde $\rightarrow q = \frac{n}{n-p}$
- **SBC** (Schwarz criterion criterion): $n \ln\left(\frac{SSE}{n}\right) + p \ln(n)$

Teniendo en cuenta que el primer sumando coincide para los tres criterios, la diferencia entre ellos se reduce a la penalización del número de parámetros, siendo el SBC el que más penaliza y AIC generalmente el que menos. Se han construido una serie modelos de regresión lineal cuya combinación de posibilidades podemos ver en la Tabla 5.1.

Modelo	Procedimiento	Método de Selección	Criterio de Selección de Variables	Nº de Grupos en Validación Cruzada	Nº de parámetros finales de la RL	Semilla Inicial	Semilla Final
1	GLM	Backward	AIC	6	21	20119	20129
2	GLM	Fordward	SBC	6	11	20119	20129
3	GLM	Stepwise	BIC	6	21	20119	20129
4	Random Select	Stepwise	AIC	6	17	20119	20129
5	Random Select	Stepwise	SBC	6	5	20119	20129
6	Random Select	Stepwise	BIC	6	16	20119	20129
7	Ninguno	Ninguno	Ninguno	6	62	20119	20129
8	Ninguno	Ninguno	Ninguno	6	62	20119	20129

Tabla 5.1: Modelos de Regresión Lineal

Una vez ejecutados todos los modelos, procedemos a compararlos mediante un diagrama de cajas donde se muestra el Error Cuadrático Medio de cada modelo:

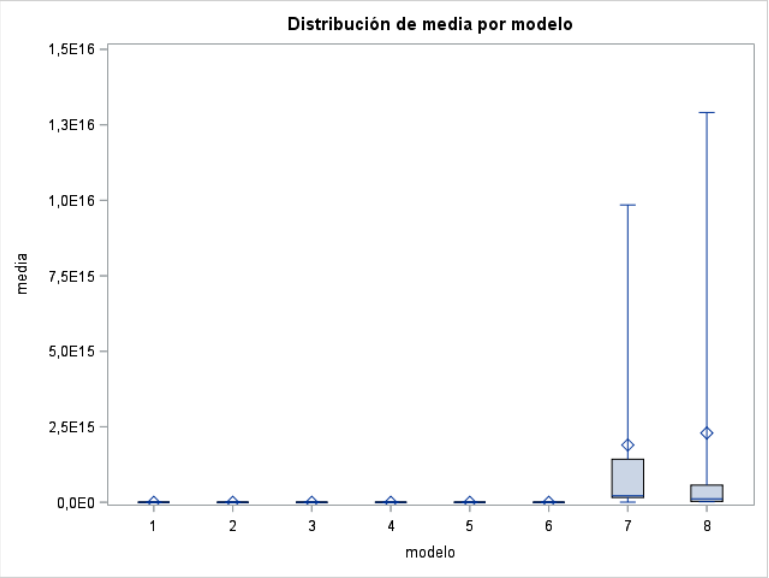

Figura 5.1: Diagrama de cajas de los 8 modelos de Regresión Lineal

Observando el diagrama, se puede ver a simple vista que los modelos 7 y 8, que son los únicos modelos en los que no se ha hecho selección de variables, son los que mayor error tienen, además de una gran variabilidad. Por tanto podemos excluirlos directamente, haciendo zoom en el resto:

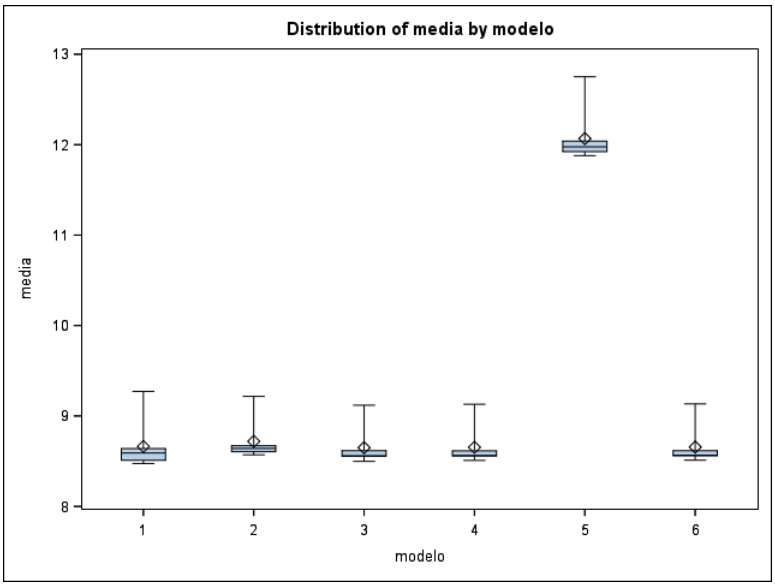

Figura 5.2: Diagrama de cajas de 6 modelos de Regresión Lineal

AIC	6818.5266	BIC	6821.0724
SBC	6977.8237	C(p)	27.0000
AIC	7121.7259	BIC	7124.4958
SBC	7318.3705	C(p)	33.0000

Figura 5.4: Criterios de comparación de la Regresión Lineal

Aislamos los modelos 1,2,3,4 y 6, ya que tienen un error cuadrático medio más bajo que los demás, y en la Figura 5.3 comprobamos que los modelos 3,4 y 6 están prácticamente igualados, aunque el 3 y 4 tienen un error ligeramente inferior al 6.

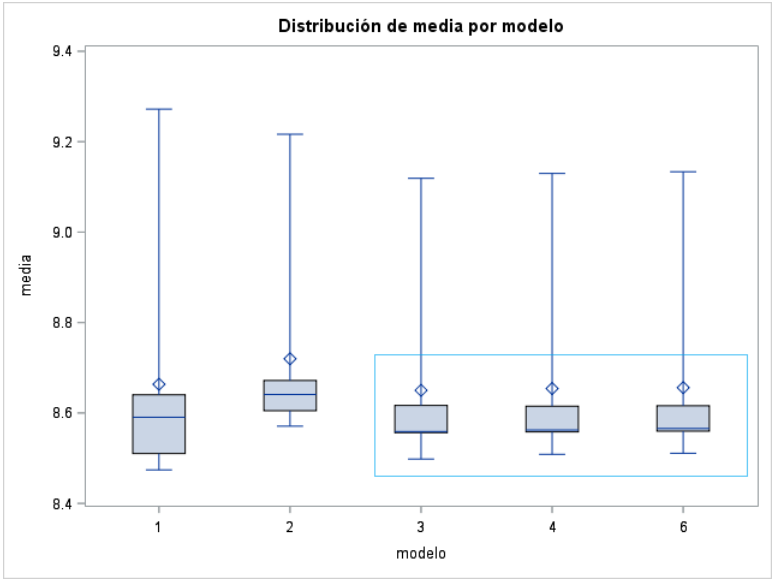

Figura 5.3: Diagrama de cajas de los 3 mejores modelos de Regresión Lineal

Finalmente, para terminar de comparar se construye en SAS Miner las 2 regresiones finalistas (3 y 4), reproduciendo exactamente la misma situación que plantean. Una vez ejecutado, se comprueba que el modelo con un mejor ajuste es el una Regresión Lineal con selección de variables mediante *Random Select* y utilizando el criterio de AIC (Modelo 4). Para la comparación de estos 2 modelos podemos ver en la Figura 5.4 que los 3 criterios son menores en el Modelo 4 (resaltado con un recuadro azul), lo que indica que es mejor que el Modelo 3.

Observando las variables utilizadas, vemos en la Figura 5.5 que prácticamente todas

Type 3 Analysis of Effects

Effect	DF	Suma de cuadrados	F-Valor	Pr > F
DIA	1	101.2630	8.17	0.0043
DIF	1	78.2112	6.31	0.0121
ETIQUETAS	1	26.1120	2.11	0.1467
FOLLOWERS2	1	204.0004	16.46	<.0001
HASHTAG	1	111.2578	8.98	0.0028
IDIOMA	1	535.2981	43.19	<.0001
LETRA	1	71.5171	5.77	0.0164
MEDIA	1	67.2197	5.42	0.0199
MES	1	114.8065	9.26	0.0024
PALABRA	1	28.4793	2.30	0.1297
Prin1	1	5.4635	0.44	0.5068
Prin18	1	26.0368	2.10	0.1473
Prin3	1	146.4832	11.82	0.0006
Prin8	1	32.3016	2.61	0.1065
SECTOR	9	1321.8146	11.85	<.0001

Figura 5.5: Efectos de las variables de la Regresión Lineal

son significativas para el modelo, ya que para el estadístico F el *p-valor* $< 0{,}05$. El test de la F nos confirma que al añadir esa variable, la información explicada por el modelo sigue siendo mayor que el error que cometemos.

En cuanto a la interpretación de la regresión, partiendo de los estimadores de las variables de la Figura 5.6, concluiríamos que: $Porlike = 5{,}0480 + 0{,}000025 * Dif - 0{,}2122 * Etiquetas - 0{,}0000000226 * Followers - 0{,}3538 * Hashtag - 1{,}1650 * (Idioma = 0) - 0{,}0256 * Letra + 0{,}000417 * Media - 0{,}2148 * Mes + 0{,}0672 * Palabra - 0{,}00491 * Prin1 - 0{,}0655 * Prin18 + 0{,}0635 * Prin3 - 0{,}0493 * Prin8 + 1{,}7740 * (Sector = Cine) + 1{,}3285 * (Sector = Musica) - 0{,}4344 * (Sector = Belleza) + 1{,}3873 * (Sector = Celebrities) + 1{,}1939 * (Sector = Deportes) + 1{,}4173 * (Sector = Influencer) - 0{,}4891 * (Sector = Marcas) + 0{,}1579 * (Sector = Medios) + 1{,}0430 * (Sector = Modelos)$. De donde podemos interpretar, entre otros, que:

- **Dif** (Minutos): El porcentaje de *Likes* aumentará 0.000025 puntos porcentuales por cada minuto de diferencia que haya entre una publicación y la anterior. Por lo tanto, podemos ver que es preferible que dos publicaciones consecutivas no estén muy próximas en el tiempo.

- **Idioma**: El porcentaje de *Likes* disminuirá 1.17 puntos porcentuales si el *Caption* no está escrito en inglés, por lo que se trata de una de las variables más influyente

Parameter		DF	Estimate	Standard Error	Valor t	Pr > |t|
Intercept		1	5.0480	0.7188	7.02	<.0001
DIF		1	0.000024	0.000010	2.41	0.0162
ETIQUETAS		1	-0.2122	0.1463	-1.45	0.1469
FOLLOWERS2		1	-2.26E-8	5.651E-9	-4.01	<.0001
HASHTAG		1	-0.3638	0.1193	-3.05	0.0023
IDIOMA	0	1	-1.1650	0.1801	-6.47	<.0001
IDIOMA	1	0	0	.	.	.
LETRA		1	-0.0256	0.0107	-2.40	0.0163
MEDIA		1	0.000417	0.000180	2.32	0.0203
MES		1	-0.2148	0.0514	-4.18	<.0001
PALABRA		1	0.0672	0.0443	1.52	0.1295
Prin1		1	-0.00491	0.00742	-0.66	0.5078
Prin18		1	-0.0655	0.0443	-1.48	0.1392
Prin3		1	0.0635	0.0185	3.43	0.0006
Prin8		1	-0.0493	0.0290	-1.70	0.0895
SECTOR	Belleza	1	-0.4344	0.6168	-0.70	0.4813
SECTOR	Celebrities	1	1.3873	0.5024	2.76	0.0058
SECTOR	Cine	1	1.7740	0.3841	4.62	<.0001
SECTOR	Deportes	1	1.1939	0.4027	2.97	0.0031
SECTOR	Influencer	1	1.4173	0.4219	3.36	0.0008
SECTOR	Marcas	1	-0.4891	0.4035	-1.21	0.2256
SECTOR	Medios	1	0.1579	0.6275	0.25	0.8014
SECTOR	Modelo	1	1.0430	0.4489	2.32	0.0202
SECTOR	Música	1	1.3285	0.3875	3.43	0.0006
SECTOR	Ocio	0	0	.	.	.

Figura 5.6: Estimadores de los parámetros de la Regresión Lineal

en el análisis.

- **Sector**: Las cuentas relacionadas con el cine (actores, directores...) tienen mayor influencia en la variable objetivo, donde el porcentaje de *Likes* aumentará 1.78 puntos porcentuales si la cuenta pertenece a este sector, en comparación con la categoría Ocio que es la de referencia.

5.4. Redes Neuronales

A continuación, veremos los distintos modelos construidos para encontrar una red neuronal óptima. Los parámetros que se han ido combinando son:

- **Nº de nodos ocultos**: Es el número de unidades que aparecerán en la capa oculta.
- **Algoritmo de Optimización**: Pueden ser Levenberg-Marquardt, Quasi-Newton, Conjugate Gradient, Trust-Region, Back Prop o DBLDog.
- **Función de Activación**: F. Tangente, F. Logística, F. Arcotangente, F. Lineal, F. Seno, F. Exponencial Normalizada y F. Gaussiana.

Nº de Modelos	Nodos Ocultos	Algoritmo de Optimización	Función de Activación	Semilla Inicial	Semilla Final
7	2 a 14	Quanew	TANH	20119	20129
15	2 a 30	Brop	ARC	20119	20129
15	2 a 30	Levmar	GAUSS	20119	20129
6	2 a 12	Levmar	TANH	20119	20129
7	10	Levmar	Todas	20119	20129
4	10	Todos	TANH	20119	20129

Tabla 5.2: Modelos de Redes Neuronales

En la Tabla 5.2 se recogen las distintas situaciones presentadas, donde los ASE de los mejores mdoelos se pueden comprobar en la Tabla 5.3.

MODELO1	ASE
TANH_LEVMAR 10	8,7089
TANH_LEVMAR 12	8,8506
TANH _LEVMAR 8	8,9505
TANH_LEVMAR 6	9,2037
TANH_LEVMAR 2	9,3898
TANH_LEVMAR 4	9,4286
ARC _LEVMAR 10	10,4926
LOG _LEVMAR 10	10,5765
GAU_LEVMAR 8	10,7801
ARC_BROP 6	10,8458
GAU_LEVMAR 10	10,8915
GAU_LEVMAR 4	10,9453
ARC_BROP 4	10,9507
TANH_QUANEW 10	10,9742
GAU_LEVMAR 14	10,998
ARC_BROP 10	11,0132
TANH_BPROP 10	11,0132
GAU_LEVMAR 12	11,0208
ARC_BROP 8	11,0226
TANH_QUANEW 8	11,0311

Tabla 5.3: Error Cuadrático Medio de los 20 mejores modelos de Redes Neuronales

En la Figura 5.7 se muestran los 10 modelos de redes con el menor error cuadrático medio. Se observa que los modelos con la función Tangente y el algoritmo de Levenberg-Marquardt dan mejor resultado con un número de nodos ocultos de 8, 10 y 12, puesto que con menos nodos parece aumentar su variabilidad.

Una vez comparado el error de todos los modelos, se concluye que el mejor modelo es

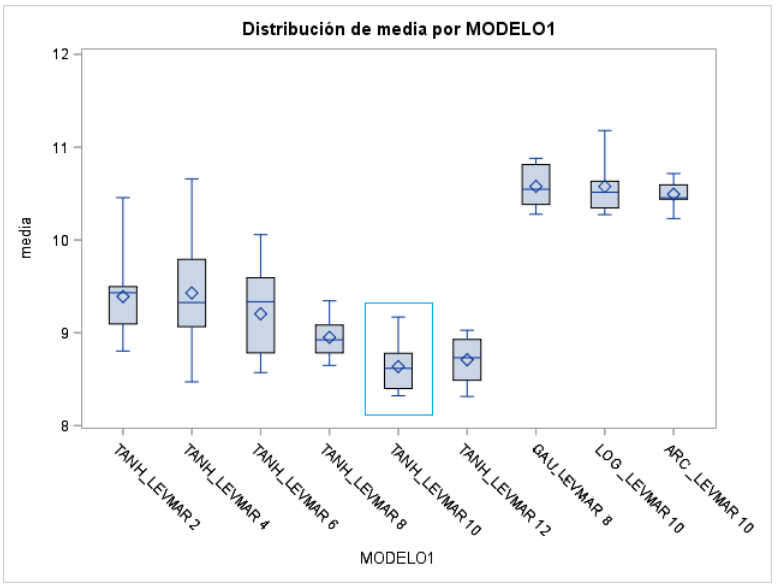

Figura 5.7: Diagrama de cajas de los 10 mejores modelos de Redes Neuronales

una Red Neuronal que consta de:

- Número de nodos ocultos: 10
- Algoritmo de Optimización: Levenberg-Marquardt
- Función de Activación: Tangente

En la construcción de la red, el modelo toma unos valores iniciales aleatorios partiendo de la semilla que se le ha indicado. Con esos valores se calcula la predicción para todas las observaciones, y el error correspondiente, buscando los valores de los parámetros que hagan decrecer el error en cada iteración. Cuando llegamos a la iteración 100 obtendremos el mínimo error.

Comparando la red ganadora con el mejor modelo de regresión, vemos que aunque el ASE de ambos modelos ronde entre el 8,6 y 8,8, el modelo de red neuronal tiene una variabilidad mucho mayor, por tanto diríamos que por ahora el mejor modelo es la regresión.

Trasladando dicho modelo a SAS Miner nos permite observar en una de las tablas exportadas del nodo Red Neuronal, el Peso Final de las variables del modelo. A partir

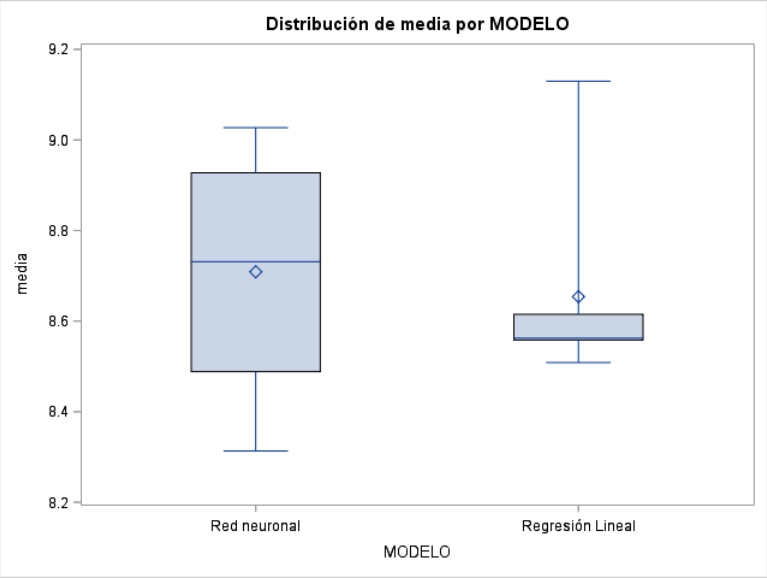

Figura 5.8: Comparación Regresión Lineal vs Red Neuronal

de éstos valores podremos calcular la importancia relativa [16] de cada variable sumando, para cada una de ellas, el valor absoluto de los pesos de las variables a los nodos ocultos.

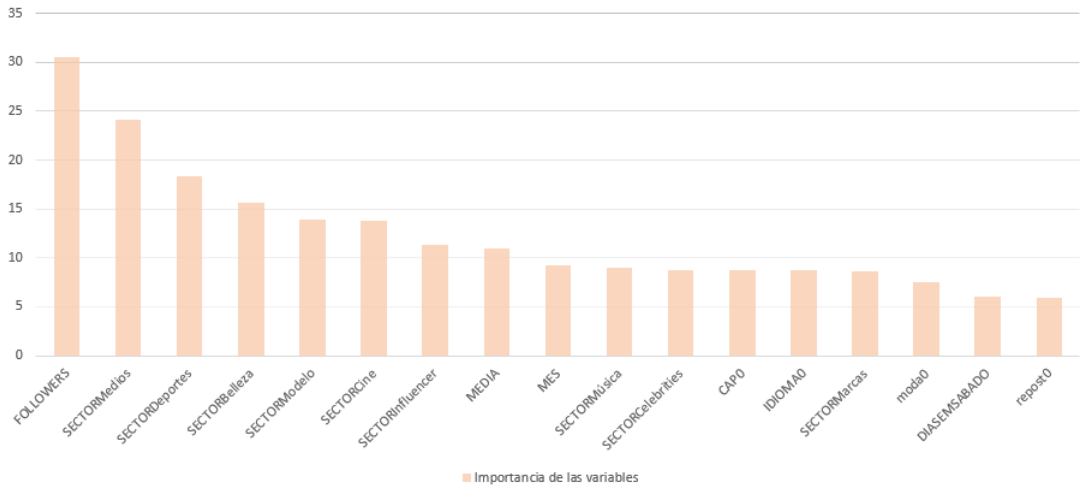

Figura 5.9: Importancia de las variables de la Red Neuronal

Concluimos la sección comentando la Figura 5.9, donde se representan las variables explicativas que más importancia tienen sobre el modelo, ordenadas de forma descendente. Pudiendo sacar como conclusión que los factores que más influencia tienen a la hora de

predecir el porcentaje de *Likes* con el modelo se red neuronal elegido son:

- *Followers*: Nº de seguidores que tenga la cuenta.
- Sector: El sector al que pertenezca la cuenta de *Instagram*.
- Media: Cantidad de publicaciones que tenga la cuenta.
- Mes: El mes en el que se publique.
- Cap: Si la publicación tiene *Caption* o no.
- Idioma: Si el *Caption* está escrito en inglés o no.
- Moda: Si la publicación está relacionada con la moda o no.
- Día: Si el día de la semana es sábado.
- Repost: Si la publicación es un repost o no.

5.5. Random Forest

En la siguiente sección veremos el proceso de construcción de los modelos de *Random Forest*, junto con los parámetros que se han tenido en cuenta, como ya se comentó en la Sección 3.3.3, y que son:

- El número máximo de árboles construidos.
- El número de variables p a muestrear en cada nodo.
- El tamaño de la hoja (número mínimo de observaciones para que se considere una hoja).
- La profundidad del árbol.
- El *p-valor* para las divisiones en cada nodo.

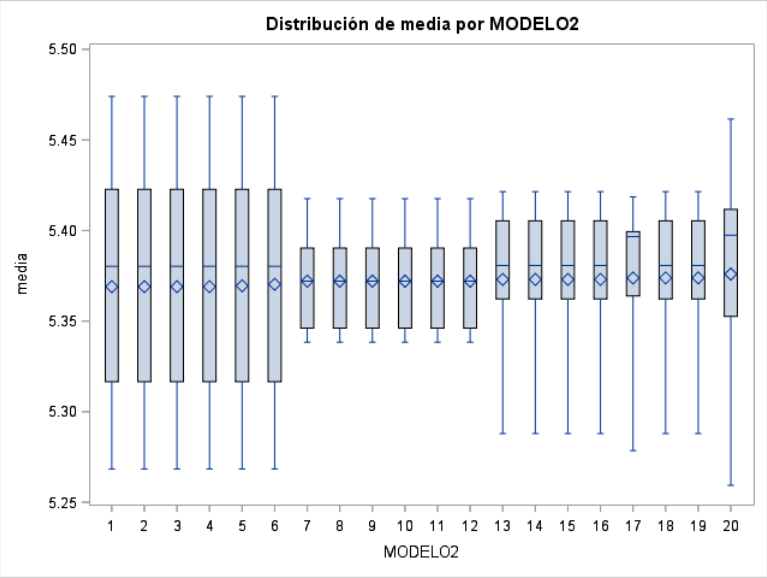

Figura 5.10: Diagrama de cajas de Random Forest

El algoritmo Random Forest trata de construir el mejor modelo incorporando un remuestreo de observaciones y variables pudiendo así generalizar el modelo, y reducir el sobreajuste. La Tabla 5.4 recoge las diferentes combinaciones que se han llevado a cabo para llegar a crear finalmente 863 random forests.

Nº árboles	**Nº de variables**	**Tamaño de la hoja**	**Profundidad del árbol**	**P-valor**
10 a 200	1 a 60	50 a 300	15 a 25	0.1

Tabla 5.4: Modelos de Random Forest

En la Figura 5.10 se observan los 20 mejores modelos de random forest. Los que obtienen menor ASE son del 7 al 12 cuya mediana es ligeramente superior a 5,35. Se pueden apreciar con más detalle estos modelos en la Figura 5.11

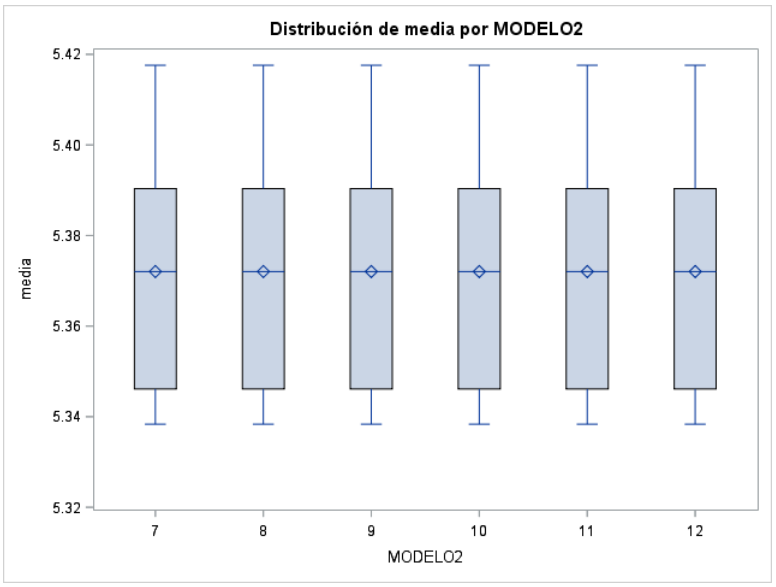

Figura 5.11: Diagrama de cajas de los mejores modelos de Random Forest

Por tanto, el modelo elegido será el que tenga las siguientes características:

- Número de árboles: 160 árboles.
- Número de variables: 51 variables por nodo.
- Tamaño de la hoja: 50 observaciones.
- Profundidad del árbol: 15.

A la vista de los resultados obtenidos por el mejor modelo de random forests, hemos conseguido disminuir el ASE en un 37 % en relación con el modelo de red neuronal y el de regresión lineal, alcanzando un ASE ligeramente superior a 5.35, cuando previamente se situaba en torno a 8,5 (Figura 5.12).

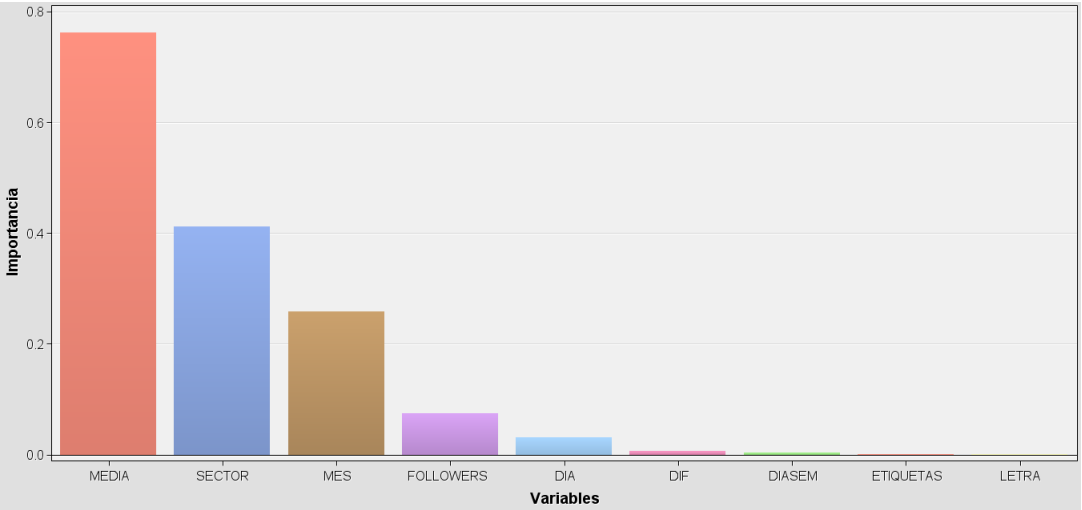

Figura 5.13: Importancia de las variables para Random Forest

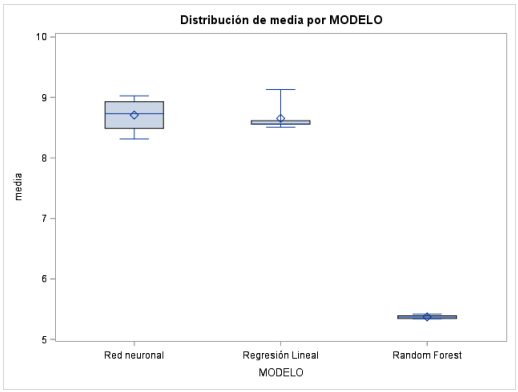

Figura 5.12: Comparación Regresión Lineal vs Red Neuronal vs Random Forest

Para concluir, debemos saber que esta técnica de predicción presenta la desventaja de la perdida de interpretabilidad de los resultados, donde solo se puede evaluar la importancia de cada una de las variables explicativas del modelo, creando un ranking de las variables según su frecuencia utilizada en el algoritmo.

Se observa en la Figura 5.13, que las 5 variables con más importancia para el modelo son:

- Media: Cantidad de publicaciones que tenga la cuenta.

- Sector: El sector al que pertenezca la cuenta de *Instagram*.
- Mes: El mes en el que se publique.
- *Followers*: Nº de seguidores que tenga la cuenta.
- Día: Día del mes en que se publica la imagen.

5.6. Gradient Boosting

Con el algoritmo que vamos a desarrollar a continuación pretendemos mejorar la estructura del árbol de decisión, creando series de dichos árboles mediante el ajuste del residual de la predicción del árbol anterior en las series. En esta sección veremos las diferentes composiciones de parámetros utilizadas, las cuales se muestran en la Figura 5.5. Los parámetros a tener en cuenta son:

- El número de iteraciones m a promediar.
- El número de hojas final o la profundidad del árbol.
- El *Maxbranch* (número de divisiones máxima en cada nodo).
- El número de observaciones mínimo en una rama-nodo.
- Parámetro de Regularización o *Shrinkage*: Grado con el que se ajustará el modelo en cada una de las iteraciones.

Parámetro de regularización	**Iteraciones**	**Tamaño de la hoja**	**Profundidad del árbol**	**Nº de divisiones de un nodo**
10 a 60	0.01, 0.05, 1.3	50 a 150	15 a 20	2 a 4

Tabla 5.5: Modelos de Gradient Boosting

Una vez realizadas todas las combinaciones posibles, se construyen en total 96 modelos de los cuales se puede ver claramente, tanto en la Figura 5.14 como en la Tabla 5.6, que hay 2 con un error menor que el resto.

MODELO	ASE	MODELO	ASE
Modelo1	3,3165	Modelo11	5,5278
Modelo2	3,3208	Modelo12	5,5278
Modelo3	4,3184	Modelo13	5,5392
Modelo4	4,3184	Modelo14	5,5392
Modelo5	4,3438	Modelo15	5,5392
Modelo6	4,3438	Modelo16	5,5392
Modelo7	4,3438	Modelo17	5,7539
Modelo8	4,3438	Modelo18	5,7541
Modelo9	4,7911	Modelo19	6,084
Modelo10	4,7912	Modelo20	6,084

Tabla 5.6: Ranking 20 mejores modelos de Gradient Boosting

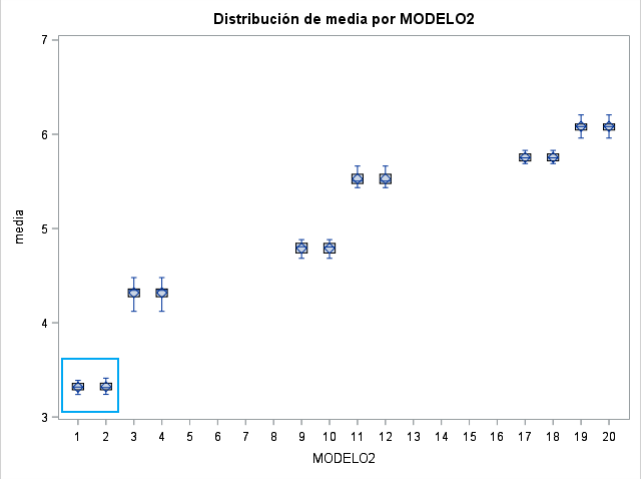

Figura 5.14: Diagrama de cajas de Gradient Boosting

En la Figura 5.14 observamos que los modelos se comportan de manera similar en grupos de 2. Comparando los parámetros de estos modelos vemos que lo único que cambia es *maxdepth* que toma los valores 15 o 20. Este parámetro nos indica la profundidad máxima del árbol, por lo que podemos deducir que nunca llega a 15, produciendo similares salidas en ambos modelos. Por otra parte, los modelos 5,6,7,8,13,14,15 y 16 obtienen el mismo error en cada iteración de la validación cruzada, por lo que no podremos apreciar su representación en la Figura 5.14, aunque sí podemos ver sus medias en la Figura 5.15.

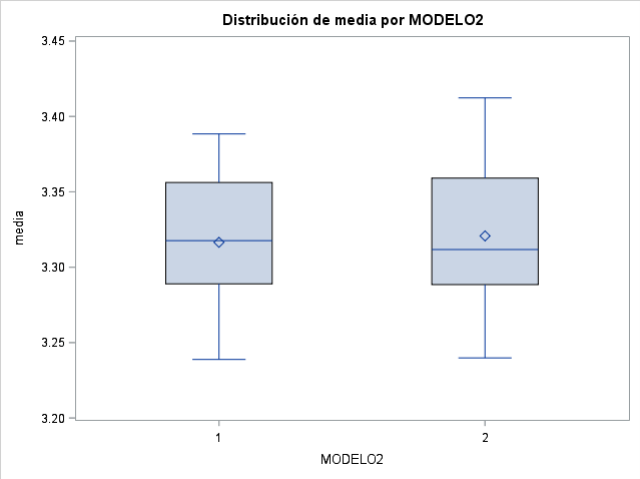

Figura 5.16: Diagrama de cajas de los 2 mejores modelos Gradient Boosting

MODELO2	MEDIA
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
5	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702
6	4.3438017702

Figura 5.15: ASE de modelos 5 y 6 de Gradient Boosting

- Número de iteraciones: 60
- Profundidad máxima del árbol: 20
- El *Maxbranch*: 2
- Tamaño hoja: 50
- Parámetro de Regularización:0.05

Se aprecia una gran mejora en las predicciones de este modelo respecto a los anteriores,

comparándolos en la Figura 5.17 vemos que el error cuadrático medio es un 38 % menor a random forest.

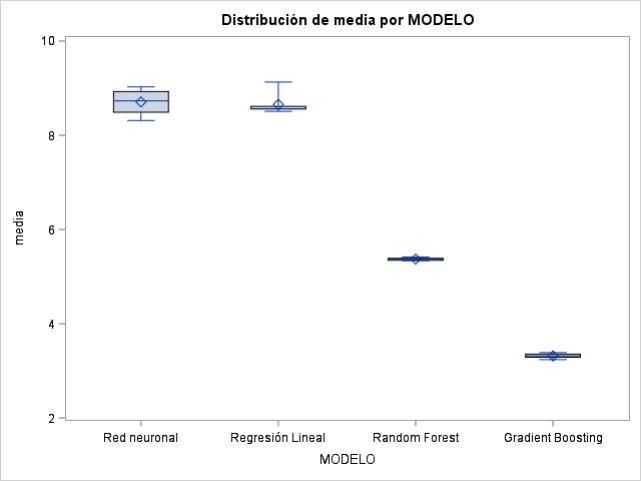

Figura 5.17: Comparación de modelos

Reproduciendo el mejor modelo en SAS Miner conseguimos ver la importancia de las variables, donde volvemos a encontrar que muchas de las variables con mayor importancia coinciden con en los modelos de Redes Neuronales y Random Forest.

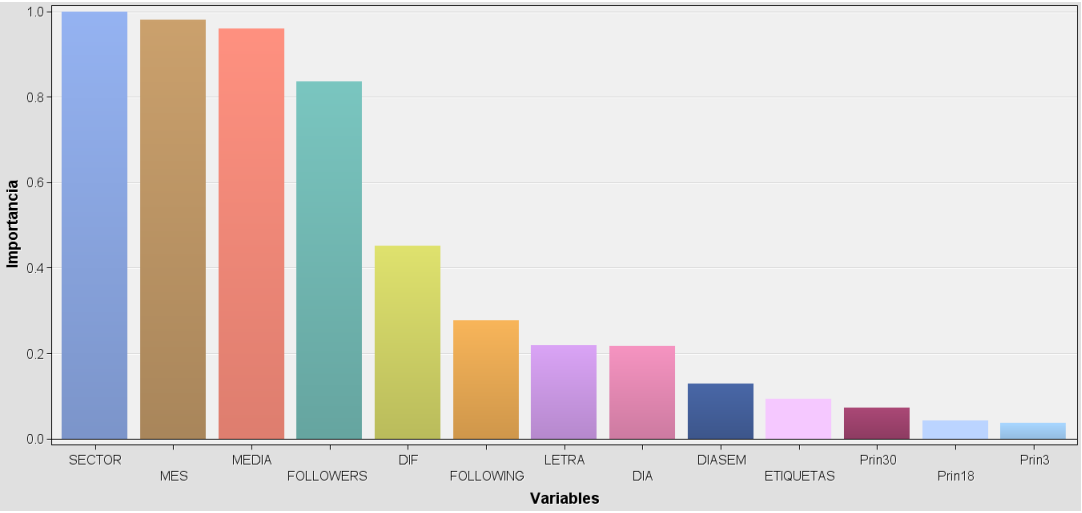

Figura 5.18: Importancia de las variables para Gradient Boosting

Analizando la Figura 5.18 vemos que para el modelo de Gradient Boosting, los 5

factores que más intervienen a la hora de predecir la variable objetivo con el menor error posible son:

- Sector: El sector al que pertenezca la cuenta de *Instagram*.
- Mes: El mes en el que se publique.
- Media: Cantidad de publicaciones que tenga la cuenta.
- *Followers*: Nº de seguidores que tenga la cuenta.
- Dif: La diferencia, en minutos, entre una publicación y la anterior.

5.7. Ensamble de Modelos

Como sección final del análisis de modelos, se desarrollará un ensamblado compuesto por el mejor modelo obtenido en los apartados anteriores con el fin de conseguir algoritmos que alcancen un error cuadrático medio menor al error cometido por cada modelo por sí solos. Para ello se obtienen las combinaciones de éstos logrando obtener modelos que mejoren las predicciones, y éstas son:

- Red Neuronal + Regresión Lineal
- Red Neuronal + Random Forest
- Red Neuronal + Gradient Boosting
- Regresión Lineal + Random Forest
- Regresión Lineal + Gradient Boosting
- Random Forest + Gradient Boosting
- Red Neuronal + Regresión Lineal + Random Forest
- Red Neuronal + Random Forest + Gradient Boosting

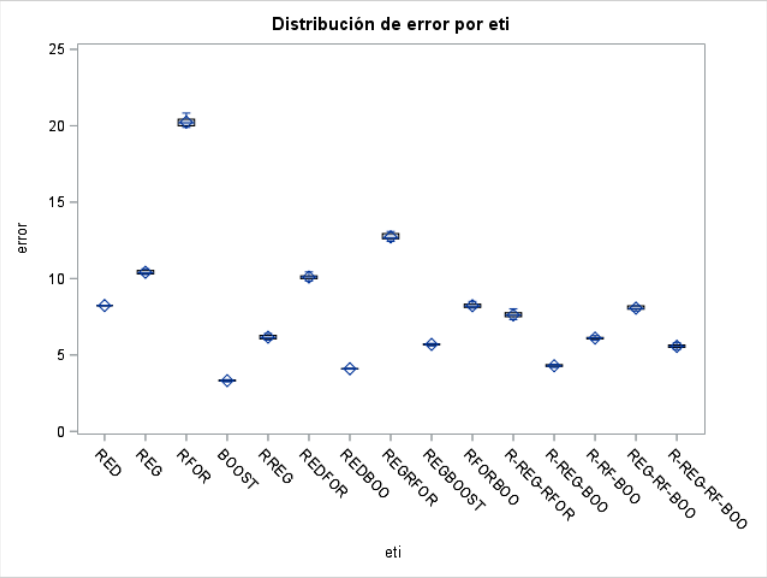

Figura 5.19: Diagrama de cajas de Ensamble

- Red Neuronal + Regresión Lineal + Gradient Boosting

- Regresión Lineal + Random Forest + Gradient Boosting

- Red Neuronal + Regresión Lineal + Random Forest + Gradient Boosting

Como resultado de la comparación de modelos de la Figura 5.19, podemos percibir que los que tienen un menor error son Gradient Boosting, Regresión con Random Forest y un combinado de Red, Random Forest y Gradient Boosting. Los cuales observamos más de cerca en la Figura 5.20.

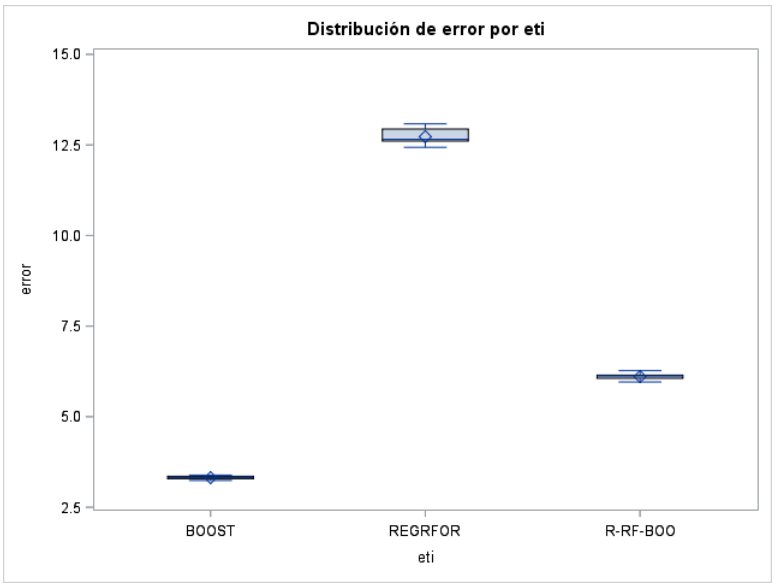

Figura 5.20: Diagrama de cajas de los 3 mejores modelos de Ensamble

Concluimos este estudio demostrando que el mejor modelo para predecir nuestra variable objetivo con los datos con los que contamos es el Gradient Boosting, con un Error Cuadrático Medio de 3,31. Si tenemos en cuenta que la varianza de la variable objetivo es 14.66 para el total de observaciones (n=8581); podríamos calcular el R^2 a partir de la siguiente ecuación: $1-(ASE/\sigma^2)$, es decir $1-(3,31/14{,}66)$. Obteniendo un R^2 aproximado del 77.4 %.

Capítulo 6

Conclusiones y trabajo futuro

En la primera sección de este capítulo se muestran las conclusiones finales del proyecto, obtenidas en función de los objetivos iniciales. En la segunda sección se muestran las posibles líneas de trabajo futuro, además de las posibilidades que tiene este análisis dependiendo de las circunstancias.

6.1. Conclusiones

El principal objetivo de este estudio era, contando con el tipo de información de partida, encontrar el mejor modelo predictivo posible para poder explicar la variable objetivo Porcentaje de *Likes*; el cual ha sido alcanzado de forma satisfactoria. En términos generales, el grado de consecución de los objetivos secundarios ha sido el esperado, cumpliendo los objetivos establecidos.

1. Conocer las variables y extraer información útil a partir de la información inicial de los datos. Este objetivo se ha cumplido principalmente en la fase de Depuración y Exploración de los datos, donde se ha podido comprobar la influencia real de cada variable explicativa. Además de poder profundizar en algunos factores que podría parecer no tener relevancia, pero que finalmente se ha demostrado que cuando esos factores adquieren un valor determinado pueden contribuir fuertemente en el modelo

predictivo. Derivado de este punto se ha conseguido entender de manera más precisa el comportamiento de esta red social.

2. Reducir la dimensionalidad de la gran cantidad de variables creadas a raíz de los píxeles de la imagen. Este objetivo se ha cumplido eficientemente ya que se consiguió reducir 300 variables, relativas a la información de los píxeles, en 33 componentes principales; facilitando en gran medida la construcción de modelos y agilizando la capacidad computacional de los procesos estadísticos ejecutados durante el proceso.

3. Eliminar las variables que no aporten información necesaria al modelo predictivo. Se ha llevado a cabo la selección de variables para la construcción del modelo de Regresión Lineal, eliminando así las variables que tiene menos influencia sobre la variable objetivo.

4. Realizar una comparativa de los diferentes modelos predictivos de forma que sea posible identificar el modelo que mejor se adapte a nuestros datos. Se ha realizado un balance de los mejores modelos construidos en el estudio, identificando como mejor modelo Gradient Boosting.

5. Determinar qué variables pueden ser más influyentes a la hora de intentar alcanzar una buena proporción de *likes*. Contemplando las variables que han ido apareciendo a lo largo del análisis, podríamos decir que las más importantes son el número de *Followers*, el número de publicaciones, el Sector de la cuenta y el mes en el que se publique. Sin embargo, los píxeles de la imagen, así como la hora de publicación, no han tendio la relevancia esperada, lo que nos lleva a pensar que quizás no se hayan incluido de la mejor manera posible.

A nivel personal, cabe mencionar que este trabajo ha conseguido aportar una visión con más perspectiva del proceso de SEMMA, dando además la oportunidad de ampliar el conocimiento en lenguaje de código en R-Studio.

6.2. Trabajo a futuro

A raíz de este estudio han ido surgiendo una serie de ideas y necesidades, algunas en el transcurso de la investigación, y otras al final. Algunas de ellas son:

- Siendo conscientes del tiempo del que se disponía y de la capacidad del *hardware* existente, una de las lineas a cubrir en el futuro sería el desarrollo de más modelos con el mayor número de combinaciones de parámetros posible, para conseguir acercarnos lo máximo posible al mejor modelo existente.

- Otra de las líneas a tratar sería un análisis más exhaustivo de la imagen. En este trabajo, desde el principio, se ha reducido el tamaño de las imágenes para poder realizar una descomposición más sencilla de las fotos. Pero como consecuencia no se ha adquirido información suficientemente relevante como para que los modelos la tratasen como variables de importancia para la predicción. Por eso, uno de los siguientes pasos de este análisis sería un planteamiento para conseguir una descomposición sin disminuir demasiado las dimensiones de las imágenes, además de encontrar la forma de detectar los efectos fotográficos aplicados.

- Aunque en esta investigación se han utilizado un número considerable de variables, las aplicaciones van avanzando cada día más, apareciendo más factores susceptibles de influir en la cantidad de *Likes* de una publicación. Por tanto, otra línea sería encontrar esas variables y cruzarlas con las ya existentes.

- Partiendo de los dos puntos anteriores, un pasó a seguir sería la automatización del proceso de extracción de la información. Por ejemplo, la descarga y descomposición de imágenes directamente de la página web.

- Como última línea de trabajo, este estudio sería un buen punto de partida para desarrollar una aplicación móvil conectada a Instagram, la cual tenga la capacidad de analizar una fotografía de un teléfono móvil y hacer una estimación del porcentaje de seguidores que darán *Likes* a la publicación. Se hace un primer diseño de la

aplicación móvil que se muestra en la Figura 6.1.

Figura 6.1: Diseño aplicación móvil

Apéndice A

Códigos

A continuación se presentan parte del código utilizado a lo largo del análisis.

A.1. Extracción y transformación de datos

Se muestra el código empleado para la extracción, unión y lectura de datos mediante R Studio.

A.1.1. Unión de archivos en una base única

```
1
2  setwd("C:/Users/marin/Desktop/TFM/aida") #leer carpeta de trabajo
3  fileNames <- Sys.glob("*.csv") #leer todos los csv de la carpeta de
       trabajo
4  fileNames                      #para mostrar todos los csv que ha
      leido
5  TOTAL <- NULL                  #crear la tabla TOTAL vacia
6
7
8  #CREAR BUCLE:
9  for (fileName in fileNames) {
10
```

```
11    print("---------")
12    print(fileName)
13    print("---------")
14    #Leemos todos los archivos CSV
15    datos <- read.csv(fileName,  stringsAsFactors=FALSE, header=TRUE,
          sep=",")
16    #crear la variable NOMBRE a partir del nombre del csv
17    NOMBRE <-tools::file_path_sans_ext(fileName)
18    #crear USUARIO cogiendo solo el nombre
19    USUARIO<-unlist(strsplit(NOMBRE,split = 'ExportData_user_',fixed=
          TRUE))[2]
20    #para quitar todo lo que haya despues de la ultima barra baja
21    USUARIO <- sub("_[^_]+$", "", USUARIO)
22    datos <-cbind(USUARIO=USUARIO,datos) #meter columna USUARIO
23
24    #unir cada tabla DATOS en una TOTAL
25    TOTAL<- rbind(TOTAL,datos)
26      }
27
28    #Usuario lo paso a tipo character
29    TOTAL$USUARIO <- as.character(TOTAL$USUARIO)
30
31    #Cambiamos el nombre de las variables
32    colnames(TOTAL)[2] <- "MEDIA_ID"
33    colnames(TOTAL)[3] <- "URL PUBLICACION"
34    colnames(TOTAL)[5] <- "FECHA"
35    colnames(TOTAL)[6] <- "TITULO"
36    colnames(TOTAL)[7] <- "COMENTARIOS"
37    colnames(TOTAL)[8] <- "LIKES"
38    #colnames(TOTAL)[10] <- "URL VIDEO"
39    colnames(TOTAL)[11] <- "URL IMAGEN MIN"
40    colnames(TOTAL)[12] <- "URL IMAGEN MAX"
```

```
41    #colnames(TOTAL)[13] <- "ID LOCALIZACION"
42    colnames(TOTAL)[14] <- "UBICACION"
43    colnames(TOTAL)[15] <- "URL UBICACION"
44    colnames(TOTAL)[16] <- "LATITUD"
45    colnames(TOTAL)[17] <- "LONGITUD"
46
47    #Elimino las variables date y video_views
48    TOTAL <- subset(TOTAL, select = -c(MEDIA_ID,date, video_views,
          location_id) )
49
50    TOTAL$TITULO <- gsub("\t", " ", TOTAL$TITULO)
51    TOTAL$TITULO <- gsub("\n", " ", TOTAL$TITULO)
52
53    write.table(TOTAL, file="C:/Users/marin/Desktop/TFM/R/Resultados/
          TOTALV2.txt", quote=F,sep="\t",  na="", col.names=T, row.names
           = F,fileEncoding = "ASCII")
```

A.1.2. Lectura de URL de las imágenes

```
1   #CODIGO PARA RECUPERAR EL ENLACE DE LAS FOTOS:
2
3   datos<-read.sas7bdat("C:/Users/marin/Desktop/TFM/Miner/trabajo.
        sas7bdat", debug=FALSE)
4   datos$URL_PUBLICACION <- as.character(datos$URL_PUBLICACION) #
        convertir URL de la publicacion en texto
5
6   #FUNCION PARA ENCONTRAR UNA PALABRA ENTRE DOS
7   getstr = function(mystring, initial.character, final.character)
8   {
9
10    # check that all 3 inputs are character variables
11    if (!is.character(mystring))
```

```
12      {
13        stop('The parent string must be a character variable.')
14      }
15
16      if (!is.character(initial.character))
17      {
18        stop('The initial character must be a character variable.')
19      }
20
21
22      if (!is.character(final.character))
23      {
24        stop('The final character must be a character variable.')
25      }
26
27
28
29      # pre-allocate a vector to store the extracted strings
30      snippet = rep(0, length(mystring))
31
32
33
34      for (i in 1:length(mystring))
35      {
36        # extract the initial position
37        initial.position = gregexpr(initial.character, mystring[i])
              [[1]][1] + 1
38
39        # extract the final position
40        final.position = gregexpr(final.character, mystring[i])
              [[1]][1] - 1
41
```

```
42      # extract the substring between the initial and final
            positions, inclusively
43      snippet[i] = substr(mystring[i], initial.position, final.
            position)
44    }
45
46    return(snippet)
47  }
48
49  #bucle que recorre todas las observaciones e inserta en una
        columna nueva el link de la foto
50  for (row in 1:nrow(datos)){
51    #prueba si puede acceder a la publicacion, y si no puede lo deja
          a NULL
52    tmp <- tryCatch(readLines(datos$URL_PUBLICACION[row], warn=F),
          error = function (e) NULL)
53
54    if (is.null(tmp)) {
55
56      datos<-datos[-c(row), ]#si no pudo acceder, borra esa
            observacion
57      next() # pasa a la siguiente iteracion del bucle
58    }
59    print(row)
60    html<- paste(readLines(datos$URL_PUBLICACION[row]),collapse="\n
          ")#lee todo el codigo de html de la web, y lo mete en la
          variable HTML
61    #print(html)
62    foto <- getstr(html,'content=\"https://instagram.fmad6-1.fna.
          fbcdn.net','.jpg\"')#Busca el link de la foto
63    #print(foto)
```

```
64      foto<-gsub("ontent=\"","",foto)#le quita al codifo de la foto el
            principio
65      foto<-paste(foto,".jpg",sep="")#le mete al link el ".jpg"
66      #print(url3)
67      datos$URL_IMAGEN_NUEVA[row]<-foto#mete el link de la foto en la
            variable URL_IMAGEN_NUEVA
68
69    }
```

A.1.3. Descarga y transformación de imágenes

```
1
2   #BUCLE PARA LEER LOS PIXELES DE LAS FOTOS
3   library(httr)
4   library(imager)
5
6
7   DATOSOK<-read.sas7bdat("C:/Users/marin/Desktop/TFM/R/Resultados/
        baseok.sas7bdat", debug=FALSE)
8   DATOSOK$URL_IMAGEN_NUEVA <- as.character(DATOSOK$URL_IMAGEN_NUEVA)
9
10  #lees el conjunto "datos"
11  #hacer un bucle que recorra las filas i
12
13
14  plot(thmb,main="Thumbnail")
15  thmb[,,1,1]
16
17  DATOSOK[1,1:22] <- thmb[,,1,1]
18  foto<-as.data.frame(thmb) %>% head(300)
19  foto$xyrgb <-paste( foto$x, foto$y, foto$cc,sep="")
20  foto <- subset(foto, select = -c(x,y,cc))
```

```
21
22  if (!require(reshape2)){install.packages('reshape2') library(
        reshape2)}
23
24  foto$aux <-1
25  mydt = dcast(foto,aux~xyrgb,value.var = "value")
26
27  #DATOSOK->8627
28  datoFoto<-data.frame()
29  #nrow(DATOSOK)
30  for (row in 1:nrow(DATOSOK)){
31
32    try <- tryCatch(GET(DATOSOK$URL_IMAGEN_NUEVA[row], warn=F),error
          = function (e) NULL)
33
34    if (is.null(try)) {
35
36      DATOSOK<-DATOSOK[-c(row), ]#si no pudo acceder, borra esa
            observacion
37      next() # pasa a la siguiente iteracion del bucle
38    }
39
40   GET(DATOSOK$URL_IMAGEN_NUEVA[row],write_disk("C:/Users/marin/
        Desktop/TFM/R/Resultados/Imagenes/imagen.jpg",overwrite = T))
        #lee la imagen de internet
41
42   try <- tryCatch(load.image("C:/Users/marin/Desktop/TFM/R/
        Resultados/Imagenes/imagen.jpg"),error = function (e) NULL)
43
44   if (is.null(try)) {
45
```

```
46      DATOSOK<-DATOSOK[-c(row), ]#si no pudo acceder, borra esa
            observacion
47      next() # pasa a la siguiente iteracion del bucle
48    }
49
50
51    im <- load.image("C:/Users/marin/Desktop/TFM/R/Resultados/
          Imagenes/imagen.jpg") #lee de tu ordenador a R
52    thmb <- resize(im,10,10) #comprime la imagen
53
54   foto<-as.data.frame(thmb) %>% head(300)
55   foto$xyrgb <-paste( foto$x, foto$y, foto$cc,sep="")
56   foto <- subset(foto, select = -c(x,y,cc))
57   tmp <- as.data.frame(t(foto[,1]))
58   colnames(tmp) <- foto$xyrgb
59   tmp$URL_INSTAGRAM <- DATOSOK$URL_PUBLICACION[row]
60
61  print(row)
62   datoFoto <- rbind(datoFoto, tmp)
63   }
64
65   write.table(datoFoto, file="C:/Users/marin/Desktop/TFM/R/
         Resultados/DatosFoto.txt", quote=F,sep="\t",  na="", col.names=
         T, row.names = F,fileEncoding = "ASCII")
66   write.table(DATOSOK, file="C:/Users/marin/Desktop/TFM/R/Resultados
         /DatosOK.txt", quote=F,sep="\t",  na="", col.names=T, row.names
          = F,fileEncoding = "ASCII")
67
68
69   as.vector(t(thmb))
70   as.vector(thmb)
71   thmb
```

```
72
73  m <- cbind(m, 8:14)[, c(1, 3, 2)] # insert a column
74
75
76  DATOSOK[1,ncol(DATOSOK):(ncol(DATOSOK)+100)]<-thmb[,,1,1]
77  datos[1,(ncol(datos)+101):(ncol(datos)+200)]<-thmb[,,1,2]
78  datos[1,(ncol(datos)+201):(ncol(datos)+300)]<-thmb[,,1,3]
79
80  #fin del bucle
```

A.2. Variables

Se muestran algunos de los códigos utilizados para crear las variables input mediante SAS BASE.

A.2.1. Creación y transformación de variables

```
1
2  /*Variables referentes a la fecha*/
3
4  DATA A; SET BASE1; FORMAT HORA time10. DIA \$10. FECHA1 DDMMYY. FS
       8. MES 8
5  HORA=TIMEPART(FECHA);
6  FECHA1=DATEPART(FECHA);RUN;
7
8  DATA A;SET A;
9  DIA=WEEKDAY(FECHA1)
10 MES=MONTH(FECHA1);RUN;
11
12 DATA A ;SET A; FORMAT DIASEM $10. ;
13 IF DIA=1 THEN DIASEM='DOMINGO';
14 IF DIA=2 THEN DIASEM='LUNES';
```

```
15 IF DIA=3 THEN DIASEM='MARTES';
16 IF DIA=4 THEN DIASEM='MIERCOLES';
17 IF DIA=5 THEN DIASEM='JUEVES';
18 IF DIA=6 THEN DIASEM='VIERNES';
19 IF DIA=7 THEN DIASEM='SABADO';RUN;
20
21 DATA A;SET A;FORMAT FS 8.;
22 IF DIASEM IN('SABADO','DOMINGO') THEN FS=1;
23 IF DIASEM NOT IN('SABADO','DOMINGO') THEN FS=0;RUN;
24
25 /*REAGRUPACION HORA*/
26 PROC SORT DATA=BASE;BY HORA;RUN;
27 DATA BASE;SET BASE;FORMAT HORA1 10.;HORA1=hour(HORA);RUN;
28
29 DATA BASE;SET BASE;format HORA2 $250.;
30 IF HORA1 IN (7,8,9,10)                                    THEN HORA2
      ='MANANA';
31 IF HORA1 IN (11,12)                                      THEN HORA2
      ='MEDIA MANANA';
32 IF HORA1 IN (13,14)                                      THEN HORA2
      ='MEDIO DIA';
33 IF HORA1 IN (15,16,17,18,19)                        THEN HORA2='TARDE';
34 IF HORA1 IN (20,21,22)                                   THEN HORA2
      ='NOCHE';
35 IF HORA1 IN (23,0,1)                                     THEN HORA2
      ='MEDIA NOCHE';
36 IF HORA1 IN (2,3,4,5,6)                                  THEN HORA2
      ='MADRUGADA';
37 RUN;
38
39 /*DIFERENCIA ENTRE PUBLICACIONES (MINUTOS)*/
40 PROC SORT DATA=A;BY USUARIO FECHA;RUN;
```

```
41 DATA A;SET A;BY USUARIO;DIF=dif(FECHA);if first.USUARIO then call
      missing(dif);RUN;
42 DATA A;SET A;DIF=DIF/60;RUN;
43
44
45 /* HASHTAG*/
46 DATA Y;SET Y; FORMAT HASHTAG 8.; HASHTAG=length(compress(upcase(
      TITULO))) - length(compress(compress(upcase(TITULO)),\"\#\"));
      RUN;
47
48 /*ETIQUETAS*/
49 DATA Y;SET Y; FORMAT ETIQUETAS 8.; ETIQUETAS=length(compress(upcase
      (TITULO))) - length(compress(compress(upcase(TITULO)),"@"));RUN;
50
51 /*LONGITUD DEL CAPTION*/
52 DATA Y;SET Y;FORMAT LETRA 8. PALABRA 8.; LETRA=LENGTH(compress(
      upcase(TITULO)));RUN;
53 DATA Y;SET Y; PALABRA=LENGTH(upcase(TITULO))-length(compress(upcase
      (TITULO))) +1;RUN;
54
55 /*VAR. BLANCO Y NEGRO*/
56 DATA A;SET A;FORMAT BYN 8.;BYN=0;
57 IF  $(_111=_112 AND _112=_113) and (_121=_122 AND _122=_123)  and (
      _1101=_1102 AND _1102=_1103) and
58 (_211=_212 AND _212=_213) and (_221=_222 AND _222=_223)  and (_2101
      =_2102 AND _2102=_2103)      and
59 (_311=_312 AND _312=_313) and (_321=_322 AND _322=_323)  and (_3101
      =_3102 AND _3102=_3103)      and
60 (_411=_412 AND _312=_313) and (_421=_422 AND _422=_423)  and (_4101
      =_4102 AND _4102=_4103)      and
61 (_511=_512 AND _312=_313) and (_521=_522 AND _522=_523)  and (_5101
      =_5102 AND _5102=_5103)      and
```

```
62 (_611=_612 AND _312=_313) and (_621=_622 AND _622=_623)  and (_6101
      =_6102 AND _6102=_6103)      and
63 (_711=_712 AND _312=_313) and (_721=_722 AND _722=_723)  and (_7101
      =_7102 AND _7102=_7103)      and
64 (_811=_812 AND _312=_313) and (_821=_822 AND _822=_823)  and (_8101
      =_8102 AND _8102=_8103)      and
65 (_951=_952 AND _952=_953) and (_951=_952 AND _952=_953)  and (_951=
      _952 AND _952=_953)  and
66 (_1011=_1012 AND _1012=_1013) and (_1021=_1022 AND _1022=_1023)
      and (_10101=_10102 AND _10102=_10103) and
67 (_131=_132 AND _132=_133) and (_241=_242 AND _242=_243) and  (_351=
      _352 AND _352=_353) and
68 (_461=_462 AND _462=_463) and (_571=_572 AND _572=_573) and (_681=
      _682 AND _682=_683) and
69 (_791=_792 AND _792=_793) and (_8101=_8102 AND _8102=_8103) and (
      _821=_822 AND _822=_823)  and (_8101=_8102 AND _8102=_8103) $
      THEN BYN=1;RUN;
70
71 /*TEMAS*/
72 DATA z;SET z;format felicitacion 8. positivo 8. deportes 8. musica
      8. moda 8. repost 8.;
73 felicitacion=0; positivo=0; deportes=0; musica=0; moda=0; repost=0;
74
75 /*FELICITACION*/
76 IF FIND (TITULO,'BIRTHDAY')^=0  THEN DO; FELICITACION=1;END;
77 IF FIND (TITULO,'HAPPY')^=0  THEN DO; FELICITACION=1;END;
78 IF FIND (TITULO,'CHRISTMAS')^=0  THEN DO; FELICITACION=1;END;
79 IF FIND (TITULO,'THANKSGIVING')^=0  THEN DO; FELICITACION=1;END;
80 IF FIND (TITULO,'ANNIVERSARY')^=0  THEN DO; FELICITACION=1;END;
81 IF FIND (TITULO,'DIWALI')^=0  THEN DO; FELICITACION=1;END;
82 IF FIND (TITULO,'CONGRATULATION')^=0  THEN DO; FELICITACION=1;END;
83 IF FIND (TITULO,'FELIZ')^=0  THEN DO; FELICITACION=1;END;
```

```
84  IF FIND (TITULO,'FELICIDADES')^=0  THEN DO; FELICITACION=1;END;
85
86  /*POSITIVO*/
87  IF FIND (TITULO,'LOVE')^=0  THEN DO; POSITIVO=1;END;
88  IF FIND (TITULO,'LOVING')^=0  THEN DO; POSITIVO=1;END;
89  IF FIND (TITULO,'DREAM')^=0  THEN DO; POSITIVO=1;END;
90  IF FIND (TITULO,'FREE')^=0  THEN DO; POSITIVO=1;END;
91  IF FIND (TITULO,'TRUST')^=0  THEN DO; POSITIVO=1;END;
92  IF FIND (TITULO,'THANKYOU')^=0  THEN DO; POSITIVO=1;END;
93  IF FIND (TITULO,'INSPIRATION')^=0  THEN DO; POSITIVO=1;END;
94  IF FIND (TITULO,'FRIEND')^=0  THEN DO; POSITIVO=1;END;
95  IF FIND (TITULO,'PEACE')^=0  THEN DO; POSITIVO=1;END;
96  IF FIND (TITULO,'READY')^=0  THEN DO; POSITIVO=1;END;
97  IF FIND (TITULO,'GOOD')^=0  THEN DO; POSITIVO=1;END;
98  IF FIND (TITULO,'LUCK')^=0  THEN DO; POSITIVO=1;END;
99  IF FIND (TITULO,'BEAUTIFUL')^=0  THEN DO; POSITIVO=1;END;
100 IF FIND (TITULO,'SOUL')^=0  THEN DO; POSITIVO=1;END;
101 IF FIND (TITULO,'COLOUR')^=0  THEN DO; POSITIVO=1;END;
102 IF FIND (TITULO,'HAPPINESS')^=0  THEN DO; POSITIVO=1;END;
103 IF FIND (TITULO,'GREAT')^=0  THEN DO; POSITIVO=1;END;
104 IF FIND (TITULO,'SUMMER')^=0  THEN DO; POSITIVO=1;END;
105 IF FIND (TITULO,'TALENT')^=0  THEN DO; POSITIVO=1;END;
106 IF FIND (TITULO,'YEAH')^=0  THEN DO; POSITIVO=1;END;
107 IF FIND (TITULO,'PAZ')^=0  THEN DO; POSITIVO=1;END;
108 IF FIND (TITULO,'AMOR')^=0  THEN DO; POSITIVO=1;END;
109 IF FIND (TITULO,'AMIGO')^=0  THEN DO; POSITIVO=1;END;
110 IF FIND (TITULO,'LOVELY')^=0  THEN DO; POSITIVO=1;END;
111 IF FIND (TITULO,'INSPIRATION')^=0  THEN DO; POSITIVO=1;END;
112 IF FIND (TITULO,'WONDERFUL')^=0  THEN DO; POSITIVO=1;END;
113 IF FIND (TITULO,'HOLIDAY')^=0  THEN DO; POSITIVO=1;END;
114 IF FIND (TITULO,'JOY')^=0  THEN DO; POSITIVO=1;END;
115
```

```
116 /*DEPORTES*/
117 IF FIND (TITULO,'SPORT')^=0  THEN DO; DEPORTES=1;END;
118 IF FIND (TITULO,'WIN')^=0  THEN DO; DEPORTES=1;END;
119 IF FIND (TITULO,'GAMES')^=0  THEN DO; DEPORTES=1;END;
120 IF FIND (TITULO,'TRAINING')^=0  THEN DO; DEPORTES=1;END;
121 IF FIND (TITULO,'CHAMPION')^=0  THEN DO; DEPORTES=1;END;
122 IF FIND (TITULO,'HALAMADRID')^=0  THEN DO; DEPORTES=1;END;
123 IF FIND (TITULO,'PITCH')^=0  THEN DO; DEPORTES=1;END;
124 IF FIND (TITULO,'FOOTBALL')^=0  THEN DO; DEPORTES=1;END;
125 IF FIND (TITULO,'MATCH')^=0  THEN DO; DEPORTES=1;END;
126 IF FIND (TITULO,'SOCCER')^=0  THEN DO; DEPORTES=1;END;
127 IF FIND (TITULO,'PARTIDO')^=0  THEN DO; DEPORTES=1;END;
128 IF FIND (TITULO,'ATLETICO')^=0  THEN DO; DEPORTES=1;END;
129 IF FIND (TITULO,'REFEREE')^=0  THEN DO; DEPORTES=1;END;
130 futbol
131
132 /*MODA*/
133 IF FIND (TITULO,'FASHION')^=0  THEN DO; MODA=1;END;
134 IF FIND (TITULO,'GIORGIOARMANI')^=0  THEN DO; MODA=1;END;
135 IF FIND (TITULO,'VERSACE')^=0  THEN DO; MODA=1;END;
136 IF FIND (TITULO,'VOGUE')^=0  THEN DO; MODA=1;END;
137 IF FIND (TITULO,'OUTFIT')^=0  THEN DO; MODA=1;END;
138 IF FIND (TITULO,'DRESS')^=0  THEN DO; MODA=1;END;
139 IF FIND (TITULO,'SPARKLE')^=0  THEN DO; MODA=1;END;
140 IF FIND (TITULO,'JACKET')^=0  THEN DO; MODA=1;END;
141 IF FIND (TITULO,'GLAMOUR')^=0  THEN DO; MODA=1;END;
142 IF FIND (TITULO,'BURBERRY')^=0  THEN DO; MODA=1;END;
143 IF FIND (TITULO,'GABANA')^=0  THEN DO; MODA=1;END;
144
145 /*MUSICA*/
146 IF FIND (TITULO,'MUSIC')^=0  THEN DO; MUSICA=1;END;
147 IF FIND (TITULO,'DANCE')^=0  THEN DO; MUSICA=1;END;
```

```
148 IF FIND (TITULO,'STUDIO')^=0  THEN DO; MUSICA=1;END;
149 IF FIND (TITULO,'CONCERT')^=0  THEN DO; MUSICA=1;END;
150 IF FIND (TITULO,'SONG')^=0  THEN DO; MUSICA=1;END;
151
152 /*REPOST*/
153 IF FIND (TITULO,'REPOST')^=0  THEN DO; REPOST=1;END;
154 RUN;
```

A.2.2. Componentes Principales

```
1
2  DATA Y;SET Z;KEEP  URL_PUBLICACION _111 --_10103 ;RUN;
3
4  proc princomp DATA=Y n=33 plots=ALL outstat=EST1 out=PROY;
5  VAR  _111 -- _10103 ;
6  ID  URL_PUBLICACION ;
7  RUN;
8
9  DATA PROY;SET PROY;DROP  _111 -- _10103 ;RUN;
10
11 PROC SORT DATA=PROY;BY  URL_PUBLICACION ;RUN;
12 PROC SORT DATA=Z;BY  URL_PUBLICACION ;RUN;
13
14 DATA UNION;MERGE Z(IN=A) PROY(IN=B);BY  URL_PUBLICACION ;RUN;
15 DATA UNION;SET UNION;DROP  _111 -- _10103 ;RUN;
```

A.3. Modelos

Regresión Lineal

```
1
```

```
%macro cruzada  (archivo=,vardepen=,conti=,categor=,ngrupos=,
   sinicio=,sfinal=);
data final;run;
%do semilla=&sinicio %to &sfinal;
        data dos;set &archivo;u=ranuni(&semilla);
        proc sort data=dos;by u;run;
        data dos;
        retain grupo 1;
        set dos nobs=nume;
        if  \_n\_>grupo*nume/&ngrupos then grupo=grupo+1;
        run;
        data fantasma;run;
        %do exclu=1 %to &ngrupos;
                data tres;set dos;if grupo ne &exclu then vardep=&
                   vardepen;
                proc glm data=tres noprint;
                %if &categor ne %then %do;class &categor;model
                   vardep=&conti &categor;%end;
                %else %do;model vardep=&conti;%end;
                output out=sal p=predi;run;
                data sal;set sal;resi2=(&vardepen-predi)**2;if
                   grupo=&exclu then output;run;
                data fantasma;set fantasma sal;run;
        %end;
        proc means data=fantasma sum noprint;var resi2;
        output out=sumaresi sum=suma mean=media;
        run;
        data sumaresi;set sumaresi;semilla=&semilla;
        data final (keep=suma media semilla);set final sumaresi;if
            suma=. then delete;run;
%end;
proc print data=final;run;
```

```
29 %mend;
```

Red Neuronal

```
1  %macro cruzadaneural (archivo=,vardepen=,conti=,categor=,ngrupos=,
       sinicio=,sfinal=,ocultos=,algo=,acti=,early=,directorio=);
2
3  proc printto print="&directorio\\basura.txt";
4  data final;run;
5  %do semilla=&sinicio %to &sfinal;
6          data dos;set &archivo;u=ranuni(&semilla);
7          proc sort data=dos;by u;run;
8          data dos (drop=nume);
9           retain grupo 1;
10          set dos nobs=nume;
11          if  _n_>grupo*nume/&ngrupos then grupo=grupo+1;
12         run;
13         data fantasma;run;
14         %do exclu=1 %to &ngrupos;
15                 data trestr tresval;
16                 set dos;if grupo ne &exclu then output trestr;else
                       output tresval;
17                 PROC DMDB DATA=trestr dmdbcat=catatres;
18                 target &vardepen;
19                 var &vardepen &conti;
20                 %if &categor ne %then %do;class &categor;%end;
21                 run;
22                 proc neural data=trestr dmdbcat=catatres random=789
                       ;
23                 input &conti;
24                 %if &categor ne %then %do;input &categor /level=
                       nominal;%end;
```

```
25                  target &vardepen;
26                  hidden &ocultos /act=&acti;
27
28          %if &early ne %then %do;
29                  nloptions maxiter=&early;
30                  netoptions randist=normal ranscale=0.1 random
                        =15115;%end;
31
32          %if &early ne %then %do;
33                  train maxiter=&early outest=mlpest technique=&algo
                        ;%end;
34                  %else %do;train maxiter=100 outest=mlpest technique
                        =&algo;%end;
35                  score data=tresval role=valid out=sal ;
36                  run;
37                  data sal;set sal;resi2= (p_&vardepen-&vardepen)**2;
                        run;
38                  data fantasma;set fantasma sal;run;
39          %end;
40          proc means data=fantasma sum noprint;var resi2;
41          output out=sumaresi sum=suma mean=media;
42          run;
43          data sumaresi;set sumaresi;semilla=&semilla;
44          data final (keep=suma media semilla);set final sumaresi;if
               suma=. then delete;run;
45          %end;
46  proc printto;run;
47  proc print data=final;run;
48  %mend;
```

Random Forest

```
%macro cruzadarandomforest(archivo=,vardep=,listconti=,listcategor
   =,maxtrees=,variables=,porcenbag=,maxbranch=,tamhoja=,maxdepth=,
   pvalor=,ngrupos=,sinicio=,sfinal=);

data final;run;
%do semilla=&sinicio %to &sfinal;

data dos;set &archivo;u=ranuni(&semilla);
        proc sort data=dos;by u;run;
        data dos ;
        retain grupo 1;
        set dos nobs=nume;
        if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
        run;


data fantasma;run;

%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardep;


ods listing close;
proc hpforest data=tres
maxtrees=&maxtrees
vars_to_try=&variables
trainfraction=&porcenbag
leafsize=&tamhoja
maxdepth=&maxdepth
alpha=&pvalor
exhaustive=5000
```

```
30 missing=useinsearch ;
31 target &vardep/level=interval;
32 input &listconti/level=interval;
33 %if (&listcategor ne) %then %do;
34         input &listcategor/level=nominal;
35         %end;
36 score out=sal;
37 run;
38 ods listing ;
39
40 data sal;set sal;resi2=(p_&vardep-&vardep)**2;run;
41 data fantasma;set fantasma sal;run;
42
43 %end;
44
45         proc means data=fantasma sum noprint;var resi2;
46         output out=sumaresi sum=suma mean=media;
47         run;
48         data sumaresi;set sumaresi;semilla=&semilla;
49         data final (keep=suma media semilla);set final sumaresi;if
              suma=. then delete;run;
50
51
52 %end;
53
54 proc print data=final;run;
55
56 %mend;
```

Gradient Boosting

```
%macro cruzadaboosting(archivo=,vardep=,listconti=,listcategor=,
   porcenbag=,maxbranch=,tamhoja=,maxdepth=,pvalor=,ngrupos=,
   sinicio=,sfinal=,shrink=,iterations=);

data final;run;
%do semilla=&sinicio %to &sfinal;

data dos;set &archivo;u=ranuni(&semilla);
        proc sort data=dos;by u;run;
        data dos ;
        retain grupo 1;
        set dos nobs=nume;
        if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
        run;


data fantasma;run;

%do exclu=1 %to &ngrupos;
data tres;set dos;if grupo ne &exclu then vardep=&vardep;


ods listing close;

proc treeboost
data=tres
shrinkage=&shrink
maxbranch=&maxbranch
maxdepth=&maxdepth
iterations=&iterations
leafsize=&tamhoja;
        %if (&listcategor ne) %then %do;
```

```
31          input &listcategor/level=nominal;
32          %end;
33          input &listconti/level=interval;
34          target &vardep /level=interval;
35  score data=tres out=sal;
36  run;
37  ods listing ;
38
39  data sal;set sal;resi2=(p_&vardep-&vardep)**2;run;
40  data fantasma;set fantasma sal;run;
41
42
43  %end;
44
45
46          proc means data=fantasma sum noprint;var resi2;
47          output out=sumaresi sum=suma mean=media;
48          run;
49          data sumaresi;set sumaresi;semilla=&semilla;
50          data final (keep=suma media semilla);set final sumaresi;if
                suma=. then delete;run;
51
52  %end;
53
54  proc print data=final;run;
55
56  %mend;
```

Ensamble

```
1
```

```
%macro cruzadastack (archivo=,vardepen=,listcategor=,listconti=,
   ngrupos=,seminicio=,semifinal=,nodos=);

data final;run;
proc printto print='C:\Users\marin\Desktop\TFM\Miner\Apoyo\ca.txt'
   log='C:\Users\marin\Desktop\TFM\Miner\Apoyo\loga.txt';run;
%do semilla=&seminicio %to &semifinal;
data dos;set &archivo;u=ranuni(&semilla);
proc sort data=dos;by u;run;
data dos (drop=nume);
retain grupo 1;
set dos nobs=nume;
if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
run;

data fantasma;run;

%do exclu=1 %to &ngrupos;

data tres;set dos;if grupo ne &exclu then vardep=&vardepen*1;run;

/* REGRESION */
proc glm data=tres noprint;
class BYN       IDIOMA  MES     SECTOR DIA;
model PORLIKE=DIF       ETIQUETAS       FOLLOWERS2      HASHTAG
   MEDIA Prin1 Prin8 LETRA PALABRA Prin3 Prin18 BYN        IDIOMA
   MES     SECTOR DIA;
output out=saco p=predi;
;run;

data sal1 ;set saco;predi1=predi;run;

```

```
30 /*RED */
31
32 PROC DMDB DATA=tres dmdbcat=catatres;
33 target &vardepen;
34 var &listconti &vardepen;
35 class &listcategor;
36 run;
37
38 proc neural data=tres dmdbcat=catatres;
39 input &listconti / id=i;
40 input &listcategor / level=nominal;
41 target &vardepen / id=o;
42 hidden 10 / id=h act=GAU;
43 nloptions maxiter=100;
44 netoptions randist=normal ranscale=0.1 random=15115;
45 train maxiter=25 outest=mlpest estiter=1 technique=LEVMAR;
46 score data=tres out=salred ;
47 run;
48
49 data sal2 (keep=&vardepen predi2 grupo vardep);set salred;predi2=p_
      &vardepen;run;
50
51
52 /*RANDOM FOREST*/
53 proc hpforest data=tres
54 maxtrees=160
55 vars_to_try=51
56 trainfraction=0.7
57 leafsize=50
58 maxdepth=15
59 alpha=0.1
60 exhaustive=5000
```

```
61 missing=useinsearch ;
62 target &vardepen/level=interval;
63 input &listconti/level=interval;
64 %if (&listcategor ne) %then %do;
65         input &listcategor/level=nominal;
66         %end;
67 score out=sal;
68 run;
69 data sal3 (keep=&vardepen predi3 grupo vardep);set sal;predi3=p_&
      vardepen;run;
70
71
72 /*GRADIENT BOOSTING */
73 proc treeboost
74 data=tres
75 shrinkage=0.05
76 maxbranch=2
77 maxdepth=20
78 iterations=60
79 leafsize=50;
80         %if (&listcategor ne) %then %do;
81         input &listcategor/level=nominal;
82         %end;
83         input &listconti/level=interval;
84         target &vardepen /level=interval;
85 score data=tres out=sal;
86 run;
87
88 data sal4 (keep=&vardepen predi4 grupo vardep);set sal;predi4=p_&
      vardepen;run;
89
90 data unionsal (drop=ygorro);merge sal1 sal2 sal3 sal4;
```

```
91  predi5=(predi1+predi2)/2; /* RED -LOG */
92  predi6=(predi1+predi3)/2;/* RED -RFOR */
93  predi7=(predi1+predi4)/2;/* RED -BOOST*/
94  predi8=(predi2+predi3)/2;/* LOG-RFOR */
95  predi9=(predi2+predi4)/2;/* LOG-BOOST */
96  predi10=(predi3+predi4)/2;/* RFOR-BOOST */
97  predi11=(predi1+predi2+predi3)/3;/* RED -LOG-RFOR */
98  predi12=(predi1+predi2+predi4)/3;/* RED -LOG-BOOST*/
99  predi13=(predi1+predi3+predi4)/3;/* RED -RFOR-BOOST*/
100 predi14=(predi2+predi3+predi4)/3;/* LOG-RFOR-BOOST*/
101 predi15=(predi1+predi2+predi3+predi4)/4;/* RED-LOG-RFOR-BOOST*/
102 run;
103
104 data salfin (keep=&vardepen vardep predi1-predi15 grupo);set
       unionsal;if grupo=&exclu then output;run;
105
106 data salbos (drop=i);
107 array predi{15};
108 array ase{15};
109 set salfin;
110 do i=1 to 15;
111 ase{i}=(predi{i}-&vardepen)**2;
112 end;
113 run;
114
115 data fantasma;set fantasma salbos;run;
116
117 %end;
118
119 proc means data=fantasma noprint;var ase1-ase15;
120 output out=mediaresi mean=ase1-ase15;
121 run;
```

```
122 data mediaresi;set mediaresi;semilla=&semilla;run;
123 data final (keep=ase1-ase15 semilla);set final mediaresi;if ASE1=.
       then delete;run;
124 %end;
125 proc printto; run;
126 proc print data=final;run;
127 %mend;
```

Glosario

H

Hashtags Un hashtag consta de palabras o frases (sin espacios) precedidas de un signo almohadilla #. Esto permite nombrar un tema particular, y de esta manera que las palabras queden agrupadas bajo una misma etiqueta. Ejemplo: #SocialMediaMarketing #Twitter #Blogging . 3

I

Influencers Persona que cuenta con cierta credibilidad sobre un tema concreto, y por su presencia e influencia en redes sociales puede llegar a convertirse en un prescriptor para una marca.. 2

K

Kodak Instamatic La Kodak Camera era una caja con una lente muy simple, una manivela para avanzar un rollo de película con 100 exposiciones y un botón de disparo. Los modelos Instamatic, se caracterizaban por no usar rollos de película sino cartuchos que bastaba introducir en la cámara para empezar a fotografiar.. 1

P

Polaroid Es una cámara que utiliza un tipo de película que permite crear un positivo directo, revelándolo "al momento", después de hacer la foto; normalmente en formato cuadrado.. 1

Lista de acrónimos

A

ACP Análisis de Componentes Principales. 29, 30

ASE Average Squared Error. 33, 34, 40, 44, 45

H

HTML HyperText Markup Language. 5

U

URL Uniform Resource Locator. 5, 61

Bibliografía

[1] Varios autores, "Instagram." https://es.wikipedia.org/wiki/Instagram, 2018.

[2] Varios autores, "Qué es y cómo funciona un influencer." https://blogginzenith.zenithmedia.es/que-es-y-como-funciona-un-influencer-diczionario/, 2015.

[3] Varios autores, "Modos de color: Rgb, cmyk y srgb." http://www.fotonostra.com/grafico/rgb.htm, 2015.

[4] Asociación AEIPRO, *Metodologías para la realización de proyectos de Data Mining*, 2015.

[5] D. Montoro Cazorlal, "Regresión lineal simple," 2015.

[6] D. J. Matich, *Redes Neuronales: Conceptos Básicos*. PhD thesis, Universidad Tecnológica Nacional, Facultad Regional Rosario, 2001.

[7] F. Cánovas-García, F. Alonso-Sarría, and F. Gomariz-Castillo, "Modificación del algoritmo random forest para su empleo en clasificación de imágenes de teledetección," *Aplicaciones de las Tecnologías de la Información Geográfica (TIG) para el desarrollo económico sostenible*, 2016.

[8] R. E. Lopez Briega, "Boosting en machine learning," 2017.

[9] F. Sancho Caparrini, "Métodos combinados de aprendizaje," 2017.

[10] M. Sewell, "Ensemble methods," *Relatorio Técnico RN/11/02, University College London Departament os Computer Science*, 2011.

[11] J. Portela, *Métodos de Ensamble*. Facultad de Estudios Estadísticos, Universidad Complutense de Madrid: Apuntes de la asignatura Redes Neuronales, 2016.

[12] L. C. Molina Félix, "Mineria de textos o text mining." `http://textmining.galeon.com/`, 2015.

[13] M. Gurrea, "Análisis de componentes principales," *Proyecto e-Math Financiado por la Secretaría de Estado de Educación y Universidades (MECD)*, 2000.

[14] J. Portela, "Introducción," 2016.

[15] Varios autores, "Error cuadrático medio." `https://es.wikipedia.org/wiki/Error_cuadr%C3%A1tico_medio`, 2017.

[16] Varios autores, "Variable importance in neural networks." `https://www.r-bloggers.com/variable-importance-in-neural-networks/`, 2013.