UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN

MENCIÓN EN TELEMÁTICA

Plataformas de publicidad en Internet: Google Ads y Criteo. Análisis, visualización de datos y obtención de modelos predictivos.

Autor:

Javier Olmedo Gutiérrez

Tutor:

Ignacio de Miguel Jiménez

Valladolid, septiembre de 2019

Resumen

Gracias al uso que damos a las nuevas tecnologías ha surgido una nueva forma de publicidad, la publicidad en Internet. Hoy en día, es fácil ver por la calle a personas empleando smartphones en todo momento, usando tabletas u ordenadores portátiles sentados en una cafetería, etc. Es decir, en la actualidad podemos estar conectados a Internet en todo momento.

Con esta idea, surgió la publicidad digital, permitiendo llegar a los usuarios en cualquier momento y de formas muy diversas, adaptando la publicidad a un usuario y sus circunstancias en concreto.

El principal objetivo de este TFG consiste en el desarrollo de un modelo predictivo mediante aprendizaje supervisado, capaz de determinar si un anuncio en Internet tendrá éxito, esto es, si tras visualizar un anuncio que le aparece en pantalla, el usuario hará clic y accederá a él.

En primer lugar, mostraremos las principales plataformas publicitarias disponibles en el mercado digital. Seguidamente se estudiarán las posibilidades que ofrecen los distintos modelos de aprendizaje supervisado. Después, se realizará un análisis y visualización de los datos disponibles con el fin de facilitar el mejor diseño posible del modelo predictivo que se llevará a cabo. Finalmente, se presentarán los resultados obtenidos, así como las conclusiones a las que se han llegado y las posibles líneas futuras.

Palabras clave: publicidad digital, aprendizaje supervisado, modelo predictivo, clic.

Abstract

Thanks to the use we give to new technologies, a new form of advertising emerged, advertising on the Internet. Nowadays, it is always easy to see people on the street using smartphones, using tablets or laptops in a cafeteria, etc. That is, at present we can always be connected to the Internet.

With this idea, digital advertising emerged, we were able to adapt advertising to a user and their specific circumstances.

The main objective of this TFG is the development of a predictive model through supervised learning, which can determine the success of an advertisement on the Internet, this, if after the display of the ad on the screen, the user clicks and accesses it.

First, we will show the main advertising platforms available in the digital market. Next, the possibilities offered by the different models of supervised learning will be studied. Afterwards, an analysis and a visualization of the available data will be shown. Finally, the results will be presented.

Keywords: digital advertising, supervised learning, predictive model, click

Agradecimientos

En primer lugar, me gustaría mostrar mi agradecimiento a Ignacio de Miguel Jiménez, por su ayuda y asesoramiento a lo largo de la elaboración de este Trabajo Fin de Grado.

También mencionar el apoyo tanto del Grupo de Comunicaciones Ópticas de la Universidad de Valladolid como de Luce IT por permitirme trabajar en este proyecto.

A mis amigos y mi pareja, que en todo momento han sido un apoyo, siempre dispuestos a echarme una mano cuando ha sido necesario.

Finalmente, a mis padres y a mi hermana, por la ayuda y consejos a lo largo de todos estos años, y sin los cuales no hubiera sido posible llegar hasta aquí.

Índice general

Índice de figuras

Índice de tablas

1. Introducción.

En este capítulo se presenta el tema a tratar en este Trabajo Fin de Grado, las motivaciones que nos han llevado a su desarrollo y los objetivos que se buscan alcanzar con su realización. Del mismo modo, se presentan las fases y medios empleados para alcanzar dichos objetivos.

1.1. Motivación y objetivos

La publicidad se define según la RAE como "*Divulgación de noticias o anuncios de carácter comercial para atraer a posibles compradores, espectadores, usuarios, etc.*" [1].

A lo largo de la historia la publicidad ha ido evolucionando con la aparición de las nuevas tecnologías. Podemos considerar que la publicidad surgió en Egipto, con la elaboración de papiros publicitarios y, posteriormente continuaría con el pueblo fenicio, Grecia y Roma.

Todo cambiaría en 1440, cuando Johannes Gutenberg inventa la imprenta, permitiendo, por primera vez en la historia, la difusión masiva de textos impresos, consolidando la publicidad como una importante forma de comunicación.

Otro hito importante en la publicidad se produce en la década de 1920, con la aparición de los anuncios radiofónicos. A lo que le seguiría, en el año 1941, la publicidad televisiva [2].

Finalmente, con la llegada de Internet y las nuevas tecnologías, la publicidad sufre una nueva evolución, dando lugar a la publicidad digital. Esta nueva versión de publicidad se basa en el uso de las técnicas tradicionales, aplicadas en entornos digitales. La publicidad digital pretende dirigirse a la mayor población posible, tratando a cada individuo de forma personal, adaptándose a sus intereses y necesidades. [3].

En este nuevo ámbito publicitario surgió en el año 2000 Google AdWords, predecesor de Google Ads, basado inicialmente en anuncios de texto en el buscador de Google. Posteriormente, esta publicidad se expandiría en anuncios de video y otros tipos. Finalmente, en 2018 se comenzó a introducir aprendizaje automático en las campañas publicitarias, y en julio de este mismo año, Google AdWords pasó a llamarse Google Ads [4].

Por otro lado, Criteo es una empresa fundada en París en 2005 que según la propia empresa "*se ha convertido en un líder mundial en Commerce Marketing. Su éxito reside en la tecnología de Machine Learning, datos y resultados a escala y un ROI medible para nuestros clientes*" [5].

El principal objetivo de este TFG será crear un modelo predictivo, empleando en aprendizaje supervisado, capaz de determinar el posible éxito de un anuncio, basándonos en datos obtenidos anteriormente. Con el fin de alcanzar este objetivo principal, podemos añadir unos objetivos intermedios:

- Obtención de un dataset sobre el cual realizar nuestro estudio.
- Análisis y visualización del dataset para su mejor comprensión y facilitar la selección de las características más importantes de los datos empleados para la creación de nuestro modelo.

1.2. Fases

Con el objetivo marcado, se establecen varias fases con el fin de ayudar a su cumplimiento:

1. Adquisición de conocimientos relativos al aprendizaje automático y plataformas de publicidad digital. Búsqueda de información y bibliografía sobre los temas que se van a tratar en desarrollo de este TFG.

2. Búsqueda de dataset relacionados con las plataformas de publicidad en Internet estudiadas.

    a. Dataset de Google AdWords (predecesor de Google Ads)
    b. Dataset de Criteo.

3. Análisis y preprocesado del dataset. Con el fin de diseñar modelos predictivos, en primer lugar, debemos acondicionar nuestro dataset con un preprocesado que permitirá realizar las siguientes fases.

4. Propuesta de modelos predictivos. Con el dataset ya preparado, se procede a realizar distintos modelos, con el fin de poder compararlos y tomar la mejor decisión para alcanzar nuestro objetivo.

5. Comprobación de resultados. Después de la obtención de un nuevo modelo, este será estudiado, analizando los resultados obtenidos y comparándolos con los obtenidos en otros modelos. Estos dos últimos puntos, son iterativos, ya que tras la obtención de un modelo se estudiarán sus resultados y se procederá a obtener otro modelo hasta alcanzar el objetivo planteado.

1.3 Medios disponibles

Para llevar a cabo este proyecto se ha empleado Python como lenguaje de programación. Con el fin de facilitar el trabajo con este lenguaje emplearemos el entorno Jupyter.

El Proyecto Jupyter es un proyecto de código abierto, sin fines de lucro, nacido en 2014 del Proyecto IPython, para respaldar la ciencia de datos iterativa y la computación científica en todos los lenguajes de programación. Jupyter se desarrolla abiertamente en GitHub [6].

Jupyter Notebook es una aplicación web de código abierto que permite crear y compartir documentos que incluyan "código vivo", ecuaciones, gráficas y texto explicativo. Los principales usos que proporciona Jupyter Notebook son limpieza y transformación de datos, simulación numérica, visualización de datos y aprendizaje automático [6].

El tratamiento de gran cantidad de datos y el uso de aprendizaje supervisado en este proyecto nos ha llevado al uso, entre otras, de las siguientes librerías de Python:

- *Pandas*: librería de código abierto con licencia BSD. Proporciona estructuras de datos de alto rendimiento y herramientas para el análisis de datos en el lenguaje de programación Python [7].
- *NumPy*: al igual que pandas, es una librería bajo licencia BSD, lo que permite su reutilización con pocas restricciones. Es un paquete fundamental para la computación científica ya que permite trabajar con matrices N-dimensionales, posee funciones sofisticadas y herramientas de integración de código C/C++ y Fortran. También permite herramientas para álgebra lineal, transformada de Fourier y permite trabajar con números aleatorios [8].
- *Matplotlib*: librería de Python para la representación en 2D de figuras de alta calidad en múltiples formatos. *Matplotlib* puede ser empleado en scripts de Python, en *shells* de Python e IPython, en Jupyter Notebook, etc. [9].
- *Scikit-learn*: librería de código abierto, con licencia BSD, diseñada para la realización de aprendizaje automático en Python. Librería construida en *NumPy*, *SciPy* y *Matplotlib* [10].

1.4 Estructura del documento

Esta memoria está estructurada en seis capítulos, en los que se abordan las distintas fases seguidas para alcanzar el objetivo de este TFG.

En este primer capítulo, se ha realizado una breve introducción al tema principal de este trabajo, el marketing digital, explicando la motivación que nos ha llevado a

realizarlo. También se han explicado los objetivos que pretendemos alcanzar, así como las fases y medios empleados para cumplir dicho objetivo.

En el segundo capítulo, se aborda el concepto de marketing online, haciendo hincapié en dos de las principales plataformas de marketing digital, Google Ads y Criteo. Profundizaremos en dichas plataformas con el fin de comprender su funcionamiento interno, para comprender los principales factores que afectan a la calidad de los anuncios. Finalmente, se muestran los dataset con los que se va a trabajar.

El tercer capítulo está centrado en el aprendizaje supervisado, comenzando con una introducción y breve historia. A continuación, se explican los principales conceptos y modelos que se emplearán a lo largo de este proyecto.

En el cuarto capítulo se muestra el análisis realizado sobre los dataset obtenidos, tanto de Google Ads como de Criteo, facilitando su compresión con la visualización de distintas gráficas.

El quinto capítulo incluye los modelos empleados, trabajando únicamente con el dataset de Criteo, así como el estudio de los resultados obtenidos.

Finalmente, en el sexto capítulo se extraen las conclusiones de este TFG y se muestran posibles ideas para continuar trabando con el mismo en el futuro.

2. Publicidad en Internet.

El marketing engloba el conjunto de principios y prácticas cuyo objetivo es potenciar la actividad comercial, centrándose en los procedimientos y recursos para alcanzar este fin.

El marketing digital surge con el auge de las nuevas tecnologías y la forma en las que los usuarios las usamos, y consiste en emplear las técnicas del marketing tradicional en entornos digitales.

El marketing pretende crear un entorno en el que empresa y consumidor estén al mismo nivel, proporcionando comodidad y seguridad al cliente, con el fin de conocerle mejor y personalizar el modo de tratarle. Por otro lado, el marketing digital pretende llegar al mayor número de población posible en el entorno digital, pero continuando con ese trato personal.

2.1 Google Ads

Google Ads es una plataforma publicitaria en Internet que sucede a Google AdWords como sistema de pago por clic (CPC, de las siglas en ingles *Cost Per Click*). Esta herramienta permite elaborar y publicar anuncios a lo largo de toda la Red de Búsqueda y Visualización de Google, mejorando la visibilidad de las compañías [11].

Su principal virtud es el gran alcance que posee a través de todo Internet ya que actualmente copa en torno al 25% de los clics realizados por los usuarios en cualquier tipo de búsqueda.

La principal diferencia entre Google Ads y Google AdWords es que esta nueva actualización emplea Machine Learning para ofrecer mejoras en los anuncios existentes basándose en datos de la audiencia y de su interacción.

Google Ads ofrece numerosas ventajas a los comerciantes que quieran emplear sus servicios como, por ejemplo:

- Sistema de orientación para mostrar los anuncios adecuados a las personas adecuadas, en el lugar y momento adecuados. Para ello recurre a palabras claves, datos demográficos y geográficos, etc.
- Control total de los gastos, pudiendo seleccionar un presupuesto diario, semanal o mensual.
- Mejora del rendimiento en los anuncios publicitarios emitidos, ya que gracias a sus estadísticas se pueden observar las impresiones que generan sus anuncios, los clics que realizan los usuarios, e incluso las ventas generadas como resultado

directo de la publicidad. Gracias a estas estadísticas se pueden modificar las características de los anuncios para obtener un mayor rendimiento.

2.1.1 Historia de Google Ads

A finales de 1999 Google comenzó a vender anuncios para ser mostrados en sus resultados de búsqueda, empleando el método de cobro CPM (coste por mil impresiones), es decir, cobraba cada vez que mostraba un anuncio 1.000 veces, sin importar que las visitas que generaba. En estos comienzos, Google sólo ofrecía anuncios de texto con el fin de no ser un buscador incómodo en una época en la que generalmente se empleaban banners molestos que no interesaban a los usuarios [12].

El método para vender esta publicidad era directo, es decir, un anunciante se ponía en contacto con un agente de ventas de Google y podía contratar el número de impresiones deseadas a cambio de un precio fijado dependiendo únicamente del sector en el que trabajara, lo cual no era demasiado exitoso. Poco después, en 2002, Google incluyó el PPC (pago por clic), mejorando el sistema, pero siempre teniendo en cuanta un compromiso entre anunciantes y usuarios ya que Google tenía en cuenta que, si sus búsquedas comenzaban a dar como resultados anuncios no deseados o de poco interés, el buscador iría perdiendo la cuota de mercado que había obtenido desde sus inicios.

Finalmente, en Julio de 2018, la plataforma cambió su nombre, pasando a llamarse Google Ads, recibiendo una nueva interfaz e introduciendo el uso de *machine Learning* para mejorar el rendimiento de sus campañas.

2.1.2 Funcionamiento de Google Ads

En primer lugar, debemos indicar el lugar donde los potenciales clientes pueden encontrarse con nuestros anuncios. Los anuncios generados con esta plataforma podrán aparecer a lo largo de la amplia red de Google, que podemos dividir en dos partes [4, 11]:

- Red de búsqueda: anuncios de texto que aparecen al realizar una búsqueda. Esta red abarca Google Search, Google Shopping, Google Maps, Google Play y otros sitios de búsqueda asociados con Google.
- Red de visualización: anuncios con formato enriquecido (imagen, vídeo, etc.) que pueden aparecer en lugares como YouTube, Blogger, Gmail y otros sitios web asociados con Google.

Google Ads nos ofrece varios formatos para realizar los anuncios, se deberá seleccionar el más adecuado para el contenido que se ofrece y su posible audiencia:

- Anuncio de texto: es la publicidad más empleada, en parte, por su sencillez. Este tipo de anuncio debe incluir al menos un título, la URL del anunciante y una breve descripción. También puede incluir campos de ruta y diferentes extensiones para ofrecer más información. Aparecen en la red de búsqueda de Google y asociados.
    - Título: es lo más llamativo, debe incluir alguna palabra clave para llamar la atención del cliente potencial.
    - URL: dirección de la página web del anunciante. Sólo incluye la ruta principal, sin otros campos.
    - Descripción: breve texto en el que se resaltan detalles únicos del producto o servicio. Es importante incluir en él palabras claves de la búsqueda realizada.

Figura 1: Ejemplo anuncio de texto básico.

    - Campos de ruta (opcional): aparece después de la URL, permite ofrecer una mayor idea del contenido de la página web.

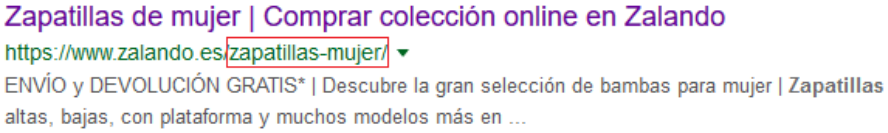

Figura 2: Ejemplo anuncio de texto con campo de ruta.

    - Extensiones: aportan información adicional. Se explicarán con más detalle a continuación. Se pueden considerar como un nuevo formato distinto a los anuncios de textos.

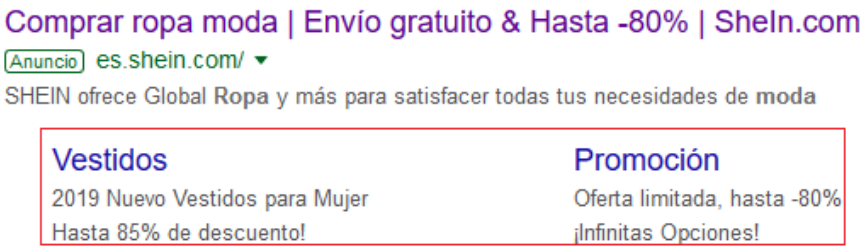

Figura 3: Ejemplo anuncio de texto con extensiones.

- Anuncio sólo de llamada: permite llamar directamente a la empresa con un solo clic. Este tipo de anuncios busca únicamente aumentar las llamadas telefónicas recibidas. Aparecen en la red de búsqueda de Google y asociados.
- Anuncio de imagen: anuncios gráficos, que pueden llamar la atención de forma más sencilla a los usuarios. Aparecen en asociados de la red de búsqueda de Google (pero no en la propia red) y en la red de visualización.
- Anuncios de vídeo: son anuncios más complejos, generalmente usados por grandes empresas. Es lo más parecido a la publicidad convencional que aparece en televisión, pero permite dirigirse a nichos más concretos. Aparecen en

asociados de la red de búsqueda de Google (pero no en la propia red) y en la red de visualización, siendo YouTube su principal medio.

- Anuncio de respuesta: la red de visualización ofrece anuncios que se ajustan a las páginas y aplicaciones que el usuario está visitando. Estos anuncios se mezclan con el contenido propio de la página o aplicación visitada, haciendo que los espectadores los vean con mayor facilidad. Son anuncios empleados para crear conciencia sobre los productos o servicios y aparecen sólo en la red de visualización.
- Anuncios de venta de productos: estos anuncios incluyen una imagen del producto, junto a su precio y otros datos de interés. Son útiles cuando se quiere presentar un extenso catálogo, ya que permite ofrecer mayor información sobre el producto antes de que el usuario haga clic en el anuncio. Aparecen en la red de búsqueda (principalmente en Google Shopping) y asociados.
- Anuncios de promoción de aplicaciones: estos anuncios permiten fomentar la imagen de las aplicaciones y su descarga de forma sencilla (en un solo clic). Aparecen en la red de búsqueda y visualización, pero exclusivamente en dispositivos compatibles con la aplicación promocionada.

Las extensiones amplían la información disponible del anuncio al que acompañan, aportando a dicho anuncio mayor visibilidad, agregando más contenido, mayor rendimiento, aumentando hasta un 15% los clics obtenidos, y mayor valor, puesto que no suponen ningún coste, es decir, tanto si el anuncio tiene extensiones, como si carece de ellas, se pagará lo mismo, dependiendo únicamente de que el usuario realice o no clic sobre él.

Un anuncio puede incluir un número ilimitado de extensiones, pero estas no se mostrarán siempre. Google Ads sólo incluye extensiones cuando considera que, con ello, se mejora el rendimiento del anuncio. De la misma forma que se determina si se añaden o no extensiones, se selecciona cuales mostrar, de forma individual en cada búsqueda realizada en Google, además, hay extensiones que se generan automáticamente, como puede ser la nota que ha recibido un producto por usuarios que ya lo han consumido previamente.

Extensiones disponibles, categorizadas según el objetivo que se busca:

- General, son denominadas extensiones universales. Hay tres:
    - Enlace de sitio web: empleadas para dirigir directamente al usuario, a una página concreta del sitio web del anunciante. Por ejemplo, podemos crear estas extensiones para mostrar en un solo clic el horario, la página donde realizar pedidos, etc.
    - Texto destacado: permiten añadir un texto adicional al anuncio. Sirve para indicar factores positivos del producto. Estos textos suelen ser del tipo “llamada gratuita”, “envío urgente”, etc.
    - Fragmentos estructurados: destacan la información que resulta más útil a los clientes. Por ejemplo, una categoría o listado de productos.
- Obtención de ventas en una tienda física:
    - Ubicación: mostrando la ubicación del negocio, junto a otras extensiones como enlace de sitio web o texto destacado (Ejemplo: “2x1 en nuestra

tienda"), puede hacer que los usuarios se animen a visitar la tienda anunciada.

- Conseguir que los clientes contacten con la empresa:
  - Llamada: añadir un número de teléfono para contactar. Muy útiles en los dispositivos móviles smartphone, en los que, con un solo clic en esta extensión, el cliente puede realizar la llamada.
  - Mensaje: permite mandar un mensaje de texto directamente desde el anuncio.
- Obtención de ventas en el sitio web:
  - Precio: muestran el precio del producto o servicio, pudiendo explorarles y compararles sin la necesidad de hacer clic en el anuncio.
  - Promoción: resalta ofertas específicas. Por ejemplo, 20% de descuento en calzado deportivo.
- Facilitar la descarga de una aplicación:
  - Aplicación: permite descargar la aplicación anunciada de una manera muy sencilla. Disponible a nivel mundial para dispositivos Android e iOS (incluyendo tabletas).

Una vez que hemos visto los diferentes tipos de anuncios disponibles en Google Ads, podemos comenzar a estudiar las campañas de anuncios. Las campañas son grupos de anuncios (cada anuncio se determina por sus palabras claves y sus pujas) que comparten un presupuesto, segmentación geográfica, idioma y distribución de Red de Google, entre otros, permitiendo organizar las categorías de productos que se ofrecen. Toda campaña contiene, al menos, un anuncio y podemos tener varias campañas en cada cuenta de Google Ads.

Las campañas se pueden clasificar en los siguientes tipos:

- Campaña de búsqueda: los anuncios aparecen en los resultados de la red de búsqueda de Google según términos que sean relevantes respecto a las palabras claves del anuncio. Estas campañas son útiles para las empresas que desean mostrar anuncios de texto a clientes de alto potencial cuando están buscando productos relacionados. Por tanto, este tipo de campaña es recomendable cuando se pretende que los anuncios se muestren cerca de los resultados de búsqueda de Google y cuando se quiere llegar a clientes que buscan el producto o servicio anunciado.
- Campaña de visualización: estos anuncios podrán aparecer a lo largo de toda la red de visualización de Google cuando su contenido coincida con el de los sitios web o aplicaciones móviles visitados por el usuario. Estas campañas son útiles para anunciantes que desean generar conciencia de su negocio y audiencias especificas en toda la web.
- Campaña de vídeo: permiten ejecutar anuncios de vídeo, principalmente en YouTube, pero también a través de otros lugares de la red de visualización de Google.
- Campaña de *shopping*: con este tipo de campañas los anuncios pueden aparecer en Google Shopping, junto a los resultados de una búsqueda y en otros lugares web asociados a Google como YouTube. Estas campañas son útiles para minoristas que deseen promocionar su inventario y encontrar mejores clientes.

- Campaña de aplicaciones: promueven la descarga de aplicaciones tanto en la red de búsqueda como en la de visualización y sitios web asociados. Este tipo de anuncios se generan automáticamente tras indicar algunas características básicas de la aplicación como idioma, ubicaciones, presupuesto, entre otros. El sistema probará diferentes combinaciones para encontrar los anuncios con mejor rendimiento sin necesidad de intervenir en ello.

Toda campaña requiere de una configuración, que será aplicada sobre todos los anuncios pertenecientes a la misma. Esta configuración dependerá del tipo de campaña seleccionado, las principales opciones son:

- Nombre de la campaña: Google Ads indica un nombre por defecto, pero se puede modificar. No es visible para los clientes.
- Tipo de campaña: al determinar un tipo de campaña, la configuración se adapta lo mejor posible para alcanzar sus objetivos.
- Redes donde se desea que aparezcan los anuncios.
- Dispositivos en los que pueden aparecer los anuncios. También es posible realizar modificaciones en el anuncio dependiendo del dispositivo (móvil, tableta u ordenador) en el que vaya a aparecer.
- Localización geográfica y lenguaje.
- Pujas y presupuesto: permite establecer las pujas manualmente o dejar que Google Ads lo realice automáticamente, esto indica el máximo que se puede pagar por un clic en el anuncio, pero no tiene por qué ser lo que se va a pagar, es decir, se puede ganar una subasta por una cantidad menor. La estrategia de pujas establece la forma en que los usuarios interactúen con los anuncios. Por otro lado, el presupuesto indica el dinero dispuesto a invertir en la campaña diariamente.
- Extensiones de anuncios.
- Otras opciones como horario de campaña, opciones locales avanzadas, etc.

Otro aspecto fundamental en Google Ads son las palabras claves. Una palabra clave es una palabra o conjunto de pocas palabras que ayudan a determinar dónde y cuándo aparecerán los anuncios. Tener una buena lista de palabras claves permite alcanzar sólo a las personas que están realmente interesadas en los productos o servicios, y por tanto quienes más opciones tienen de convertir.

Podemos usar las palabras claves de diferentes formas, según los objetivos de negocio. Por ejemplo, si se busca llegar a un público muy amplio para crear conciencia de los productos ofrecidos, se usarán palabras claves muy generales (ejemplo: zapatillas); sin embargo, si se quiere encontrar a un sector muy concreto ya que sólo nos interesa convertir, se establecerán palabras claves muy específicas (ejemplo: zapatillas de baloncesto para niños). Por otro lado, también es posible añadir palabras claves negativas, las cuales hacen que el anuncio no aparezca si están incluidas en la búsqueda permitiendo reducir costes (ejemplo: correr).

Para gestionar cuando aparecen los anuncios creados según las búsquedas realizadas se emplean los diferentes tipos de concordancias. Esto hace posible controlar que tipo de variaciones de palabras clave hacen que el anuncio aparezca como resultado de una búsqueda. Tipos de concordancia:

- Concordancia amplia: muestra los anuncios teniendo en cuenta las palabras claves, sus sinónimos y posibles faltas de ortografía. Es la opción más general.
- Concordancia amplia modificada: muestra los anuncios a partir de las palabras claves y sus sinónimos, pero no incluye las palabras similares con faltas de ortografía.
- Concordancia de frase: muestra los anuncios basándose en frases exactas y pequeñas variaciones.
- Concordancia exacta: se muestra los anuncios que contengan las palabras claves exactas a la búsqueda realizada. Es la opción más específica.
- Concordancia negativa: excluye las palabras indicadas de las búsquedas. Evita mostrar anuncios en sitios web no relacionados con los productos ofrecidos.

Una vez se ha realizado el anuncio, este aparecerá como resultado de una búsqueda si gana la subasta. Esta subasta es el proceso que se lleva a cabo con cada búsqueda individualmente para decidir qué anuncios aparecerán con la búsqueda realizada. Las subastas se pueden resumir en tres pasos:

1. En primer lugar, se obtienen todos los anuncios que coinciden con la búsqueda realizada.
2. A continuación, se filtran estos anuncios para desechar aquellos que no sean aptos, como los anuncios orientados a otros países.
3. Finalmente, de los anuncios que pasan el filtro, sólo se mostrarán aquellos que poseen un ranking lo suficientemente alto. Para determinar el ranking de un anuncio se tiene en cuenta su oferta, la calidad del anuncio, los límites del ranking del anuncio (oferta mínima necesaria para que el anuncio se muestre en una posición determinada), el contexto de la búsqueda del usuario, la experiencia de la página web destino, etc.

Es importante tener en cuenta que la oferta económica más elevada no tiene por qué ganar la subasta, ya que se tienen muchos más factores en cuenta. Google trabaja de este modo para evitar que aparezcan malos anuncios en las búsquedas, lo cual provocaría un menor nivel en su buscador. También hay que destacar que el proceso de subasta es individual para cada búsqueda en Google y, por tanto, cada subasta puede tener resultados potencialmente distintos dependiendo de la competencia en cada momento.

Generalmente se habla de coste por clic, ya que es el más empleado, es decir una empresa paga cuando se hace clic en su anuncio, pero Google Ads permite otros:

- CPM (*Cost Per Mille*, coste cada mil impresiones): se paga por cada mil visitas en la red de visualización de Google . Se emplea esta opción cuando queremos crear conciencia de la marca.
- vCMP (*visible CMP*): versión de CPM en la que se asegura que se visualiza al menos el 50% del anuncio durante más de 1 segundo.
- CPC (*Cost Per Click*, coste por clic): se paga por cada clic que se hace en los anuncios. Como hemos mencionado anteriormente, es el formato más empleado por Google Ads. No hay que confundir el CPC con la oferta que se tiene en cada anuncio, ya que esta se puede considerar como el CPC máximo del anuncio, y generalmente el CPC real es menor.
- CPA (*Cost Per Action*, coste por adquisición): se paga cuando el usuario realiza alguna acción deseada por el anunciante (comprar un producto, registrarse, etc.).

Este formato tiene un mayor rendimiento, ya que solo se paga en caso de obtener algún beneficio, pero también es bastante más costoso.

- CPV (*Cost Per View*, coste por visita): se emplea en campañas de vídeo, pagando cuando un espectador mira durante 30 segundos el video o al interactuar con este.

El formato de puja más adecuado para cada campaña dependerá de los objetivos de esta. De forma general, podemos abarcar las campañas en tres grandes grupos de objetivos:

- Crear conciencia, centrándose en las impresiones. Se emplea CPV o vCPM.
- Conseguir visitas en la página web, centrándose en los clics. Se emplea CPC.
- Impulsar las ventas en la web, centrándose en las conversiones. Se emplea CPA.

Tras realizar los anuncios, un aspecto muy importante es revisarlos continuamente a lo largo del tiempo y hacer las variaciones que sean necesarias para obtener mayores beneficios (por ejemplo, en un anuncio con alto porcentaje de conversión, pero pocas impresiones desearemos que estas aumenten. Para ello una idea es aumentar su CPC máximo). Google Ads ofrece numerosas herramientas de análisis para explicar los resultados de las campañas.

Empleando las pujas automáticas, Google Ads puede modificar su importe máximo, teniendo en cuenta el resultado de subastas anteriores, con el fin de alcanzar el objetivo de la empresa. Dependiendo de cuál sea este objetivo, Google Ads seguirá una estrategia diferente:

Objetivo	Estrategia de puja
Aumentar las visitas al sitio web.	Maximizar clics: define las pujas automáticamente para obtener el mayor número de clics posible dentro del presupuesto asignado.
Aumentar la visibilidad.	Cuota de impresiones objetivo: define las pujas automáticamente para mostrar el anuncio en la parte superior absoluta de la página, o en la parte superior de la página, o en cualquier parte de la página de resultados de búsqueda de Google.
Obtener más conversiones con un CPA objetivo.	CPA objetivo: define las pujas automáticamente para generar el mayor número de conversiones al CPA objetivo establecido. Algunas adquisiciones se podrán producir a un mayor o menor coste que el indicado.
Alcanzar el ROAS objetivo cuando se valora cada conversión de forma diferente.	Se define ROAS (*Return Of Ad Spend*, o retorno de inversión publicitaria) como (Ingresos/Inversión)×100. ROAS objetivo: define las pujas automáticas para maximizar el valor de conversión con el ROAS objetivo. Algunas conversiones pueden tener un mayor o menor ROAS que el establecido.
Obtener más conversiones mientras se gasta el presupuesto.	Maximizar conversiones: define pujas automáticas para obtener el mayor número de conversiones mientras se gasta el presupuesto.

Tabla 1: Tabla con las diferentes estrategias de puja automática en Google Ads

A la hora de realizar una puja hay que tener en cuenta muchas consideraciones, que permitirán adaptar dichas pujas dependiendo de las circunstancias, algunas de las más importantes son:

- Dispositivo: móvil, ordenador o tableta. Hay que determinar en qué medio el producto anunciado tiene mayor interés. Por ejemplo, la descarga de una aplicación interesa que aparezca en un dispositivo móvil.
- Localización: permite ajustar las pujas por ciudades, países, distancias, etc.
- Hora del día: habrá anuncios que tengan mayor importancia a cierta hora del día como puede ser un restaurante entre las 13:00 y las 16:00.
- Contenido superior: permite realizar pujas de mayor cantidad en contenidos populares, como puede ser un video de YouTube.
- Lista de *remarketing*: es posible ajustar las pujas para obtener mayor probabilidad de aparecer en búsquedas de usuarios que ya visitó anteriormente la página web del anunciante.
- Idioma: se evita realizar pujas sobre búsquedas realizadas en un idioma diferente al que tiene el anuncio.
- Buscador, sistema operativo, etc.

2.1.3 Dataset de Google Ads

Una vez que hemos entendido el funcionamiento básico de Google Ads, se procede a buscar un dataset con el que trabajar. En este proyecto emplearemos el dataset público que podemos encontrar en *Google Code Archive* con el nombre de *open-advertesing-data* (ubicado en la página web *https://code.google.com/archive/p/open-advertising-dataset/downloads*). Este dataset consta de nueve directorios que podemos descargar:

- *bids-dataset-1*: nos encontramos con dos subdirectorios, uno del mercado de Gran Bretaña (*UK-Market)* y otro de Estados Unidos (*US-Market*), distribución que seguiremos encontrando a lo largo de los directorios de este dataset. Ambos subdirectorios contienen ficheros csv, con información sobre una campaña de Google AdWords entre el 25/12/2011 y el 31/05/2012. Cada fichero csv representa un día, cuya fecha se incluye en el nombre del fichero. Este dataset no se emplea ya que se aprecian gran cantidad de ficheros duplicados.
- *bids-dataset-2-20130214*: al igual que en el directorio anterior, nos encontramos con los mismos subdirectorios para los mercados británico y estadounidense, en los que hay 244 ficheros csv en cada uno, correspondientes al tráfico generado por ciertos anuncios de una campaña de AdWords entre las fechas 24/05/2012 y 14/02/2013 (no incluye la totalidad de los días). Haciendo un análisis un poco más detallado observamos que hay 25 días entre las fechas mencionadas de los que no tenemos información, y hay 3 días con información duplicada. En total nos encontramos con 547 registros o filas con información sobre las distintas palabras

clave del mercado. Este será uno de los dataset empleados en el análisis de Google Ads

- *bids-dataset-2:* este directorio es una versión reducida del anterior, con la misma estructura, pero con 95 ficheros csv comprendidos entre el 24/05/2012 y el 08/09/2012. Al ser una versión resumen del anterior, no se empleará.
- *bids-dataset-3-20130214:* directorio análogo a *bids-dataset-2-20130214* con 192 ficheros csv con fechas comprendidas entre el 25/07/2012 y el 14/02/2013. En este caso, hay 14 días sin información y 2 duplicados. En total nos encontramos con 751 registros o filas con información sobre las distintas palabras clave del mercado. Este dataset se empleará en el estudio realizado posteriormente.
- *bid-dataset-3*: directorio análogo a *bids-dataset-3* con 43 ficheros entre las fechas 25/07/2012 y 08/09/2012.
- *webpages-dataset-1* y *webpages-dataset-2:* contienen información acerca de la página web destino de cada campaña para los dataset *bids-dataset-1, bids-dataset-2-20130214* y *bids-dataset-2.*
- *keywords-dataset-1* y *keywords-dataset-2*: lista de las palabras claves en las campañas empleadas en los dataset *bids-dataset-1, bids-dataset-2-20130214* y *bids-dataset-2.*

Los ficheros empleados en este proyecto siguen la misma estructura: en la primera fila aparecen los nombres de las columnas del dataset, a continuación, las 4 siguientes líneas incluyen tanto información general del mercado; indicando la geolocalización (Estados Unidos o Gran Bretaña, dependiendo del subdirectorio donde nos encontremos), un ID (fijo para cada mercado a lo largo de todos los días), etc. como información del día en concreto, haciendo un resumen con el número de clics de todos los anuncios, el CPC medio, etc. Las columnas relevantes, que emplearemos en nuestro estudio son:

- *Average CPC* (CPC medio): indica el coste por clic medio en cada anuncio. A su vez, en la parte general del fichero, se calcula el CPC medio de todo el mercado en ese día en concreto.
- *Clicks:* número de clics que obtiene cada anuncio, en la parte general aparece la suma de todos los clics de cada campaña individual. Podemos observar que no son números enteros, por tanto, podemos suponer (ya que no contamos con información para concretar que indica esta columna exactamente) que no indica el número de clics de cada anuncio sino, por ejemplo, el número de clics × 1000.
- *CTR* (ratio de cliqueo): el porcentaje de clics se puede calcular como el cociente entre el número de clics que se realizan en un anuncio, y las impresiones que este genera. En este dataset, obtenemos el CTR de cada anuncio, así como el general de cada día concreto.
- *Cost* (coste): coste invertido en los anuncios. Al igual que en anteriores columnas, tendremos el coste para anuncio y el coste general del día, calculado como la suma de todos los costes de anuncios.
- *Impressions* (impresiones): número de impresiones que genera cada anuncio, es decir, las veces que gana la subasta y aparece tras la realización de una búsqueda. Como en las columnas anteriores, este dato se muestra para cada anuncio y para el día en general.

- *Keyword* (palabra clave): palabra clave de cada anuncio dentro de las campañas disponibles.

Por otro lado, la fecha del día de cada fichero viene en su propio nombre, por tanto, se ha leído dicho nombre, y se ha manipulado la cadena obtenida para añadir una nueva columna *Date* que nos permitirá el estudio del dataset a lo largo del tiempo.

Finalmente, se ha decidido trabajar con la información general del mercado ya que las campañas individuales no se llevan a cabo todos los días y, por tanto, centrándonos en el mercado general obtenemos más muestras con las que podemos trabajar.

2.2 Criteo

Criteo es una empresa de marketing online a nivel global centrada en el mercado *cross-device,* es decir, en los usuarios que se conectan a una página web desde distintos dispositivos, con el fin de sincronizar el contenido que se ofrece en todos estos dispositivos. Criteo únicamente emplea publicidad de visualización [5].

Criteo fue fundada por un pequeño grupo de ingenieros en París en 2005, desde entonces ha ido creciendo hasta el día de hoy que cuenta con más de 2.700 empleados a nivel mundial, más de 600 mil millones de dólares en transacciones analizadas (fuentes del año 2017) y más de 1.200 mil millones de anuncios publicados (fuentes del año 2017).

2.2.1 Soluciones Criteo

Criteo ofrece diferentes soluciones dependiendo de las empresas o personas que se contraten sus servicios, dividiéndoles en tres grandes grupos:

- Pequeñas, medianas empresas y *marketers* (persona encargada en ayudar a los equipos de ventas a la hora de comercializar un servicio o producto):
    - Generar tráfico e instalaciones: con la información recabada de sus compradores mensuales activos, Criteo pretende crear anuncios a nuevos clientes basándose en sus hábitos de navegación y compra.
    - Conseguir ventas: emplea anuncios de *retargeting*, técnica empleada para impactar a los usuarios que previamente han interactuado con la empresa, para mostrar ofertas muy especializadas y relevantes en cualquier lugar de internet que puedan visitar o bien dirigir a estos usuarios a la aplicación de dicha empresa, o bien a la propia tienda física. También permite alcanzar usuarios con comportamientos afines a clientes que se han estudiado.
    - Aumentar el tráfico del sitio web: presenta la marca de la empresa con ofertas relevantes y flexibles gracias a las recomendaciones inteligentes de productos,

mostrando automáticamente los productos con más probabilidad de generar visitas e interacción. Gracias al *targeting* flexible permite crear una audiencia empleando el activo de datos de Criteo o las propias listas de clientes de la empresa.

  - Aumentar las instalaciones y el tráfico en la aplicación: proporciona ofertas especiales que enlazan directamente con la aplicación para aumentar su tráfico, fomentar su uso y garantizar una relación duradera entre empresa y cliente. Con este tipo de ofertas, también se busca encontrar nuevos usuarios que se animen a instalar la aplicación.

- Marcas, aumentar la visibilidad de los productos con anuncios nativos: permite a las marcas mostrar anuncios nativos a compradores con alta intención de compra en sitios web y aplicaciones de los principales minoristas del mundo. De este modo, se busca aumentar la presencia e impulsar las ventas con campañas que se pueden gestionar en tiempo real y permiten realizar un seguimiento de resultados en todos estos minoristas.
- Minoristas, potenciar las relaciones con las marcas: permite dar a las marcas asociadas una oferta de autoservicio que incluya formatos de anuncios nativos y de visualización y segmentación de los compradores a tiempo real según su comportamiento. Permite a las marcas aprovechar el análisis a nivel SKU (*Stock-keeping unit*, traducido como número de referencia de un artículo).

2.2.2 Productos Criteo

Criteo clasifica sus productos en seis secciones, según los objetivos que podemos ver a continuación:

- *Criteo Dynamic Retargeting*: pretende atraer a nuevos compradores en su proceso de compra mediante anuncios personalizados a tiempo real gracias a la aplicación de aprendizaje automático, maximizando las ventas con una medida de hasta ×13 ROAS. Una de las principales ventajas que se observan empleando este producto es el aumento de ventas de manera rentable ya que Criteo introduce un bloque de código personalizado en las páginas web de sus clientes para ver la actividad de los consumidores y mostrar el anuncio adecuado en el momento oportuno.
- *Criteo Audience Match*: diseñado para completar a *Criteo Dynamic Retargeting*. Este producto emplea aprendizaje automático para obtener un *targeting* con altas tasas de coincidencia para dirigirse de forma muy precisa a clientes a través de anuncios de display de pago dinámicos en la web, navegadores móviles y aplicaciones. Una de las principales ventajas de este producto son aumentar las ventas entre clientes que ya han sido compradores en la página web o que han mostrado interés en la misma.
- *Criteo Customer Acquisition*: impulsa las conversiones de nuevos clientes empleando aprendizaje automático para identificar nuevos clientes a través de los ya existentes, haciendo uso de sus intereses individuales, sus patrones de compra y procesos de navegación.

- *Criteo Sponsored Products para marcas*: ofrece anuncios de productos nativos mostrados en las aplicaciones y sitios web de los principales minoristas a lo largo de todo el mundo. Este producto permite una gestión en tiempo real de campañas y consumidores objetivo, lo cual permite modificar pujas y medir y optimizar resultados.
- *Criteo Commerce Display para marcas*: permite llegar e influir en los consumidores con el fin de facilitar nuevas compras de usuarios que han comprado o mostrado interés en los productos anteriormente. Gracias a este producto se saca partido de datos relevantes como pueden ser el historial de compra, el contenido del carrito en tiempo real, el tiempo local, etc. para llegar a los consumidores con mensajes personalizados y contextualizados.
- *Criteo Direct Bidder*: maximiza los ingresos conectando directamente el inventario de los clientes de Criteo con su posible demanda.

2.2.4 Tecnología Criteo

Criteo aprovecha el *big data* obtenido de la red global para generar soluciones escalares aplicando aprendizaje automático. Los principales elementos tecnológicos que emplea Criteo para implementar sus productos son su *Motor Criteo* y el Criteo *Shopper Graph*.

El *Motor Criteo* potencia sus productos organizando y analizando continuamente la información relevante que obtiene de más de 1.400 millones de compradores activos cada mes y de transacciones producidas en el comercio electrónico por un valor superior a 600.000 millones de dólares anuales. Gracias a este motor, los clientes de Criteo pueden exprimir al máximo su presupuesto al identificar hábitos de los consumidores, entre los que destacan productos ya comprados, sitios visitados, etc.

Podemos distinguir tres componentes principales en el *Motor Criteo*:

- *Dynamic Creative Optimization+* (*DCO+*): tecnología de optimización dinámica que permite diseñar los anuncios de manera personalizada a partir tanto de la información relevante de los compradores, como de los propios estándares de la marca. De este modo, consigue generar el contenido más atractivo para generar el mayor número de ventas en todo momento. Es importante entender que, si en general el tiempo de atención al consumidor es breve, en particular, en el mundo online este tiempo es mucho menor, por ello es muy importante mostrar los anuncios de forma instantánea, gracias a *DCO+* podemos generar estos anuncios en una fracción de segundo incluyendo color, distribución y llamada a la acción según las directrices de cada marca para poder potenciar las conversiones de cada anuncio. Posteriormente veremos cómo se generan estos anuncios.
- Pujas predictivas: emplean la inmensa cantidad de datos recopilados sobre consumidores, productos y sus interacciones, así como de las principales páginas

donde aparecen los anuncios de Criteo para producir la puja adecuada, al precio idóneo y en el momento preciso para cada comprador.

- Recomendaciones de productos: la gran cantidad de información relevante recopilada sobre el comportamiento de los consumidores permite alcanzar compradores con recomendaciones personalizadas creadas para generar conversiones.

A lo largo de la sección de Criteo se ha hecho hincapié en la idea de que cada anuncio se genera y dirige específicamente a un comprador individual. Pues bien, esta idea se puede realizar gracias al *DCO+*. Pasos que sigue Criteo para la creación de anuncios personalizados:

1. Composición: desarrollo del marco de diseño exclusivo para cada marca, con todos los elementos de diseño necesarios para producir anuncios atractivos.
2. Optimización creativa en tiempo real: *DCO+* determina el diseño óptimo de la composición, identificando en tiempo real los elementos de marca, colores y diseño más atractivo para cada comprador y contexto basándose en la información relevante sobre compradores del motor Criteo.
3. Renderizado: a partir de la información relevante de la optimización creativa en tiempo real, el renderizado crea un anuncio óptimo en el tamaño necesario para cada impresión individual teniendo en cuenta recomendaciones de productos pensadas en función de la retroalimentación de productos del cliente. De este modo se crea un anuncio totalmente personalizado que atraiga realmente a los compradores, empleando sus propias recomendaciones.
4. Elementos activos: esta función permite a *DCO+* optimizar la publicidad en redes sociales como Instagram y Facebook incluyendo elementos activos como llamadas a la acción, logos de marcas, descripciones de productos o calificaciones de los compradores.

En lo referente a *Criteo Shopper Graph*, cabe destacar que se centra en emplear la información para generar resultados. Los datos detallados de los compradores son la base del ecosistema Criteo. Al reunir y agrupar tres tipos de datos clave sobre el consumidor como su identidad, sus intereses y sus mediciones, los productos Criteo son capaces de saber continuamente lo que quieren los compradores.

Esta tecnología unifica los datos de los diferentes dispositivos que emplea cada usuario tanto online como offline para lograr el mayor impacto posible. *Criteo Shopper Graph* garantiza la identificación de los más de 700 millones de compradores online activos que emplean varios dispositivos para comprar, y los más de 10.000 sitios web a lo largo de todo el mundo que comparten sus datos con Criteo.

2.2.5 Dataset de Criteo

En el proceso de búsqueda de un dataset interesante con el que poder trabajar, encontramos una competición de *Kaggle* que resultó ser muy práctica para la realización de los objetivos marcados en este proyecto.

Esta competición se titula *Dysplay Advertising Challenge* y se puede encontrar en la página web *https://www.kaggle.com/c/criteo-display-ad-challenge/overview*. Por otra parte, los datasets que emplea se pueden descargar desde Criteo, en la web *https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset.*

Descargando este archivo nos encontramos principalmente con dos dataset, uno de entrenamiento y otro de test, que emplearemos para generar un modelo predictivo y para comprobar la validez de dicho modelo respectivamente. Este archivo incluye un tercer fichero con una breve descripción de los dos dataset.

El dataset de entrenamiento, *train.txt*, contiene información sobre el tráfico generado por Criteo en un período de 7 días. Cada fila corresponde a un anuncio gráfico generado por Criteo, en total hay 45.840.617 filas ordenadas temporalmente. En lo referente a las columnas, hay 40 columnas que podemos agruparlas en tres grupos:

- La primera columna es binaria y será la salida del modelo, la pulsación. Es decir, si un anuncio ha sido clicado o no. En caso positivo, esta columna tendrá un valor 1, mientras que si el anuncio no ha resultado con un clic tendrá 0. A esta columna la llamaremos Label.
- A continuación, disponemos de 13 columnas numéricas, que corresponden a características de cada anuncio. Nombraremos estas columnas como I1-I13.
- Finalmente, hay 26 columnas con valores categóricos, correspondientes también a distintas características de los anuncios de Criteo. Nombraremos estas columnas como C1-C24.

Este dataset está totalmente anonimizado, ya que no conocemos el nombre de las columnas, ni su contenido, en el caso de las columnas categóricas ha sido anonimizado empleando un hash de 32 bits.

El dataset de test, *test.txt*, es similar en su estructura salvo por no disponer de la primera columna con el valor del clic, ya que ésta será la columna que el modelo debe predecir. Por tanto, este dataset está compuesto por 39 columnas de características tanto numéricas como categóricas de los anuncios publicados por Criteo.

Estos datos corresponden al tráfico generado por Criteo el día siguiente al tráfico recopilado en el dataset data, dando lugar a un total de 6.042.135 filas. Es decir, esta competición trabaja con los datos de 8 días de tráfico Criteo, de los cuales, los 7 primeros días corresponden al dataset de entrenamiento, y el octavo día al dataset de test.

2.3 Otras plataformas de publicidad en Internet.

Un aspecto fundamental a la hora de realizar campañas en internet es el canal empleado. A día de hoy hay numerosas posibilidades, cada una con sus ventajas e inconvenientes. No hay una plataforma perfecta, ya que cada una de ellas tiene su propio alcance y la posibilidad de alcanzar de forma más sencilla a un tipo público objetivo u otro.

Por tanto, cada posible usuario de estas plataformas debe tener en cuenta sus objetivos y las características de sus productos a la hora de seleccionar el medio que emplea para crear y mostrar sus anuncios.

Hasta ahora hemos estudiado dos plataformas, Google Ads y Criteo, ya que son en las que nos hemos centrado en este proyecto, pero como ya hemos mencionado hay muchas más. A continuación, vamos a hablar brevemente de alguna de estas otras opciones disponibles en el mercado.

En este punto, vamos a dividir el conjunto de las plataformas de publicidad online disponibles en dos grandes grupos, aquellas que ofrecen publicidad a lo largo de diferentes webs en internet (como Criteo o Google Ads) y las redes sociales ya que por lo general estas disponen de su propia plataforma con la que ofrecer publicidad en la propia red social.

En el primer grupo podemos encontrarnos junto a las ya explicadas Google Ads y Criteo, otras plataformas como AdForm, AppNexus, Sizmek, Taboola, Outbrain, etc. Estas plataformas tienen un comportamiento y una oferta similar a Criteo, cada una con diversas ventajas e inconvenientes. También cabe destacar Baidu Advertising, plataforma del buscador principal en China y por tanto con varias similitudes respecto a Google Ads.

Por otro lado, tenemos las plataformas de publicidad en redes sociales como Twitter Ads, Facebook Business, Linkedin Marketing Solutions entre otras muchas. En este caso se pueden observar más variaciones entre las plataformas, ya que cada una de ellas adecua sus anuncios a sus características. Como es comprensible, hay mucha diferencia entre, por ejemplo, una red social seria de trabajo como Linkedin o en una red social de ocio como puede ser Twitter.

Con el fin de entender el funcionamiento de estas plataformas centradas en una red social en concreto, haremos una breve introducción a Twitter Ads.

Twitter Ads ofrece la posibilidad de crear anuncios enfocados en los objetivos más importantes para cada empresa. Esta plataforma permite segmentar los clientes potenciales a través de intereses, datos demográficos o actividad en la propia red social. Twitter también facilita la obtención de nuevos clientes a través del boca a boca de seguidores que recomienden o simplemente mencionen los productos de la empresa anunciante. Otro aspecto fundamental de las campañas de Twitter Ads es que permiten cobrar únicamente en caso de que uno de los usuarios complete una acción que figure en los objetivos de la campaña como visitar el sitio web del anunciante o descargar su aplicación [13].

Existen tres opciones principales para anunciarse en Twitter Ads. La primera de ellas son los tweets promocionales, este tipo de publicidad permite aumentar el alcance de la empresa y ganar notoriedad. Estos tweets promocionales se integran totalmente en el tablón de los usuarios, lugar donde aparecen todos los tweets de las personas a las que siguen, ya que tienen la misma apariencia que cualquier otro tweet salvo porque en la parte inferior del mismo se especifica que es promocionado.

Al igual que en Google Ads, estas campañas sólo aparecen si cumplen un cierto nivel de requisitos en función de calidad, contenido similar al usuario, etc. Esto se debe a que no se desea que la publicidad en esta red social sea intrusiva y molesta.

El segundo tipo de campañas emplea cuentas promocionadas, cuyo objetivo es aumentar los seguidores formando una comunidad lo más grande posible. Gracias a esta promoción, se consigue aparecer en primer lugar en la lista de sugerencias que los usuarios visualizan.

Finalmente, podemos emplear campañas de tendencias promocionales, las cuales permiten alcanzar a una gran cantidad de personas ya que nos permite crear un *hashtag* y hacer que aparezca en la primera posición de las tendencias de un país o región.

En conclusión, hemos observado alguna de las numerosas plataformas de publicidad en internet que existen. Estas plataformas se pueden dividir en dos grandes bloques, las plataformas que muestran sus campañas en diferentes sitios webs asociados, cuyo funcionamiento es similar, pero con sus particularidades como los sitios webs asociados; y las plataformas de las distintas redes sociales, estas plataformas son muy diferentes unas a otras, ya que cada una de ellas ajusta su formato a las características de la propia red social.

3. Aprendizaje automático.

El aprendizaje automático pretende crear sistemas que aprendan de forma automatizada, es decir, por ellos mismos. Estos sistemas deberán ser capaces de resolver problemas y tomar decisiones basándose en la experiencia acumulada, o lo que es lo mismo en casos resueltos anteriormente.

Podemos considerar el aprendizaje automático como un proceso que tiene lugar en dos fases. En la primera de ellas el sistema selecciona las características más relevantes de un objeto o evento a través de un conjunto de datos. Tras ello, en la segunda fase, se procede a generar un modelo de dicho objeto o evento [14].

Podemos distinguir varios tipos o ramas dentro del aprendizaje automático:

- Aprendizaje supervisado: los datos están etiquetados o tienen un valor asociados a ellos junto a todas las características que definen al objeto o evento. Dependiendo de esta etiqueta podemos distinguir entre:
    - Clasificación: predicción de una etiqueta o categoría. Por ejemplo, determinar si en una imagen aparece un perro o un gato. (Predicción discreta).
    - Regresión: predicción de un valor numérico. Por ejemplo, predecir el dinero que puede gastar un cliente en una página web. (Predicción continua).
- Aprendizaje no supervisado: los datos no están etiquetados, sólo disponemos de las características de los objetos, a partir de los cuales pretendemos organizarlos. Podemos ver el aprendizaje no supervisado como la tarea de encontrar patrones y estructuras en los datos de entrada. También podemos distinguir dos tipos:
    - Clustering (agrupamiento): creación de grupos formados por objetos con características similares entre sí, y distintas de otros.
    - Detección de anomalías: identificación de objetos que no se ajustan a un patrón esperado.
- Otros tipos de aprendizaje:
    - Aprendizaje por refuerzo: los datos de entrenamiento no contienen el resultado objetivo sino un posible resultado junto con una medida de la calidad (cómo de bueno es) de dicho resultado. Ahora en lugar de tener características de entrada y su salida como en el aprendizaje supervisado, tendremos las características de entrada, unas posibles salidas y el grado de calidad de estas.
    - Sistemas recomendadores: sistema de filtrado de información que presenta al usuario elementos que sean de su interés.

En este Trabajo Fin de Grado nos hemos centrado en el aprendizaje supervisado. Esta rama de aprendizaje automático será empleada junto con el dataset Criteo para crear un modelo de aprendizaje capaz de predecir la probabilidad de éxito de una campaña campañas a través de sus características.

3.1 Aprendizaje automático supervisado.

En el aprendizaje automático supervisado los datos están etiquetados. El principal objetivo es predecir la etiqueta de los nuevos datos, ya sea una etiqueta (clasificación) o un valor real (regresión). En el caso estudiado en este proyecto, empleando los datos de Criteo estamos ante una clasificación ya que intentamos determinar si un anuncio va a ser clicado (etiquetado como 1) o no (etiquetado como 0). Por tanto, tenemos ante nosotros un problema de clasificación con dos etiquetas, es decir un problema de clasificación binario.

La siguiente imagen procede de un ejemplo del libro *Learning from data [15]* empleado para mostrar los diferentes elementos del aprendizaje supervisado. Este ejemplo trata de determinar el crédito máximo que se debe proporcionar en la tarjeta de crédito de un cliente. Componentes del aprendizaje supervisado:

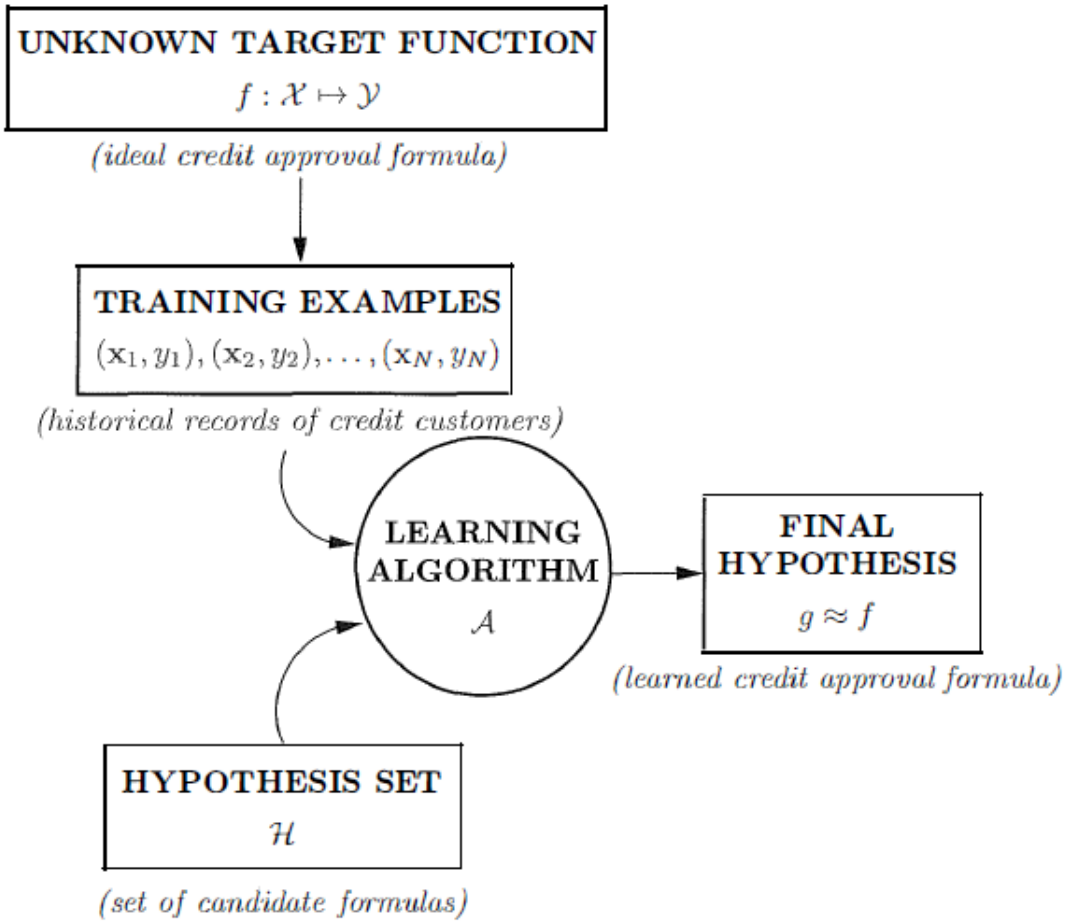

Figura 4: Componentes del aprendizaje supervisado [15].

- Función objetivo desconocida (*unknown target function*), *f(X→Y)*: fórmula ideal para obtener la salida a través de los datos de entrada. No debemos disponer de esta función ya que de lo contrario no tendría sentido aplicar aprendizaje supervisado. En el ejemplo indicaría la fórmula ideal de asignación de crédito.

- Datos de entrenamiento (*training examples*), *D*: conjunto de datos formados por las características del objeto o evento estudiado ($\mathbf{x_1, x_2, ..., x_n}$) junto con sus respectivas salidas ($y_1, y_2, \ldots, y_n$). En este ejemplo estarían formados por las características de los clientes históricos y la cantidad de crédito concedida.
- Conjunto de hipótesis (*hypothesis set*), *H*: conjunto de fórmulas a considerar para la obtención de una función *g* lo más parecida a la función objetivo *f*. Cada una de estas hipótesis *h* dentro del conjunto *H*,
- Algoritmo de aprendizaje (*learning algorithm*), *A:* mecanismo con el que determinar una función *g* a partir del conjunto de hipótesis *H*, con el fin de conseguir una *g* lo más parecida a la función objetivo *f*.
- Hipótesis final (*final hypothesis*), *g*: fórmula aprendida a partir de los datos de entrenamiento.

En el caso de Criteo, la función objetivo sería la fórmula capaz de indicar si un anuncio va a tener éxito o no empleando sus características. Por otro lado, los datos de entrenamiento están almacenados en el fichero llamado *train.txt*, como ya vimos en la explicación del dataset, la primera columna sería la salida y las demás características de los anuncios.

En este punto hay dos cuestiones claves a tener en cuenta. La primera de ellas sería como determinar el parecido entre la función objetivo *f* y la función obtenida a través de los datos de entrenamiento *g*, para ello se emplea la función de error. Funciones de error hay muchas, por ejemplo, si hablamos de clasificación podemos determinarlo a partir de los casos mal clasificados, o en regresión empleando el error cuadrático medio. En la competición de *kaggle* empleada con Criteo se calcula el error a través de "la pérdida logarítmica" (*logarithmic loss*). Esta función, también conocida como *Log Loss*, se emplea en modelos de clasificación donde la salida se determina como un valor de probabilidad entre 0 y 1.

La pérdida logarítmica penaliza la diferencia entre la probabilidad calculada por el modelo y el valor real de la etiqueta. En una clasificación binaria, este error se calcula como:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot log_e(\hat{y}_i) + (1 - y_i) \cdot log_e(1 - \hat{y}_i)]$$

Figura 5: Ecuación del error Log Loss.

La fórmula calcula la media de todos los errores, predicción por predicción, teniendo en cuenta los casos en los que la etiqueta es 0 (parte derecha de la fórmula) y los casos en los que la etiqueta es 1 (parte izquierda de la fórmula). Este error crece exponencialmente cuando la probabilidad que se ha precedido está alejada de la etiqueta real. Podemos visualizar este error en la siguiente imagen:

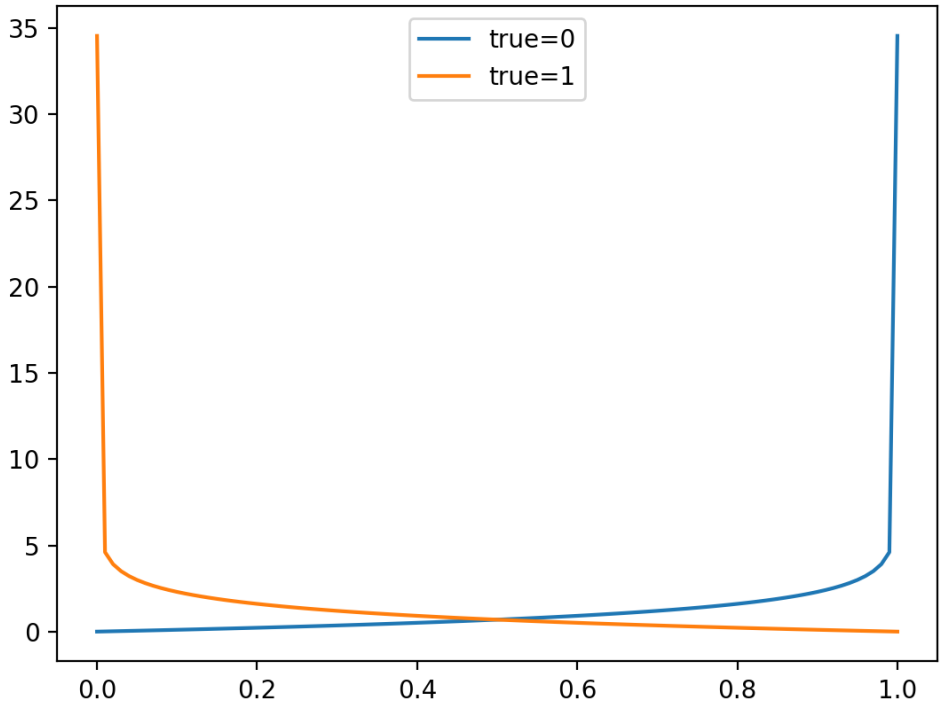

Figura 6: Gráfica del valor del error empleando Log Loss.

En el eje de abscisas se indica la predicción realizada, entre 0 y 1, mientras que en el eje de ordenadas se muestra el valor del error. La línea roja muestra el error cuando la etiqueta vale 1 y la línea azul refleja el error para etiquetas con valor 0. Con el fin de aclarar completamente esta fórmula vemos algunos ejemplos (de forma análoga se podría hacer para el valor de etiqueta 0):

- Valor real 1 y predicción de 0.9 genera un error de 0.105.
- Valor real 1 y predicción de 0.5 genera un error de 0.693.
- Valor real 1 y predicción de 0.1 genera un error de 2.303.

La segunda cuestión sería cómo aprender la función *g*. Esto se realiza a través del modelo de aprendizaje, que consta del conjunto de hipótesis *H*, junto con el algoritmo de aprendizaje *A*, que nos permite seleccionar la función *g* del conjunto de hipótesis *H* con el mínimo error posible.

Hay muchos modelos de aprendizajes, algunos muy simples, como el modelo lineal y otros más complejos como los árboles de decisión o los SVM (*Support Vector Machines*). Los modelos de aprendizaje empleados en este proyecto serán estudiados posteriormente.

Modelos de aprendizaje muy simples o con pocas muestras harán que no lleguemos a ajustar lo suficiente, por tanto, aunque el modelo será capaz de clasificar los datos de entrenamiento, tendrá problemas a la hora de hacerlo con datos nuevos, por eso se dice que el modelo no es capaz de generalizar, produciéndose un subajuste o *underfitting*.

En caso de emplear modelos de aprendizajes muy complejos, podemos ajustar demasiado los datos de entrenamiento, minimizando por completo el error producido en estos, pero al emplear nuevos datos también habrá problemas para clasificarlos ya que nuestro modelo se ha ajustado demasiado a las características concretas de los datos de entrenamiento. A esto se le denomina sobreajuste u *overfitting*.

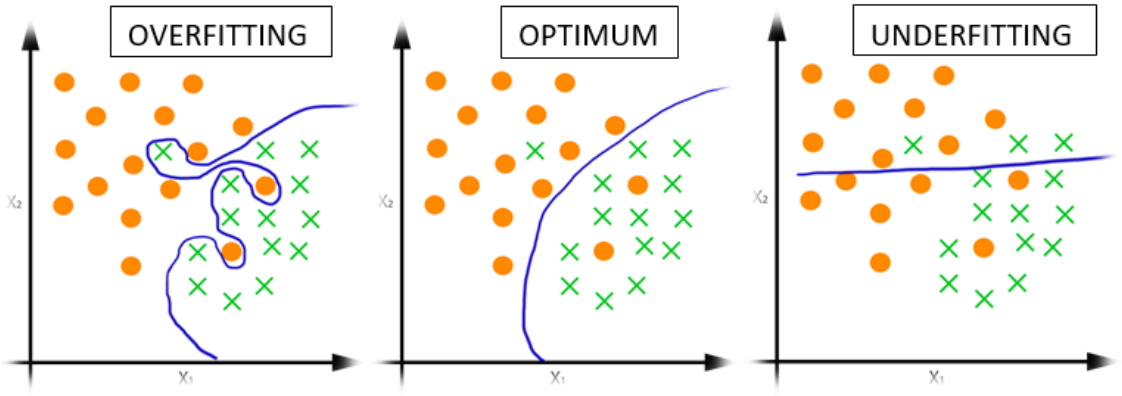

Figura 7: Overfitting y underfitting.

Podemos apreciar en la *Figura 7: Overfitting y underfitting.* cómo se generaliza al producirse *underfitting* y como se ajusta en exceso a los datos de entrenamiento al producirse *overfitting*. Por tanto, el objetivo será encontrar un modelo de aprendizaje óptimo, lo suficiente complejo para no sobregeneralizar pero sin llegar a ajustar en exceso produciéndose *overfitting*.

Como vemos podemos distinguir entre dos errores a la hora de ver cómo se comporta nuestro modelo. El primero de ellos es el error que se comete en la muestra de datos disponibles, con los que generamos el modelo de aprendizaje, denominado E_{in} (*in-sample error*) o error de entrenamiento. Por otro lado, tenemos el E_{out} (*out-of-sample error*), es decir, el error producido en nuevos datos, datos que no se han empleado para generar el modelo. El objetivo principal de todo modelo de aprendizaje es minimizar al máximo posible el E_{out}. Normalmente, se estima este error analizando el comportamiento del modelo en unos datos de test, así que hablaremos de error de test.

Modelos de aprendizaje con *underfitting* tendrán ambos errores muy elevados, es decir, ni se ajustan a los datos de entrenamiento ni a los nuevos datos. Por otro lado, modelos con *overfitting* generarán un E_{in} muy pequeño (a mayor sobreajuste, menor será este error) pero un E_{out}, es decir, el error a minimizar, mayor (de forma análoga, a mayor sobreajuste, mayor será este error). Como podemos ver en la Figura 8, el objetivo debe ser encontrar el punto óptimo de complejidad en el modelo de aprendizaje que minimice el E_{out}. Antes de este punto óptimo se produce *underfitting*, y después tendremos *overfitting*.

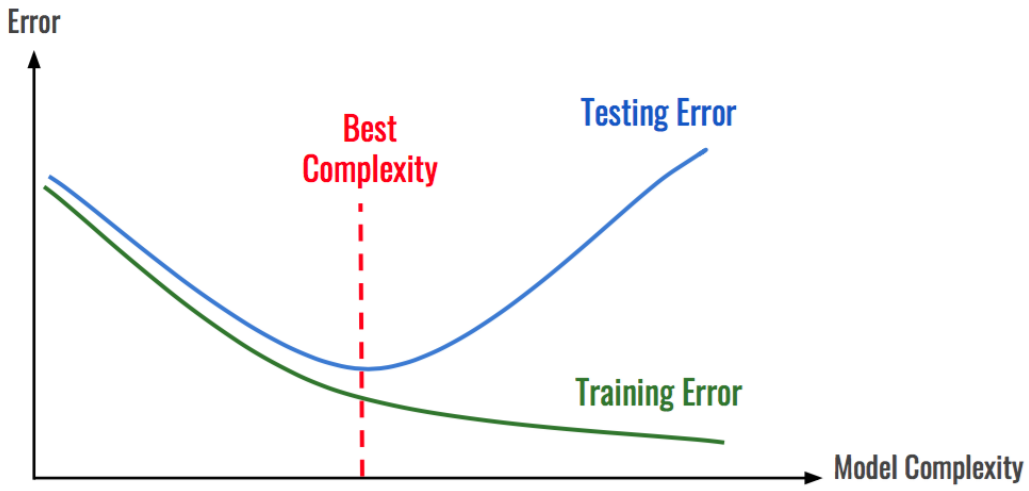

Figura 8: Ein y Eout según la complejidad del modelo [18].

Una de las principales bazas que tenemos para evitar el sobreajuste es la regularización. Para poder entender los efectos de la regularización en primer lugar vamos a recurrir al análisis sesgo-varianza, que descompone el error E_{out} en:

- Sesgo (*bias*): cómo de bien el conjunto de hipótesis *H* aproxima a la función objetivo *f*.
- Varianza (var): cómo de bien podemos seleccionar una buena hipótesis *h* dentro del conjunto de hipótesis *H*.

En la sección 2.3.1 del libro *Learning from data* podemos ver la demostración con la que se obtienen las fórmulas del sesgo y la varianza, cuya suma da lugar al E_{out}:

$$\text{bias}(\mathbf{x}) = (\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2$$
$$\text{var}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]$$

Figura 9: Ecuaciones del sesgo y la varianza.

Para comprender la diferencia entre sesgo y varianza podemos ver dos casos muy extremos. El primero de ellos sería un modelo muy pequeño, con una única hipótesis, por tanto, independientemente de los datos de entrenamiento la hipótesis *g* siempre será la misma lo cual implica que la varianza será nula. Sin embargo, el sesgo puede ser muy elevado ya que con una única hipótesis es muy difícil que el modelo se ajuste bien a la función objetivo.

En el segundo caso, tendríamos un modelo muy grande, con múltiples hipótesis, por tanto, tendremos alguna hipótesis *g* cercana a la función objetivo *f* lo cual produce un sesgo muy pequeño, cercano a nulo. Sin embargo, debido a la variedad de hipótesis generará una alta varianza.

Otro aspecto a tener en cuenta en este punto es el número de datos de entrenamiento a los que tenemos acceso. A mayor número de datos, podremos emplear modelos más complejos sin que la varianza tengo un valor demasiado elevado. Es decir, cuando tenemos pocos datos para entrenar nuestro modelo de aprendizaje debe ser simple,

mientras que al aumentar el número de datos de entrenamiento podremos ir empleando modelos más complejos. Podemos ver el error generado dependiendo del número de muestras de entrenamiento en la siguiente imagen:

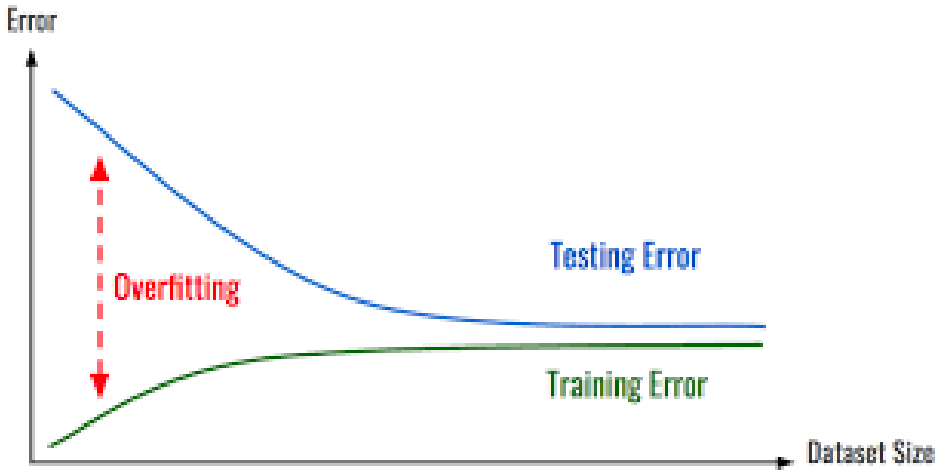

Figura 10: Eout y Ein dependiendo del número de datos de entrenamiento [18].

La regularización es un conjunto de técnicas que permiten reducir el *overfitting* en los modelos de aprendizaje. Una de las principales técnicas de regularización es la modificación de la función de coste añadiendo una penalización con el fin de favorecer los pesos pequeños. De forma general la regularización aumenta ligeramente el sesgo ya que es posible que la mejor hipótesis se ajuste menos a la función objetivo *f*. Por otro lado, la regularización reduce considerablemente la varianza en modelos complejos ya que restringir los valores de los pesos implica la reducción del número efectivo de parámetros y por tanto la reducción de la complejidad del modelo.

Para estudiar cómo afecta la regularización veremos un ejemplo con la regresión lineal y dos tipos de regularización que añaden una penalización a los pesos elevados, la penalización Lasso o L1 y Ridge o L2. Siendo *w* los coeficientes estimados y *x* los predictores del modelo, el valor que se predice en la regresión lineal $\hat{y}$ se obtendrá como:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

Figura 11: Ecuación regresión lineal.

Estableciendo como medida de error el error cuadrático medio, es decir el cuadrado de la diferencia entre el valor predicho *Xw* y el valor original *y*, cada uno de los casos tendrá como función de error final las siguientes:

Regresión lineal sin regularización	$\min_w \|\|Xw - y\|\|_2^2$
Regresión lineal con regularización Lasso	$\min_w \frac{1}{2n_{\text{samples}}} \|\|Xw - y\|\|_2^2 + \alpha \|\|w\|\|_1$
Regresión lineal con regularización Ridge	$\min_w \|\|Xw - y\|\|_2^2 + \alpha \|\|w\|\|_2^2$

Tabla 2: Función de error regresión lineal con y sin regularización.

Como vemos, en ambos casos la penalización aumenta el error multiplicando un parámetro λ por los coeficientes estimados *w*, por tanto, las hipótesis con un valor de *w* más reducido tendrán un error menor. Dependiendo del valor del parámetro λ podremos ocasionar:

- λ = 0. No hay regularización. Posible *overfitting*.
- λ muy pequeño puede no arreglar por completo el problema de *overfitting*.
- λ elevada da lugar a *underfitting* ya que tendríamos un sesgo demasiado elevado.
- Un valor λ adecuado combate el problema del *overfitting* y no genera *underfitting* ya que reduce la varianza significativa sin aumentar el sesgo.

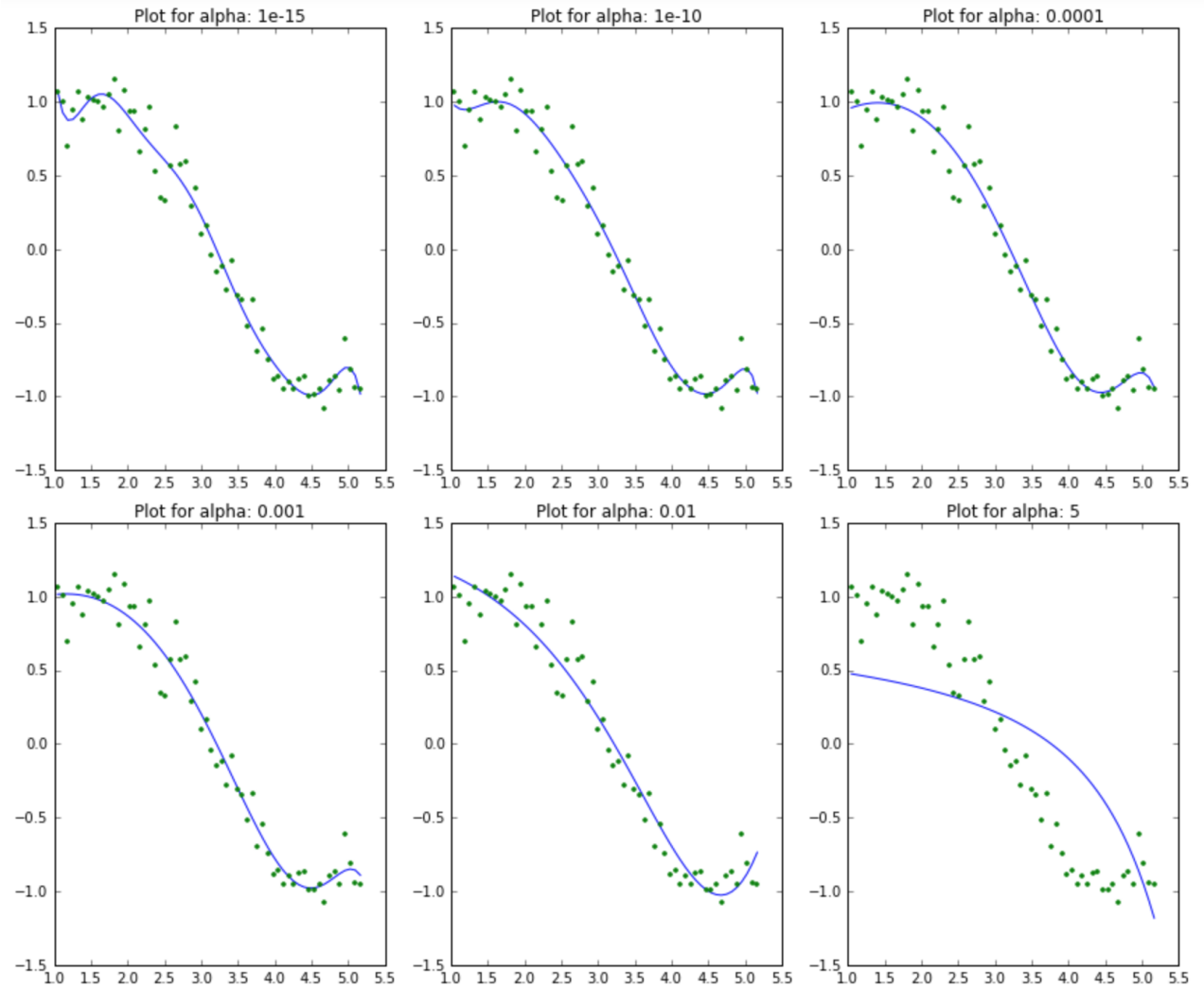

Figura 12: Efecto parámetro λ en la regularización tipo Ridge [19].

Para determinar el valor correcto del parámetro λ se emplea validación. La validación es una técnica clave en el aprendizaje automático, ya que permite además de determinar una correcta regularización, determinar una buena hipótesis y una estimación del error E_{out}.

Empezamos a explicar la validación con un caso sencillo, en el que contamos con un único dataset y empleamos un único modelo de aprendizaje.

El primer paso que debemos realizar en la validación es dividir los datos en dos partes, una de entrenamiento y otra de test. Dependiendo de las fuentes empleadas se puede leer que los valores para esta división oscilan entre el 70-80% para los datos de entrenamiento y entre el 30-20% para los datos de test. De este modo podremos generar nuestra hipótesis *g* a partir de los datos de entrenamiento y aplicar esta hipótesis sobre los datos guardados para test y obtener un error que denominaremos E_{test}.

En este punto, se puede asegurar, siempre que el conjunto de test sea lo suficientemente grande, que este E_{test} será similar a cualquier E_{out} que se pueda obtener con nuevos datos. Por otro lado, cuanto mayor sea en número de muestras en el dataset de entrenamiento podremos obtener una mejor hipótesis, por tanto, una vez obtenido el E_{test} podremos reentrenar nuestro modelo empleando ahora todos los datos que tenemos con el fin de obtener una hipótesis *g* mejor y por ello un E_{out} menor.

En resumen, para realizar validación con un único modelo y un único modelo de aprendizaje debemos seguir los siguientes pasos:

1. Dividir los datos en entrenamiento (70-80%) y test (30-20%)
2. Entrenar el modelo con los datos de entrenamiento y generar la hipótesis *g*.
3. Aplicar la hipótesis obtenida en los datos reservados para test con el fin de calcular el error empleando este modelo.
4. Reentrenar el modelo empleando la totalidad de los datos, es decir, volviendo a unir datos de entrenamiento y test (y este sería el modelo final que se emplearía).

El siguiente paso será emplear más de un modelo de aprendizaje para seleccionar el mejor de ellos. Para ello podríamos emplear el mismo criterio seguido hasta ahora y quedarnos con el que menor E_{test} presente. Imaginemos que empleamos 1.000 modelos diferentes, de esta forma por mera probabilidad habrá algún modelo que para la distribución de datos que se realiza entre entrenamiento y test, obtenga mejores resultados para E_{test} pero posteriormente peores en el E_{out}.

En este caso estamos ante un problema de sesgo de selección ya que hemos usado los datos de test para comparar entre distintos modelos, es decir, estamos incluyendo los datos de test en el proceso de aprendizaje.

En este punto, se plantea una nueva solución, volver a dividir el dataset, ahora en tres conjuntos de datos, uno de entrenamiento (60% de las muestras), como en el caso anterior, emplearemos estos datos para generar las hipótesis *g* de cada modelo (para M modelos, tendremos $g_1, g_2, \ldots, g_M$ hipótesis). El segundo conjunto de datos, al que denominamos datos de validación (20% de las muestras), se emplearía para seleccionar el modelo con el menor error, en este caso denominado E_{val} y finalmente el último conjunto de datos será el de test (20% de las muestras) con el que podremos determinar el E_{test} y por tanto estimar el E_{out} que tendrá el modelo seleccionado.

En resumen, si queremos realizar validación con un único conjunto de datos y varios modelos de aprendizajes, debemos seguir los siguientes pasos:

1. Dividir los datos en entrenamiento (60%), validación (20%) y test (20%).
2. Entrenar cada uno de los M modelos (H_1, H_2, …, H_M) empleando los datos de entrenamiento. De este modo se obtienen M hipótesis ($g_1, g_2, \ldots, g_M$).

3. Aplicar las hipótesis obtenidas en los datos de validación. Calcular el E_{val} de cada modelo y seleccionar el modelo cuyo error sea menor. A este modelo se le denomina H_*.
4. Reentrenar el modelo seleccionado empleando los datos de entrenamiento y validación (80% de los datos). Con esto obtenemos una hipótesis g_*.
5. Aplicar la hipótesis obtenida g_* en los datos reservados para test con el fin de calcular el error empleando este modelo.
6. Reentrenar el modelo empleando la totalidad de los datos, es decir, volviendo a unir datos de entrenamiento, validación y test.

Gracias a este esquema de trabajo conseguimos evitar el problema de sesgo de selección indicado anteriormente, pero nos plantea un nuevo problema. Dividir el dataset en tres conjuntos diferentes de datos puede hacer que nos quedemos con una cantidad de datos insuficiente para cualquiera de las fases de entrenar, validar o testear.

Finalmente, para subsanar este último problema que aparece se emplea la validación cruzada. En este último caso, es decir, empleando varios modelos podemos ver cómo funciona la validación cruzada en la siguiente imagen:

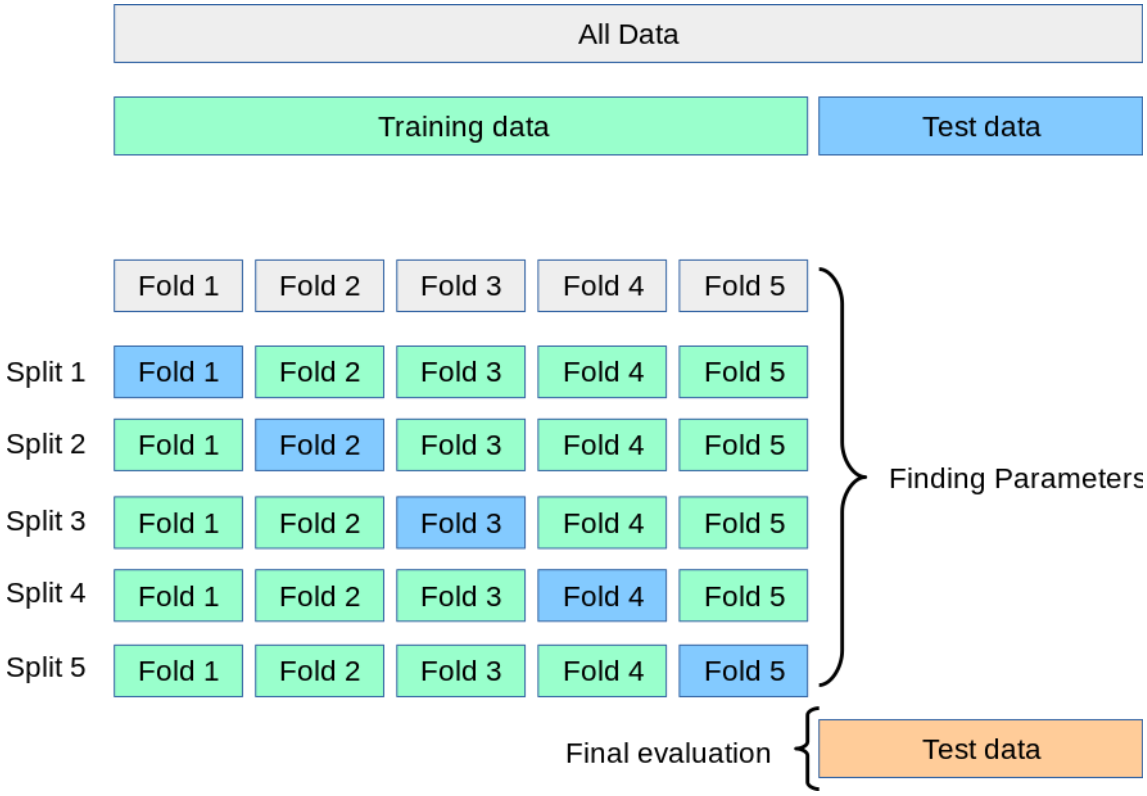

Figura 13: Validación cruzada [10].

El primer paso en la validación cruzada es dividir el dataset en datos de entrenamiento y test. A partir de ahí, trabajaremos con el conjunto de datos de entrenamiento, como vemos en la imagen la validación cruzada se realiza en N iteraciones, (en este ejemplo son 5 iteraciones, pero puede ser otro valor).

Los datos de entrenamiento se dividen en N pliegues y habrá N iteraciones. En cada una de estas iteraciones, se emplean 9 pliegues para entrenar el modelo (estos 9 pliegues equivalen a los datos de entrenamiento) y se aplica la hipótesis obtenida en el

pliegue restante (equivalente a los datos de validación) obteniendo de este modo en cada una de las iteraciones un E_{val}. Para determinar el E_{val} general de cada modelo se realiza la media del obtenido en cada una de las iteraciones.

Gracias al conjunto de datos de test que hemos dejado sin emplear en la selección del modelo podemos evitar problemas al detectar posibles anomalías en ciertos casos. Por otro lado cabe destacar que empleando validación cruzada no estamos empleando todos los datos disponibles en la selección del modelo pero es un problema que se puede ignorar si disponemos de un conjunto de datos grande.

4. Análisis y visualización de datos.

En este capítulo se comienza a trabajar con los dataset recopilados durante la fase de búsqueda de datos. En primer lugar, procedemos a realizar un análisis de los datos disponibles sobre Google Ads.

En Google Ads nos centraremos en mostrar la evolución de las impresiones causadas por las distintas palabras claves de una misma campaña a lo largo del tiempo estudiado en este proyecto. A continuación, se realiza un análisis tanto del CTR cómo del CPC de cada campaña y cada uno de sus mercados. Finalmente se realiza un *Word Cloud* en el que se visualiza la cantidad de veces que ha estado activa tanto una palabra individualmente como cada palabra clave en conjunto.

A continuación, se emplea el dataset de Criteo. En este caso haremos un estudio enfocado a determinar las características más importantes del conjunto de datos con el fin de crear un modelo predictivo capaz de determinar la probabilidad que tiene un anuncio de ser clicado.

4.1 Google Ads.

El estudio de este conjunto de datos se ha dividido en dos fases. En la primera de estas fases se ha procedido al análisis de cada una de las dos campañas individualmente, en el que se ha realizado un seguimiento de las impresiones causadas por cada una de las palabras claves de cada campaña a lo largo del tiempo disponible, así como un estudio del CTR y CPC de las palabras claves y la visualización de un Word Cloud.

Esta primera fase se realiza para ambas campañas de forma análoga, siguiendo en ambos casos el mismo proceso, con pequeñas diferencias derivadas de las características de cada una de las campañas. Por este motivo se refleja en esta memoria únicamente el proceso realizado en una de las campañas, la campaña 2.

En la segunda fase, se compara el estudio realizado sobre el CTR y el CPC entre ambas campañas.

4.1.1 Lectura y preparación de los datos.

En primer lugar, hay que recordar la estructura del dataset de Google Ads, y en este caso la estructura del dataset *bids-dataset-2-20130214* correspondiente a la campaña 2 (similar a *bids-dataset-3-20130214* de la campaña 3).

Este dataset contiene dos subdirectorios, uno por mercado, el estadounidense y el británico. Cada uno de estos directorios incluye un fichero csv con la información sobre la campaña por cada día estudiado. A su vez, cada uno de estos ficheros csv tiene varias líneas con información general y posteriormente una línea con la información de cada palabra clave de la campaña en ese día concreto. Una vez hecho este breve recordatorio se puede entender mejor el proceso realizado para la lectura de datos.

Para almacenar estos datos generamos en primer lugar tres *dataframes*, empleando la librería *pandas*, uno por cada mercado y el tercero de ellos será la unión de ambos. Debido a las características de nuestro dataset, se procede a leer uno a uno cada fichero csv de cada mercado y se almacena todos los datos en los *dataframes* ya mencionados.

De forma paralela a este proceso, se genera una nueva columna de datos que denominaremos *Date*, la fecha. Esta información se obtiene del nombre de cada fichero, por tanto, cada vez que se lee uno de estos ficheros csv se añade a cada una de las filas que contiene la fecha que le corresponde.

En el proceso de lectura hemos encontrado dos problemas, ficheros csv nombrados con el mismo día, pero con información diferente (aunque muy similar), por ejemplo, en el mercado británico de la campaña 2 los días 22/07/2012, 30/10/2012 y 25/11/2012 están duplicados, para evitar problemas en este aspecto se ha eliminado el fichero csv cuyo número menor incluido en su nombre de fichero era menor. Por otro lado, hay ficheros con información totalmente en blanco, por ejemplo, el fichero del día 05/06/2012 del mercado británico en la campaña 2, los cuales también han sido eliminados.

En este punto tenemos tres *dataframes*, uno por mercado y otro como conjunto de ambos, en cada uno de ellos hay información general de cada día, así como información de cada palabra clave cada día. Para facilitar la explicación nos centraremos en uno de los mercados individualmente y en el conjunto de ellos.

A partir de este punto, nos vamos a centrar en el mercado británico. Con el fin de facilitar la comprensión de estos datos, generamos un *dataframe* por cada palabra clave, en el que almacenamos la información de cada una de estas palabras claves a lo largo de los días estudiados. Para almacenar estos nuevos *dataframes* creamos un diccionario, cuyas claves serán las palabras claves de la campaña y cuya información serán estos *dataframes.* Cada uno de estos *dataframes* tendrá seis columnas, *Keyword, Clicks, CTR, Cost, Impressions* y *Date*, es decir, información sobre las palabras claves, clics, CTR, coste, impresiones y fecha.

En la siguiente imagen podemos ver una breve muestra de uno de los *dataframes*. En concreto, se presenta la información de los cinco primeros días estudiados de la palabra clave *olympics* en el mercado británico.

	Keyword	Clicks	CTR	Cost	Impressions	Date
0	olympics	2003.47	3.0%	5112.81	67861.0	2012-05-24
1	olympics	2172.97	3.2%	5532.68	67166.0	2012-05-25
2	olympics	1999.95	2.9%	5093.48	67803.0	2012-05-26
3	olympics	1973.85	2.9%	5029.42	67803.0	2012-05-27
4	olympics	1986.45	2.9%	5056.00	67803.0	2012-05-29

Figura 14: Muestra dataframe olympics. Mercado UK campaña 2.

El siguiente paso que hemos realizado es agrupar la información de cada una de las palabras claves, de este modo conseguimos información general de cada palabra clave, en concreto se generan un *dataframe* con las siguientes columnas:

- *Keyword*: palabra clave.
- *Count*: número de días que está activa la palabra clave.
- *Total Impressions*: número total de impresiones generadas a lo largo de los días estudiados.
- *Total Clicks*: número total de clics generados a lo largo de los días estudiados.
- *Total Cost*: coste total a lo largo de los días estudiados.
- *CTR calculate*: media del CTR, calculado como *Total Clicks / Total Impressions*.
- *CPC calculate*: media del CPC, calculado como *Total Cost / Total Clicks*.

	Keyword	Count	Total Impressions	Total Clicks	Total Cost	CTR calculate	CPC calculate
0	secure online back up	13	401.0	3.47	30.96	0.865337	8.922190
1	agile management software	168	11480.0	292.31	1997.18	2.546254	6.832404
2	crm for financial	0	0.0	0.00	0.00	0.000000	0.000000
3	disaster recovery planning for it	0	0.0	0.00	0.00	0.000000	0.000000
4	tracking a vehicle	92	10014.0	317.62	1140.15	3.171760	3.589667
5	applications in the cloud	26	852.0	19.26	134.47	2.260563	6.981828

Figura 15: Muestra de información general de cada palabra clave. Mercado UK campaña 2.

4.1.2 Análisis de las impresiones.

El primer análisis realizado con los datos de Google Ads se ha centrado en las impresiones. Para ello, lo primero que se ha hecho ha sido seleccionar las 20 palabras clave que más impresiones totales han generado. Para determinar estas palabras claves se ha ordenado el *dataframe* con información general, teniendo en cuenta la columna *Total Impressions*. Hay que tener en cuenta que, aunque nos estamos centrando en el mercado británico, este proceso se ha repetido para cada uno de los *dataframes* generales que tenemos, por tanto, las palabras claves empleadas en cada uno de estos análisis podrá ser diferente.

Una vez seleccionadas las palabras claves, se crea una versión resumida del diccionario palabra clave-*dataframe* que contendrá únicamente estas palabras claves. Con el nuevo diccionario se crea un *dataframe* de impresiones cuyas columnas son los días estudiados y tiene una fila por cada palabra clave donde se incluye información de las impresiones. Es decir, tenemos la información de las impresiones por día y palabra clave. Podemos ver una muestra de este *dataframe* en la siguiente figura:

	2012-05-24	2012-05-25	2012-05-26	2012-05-27	2012-05-29	...	2013-02-06	2013-02-07	2013-02-08	2013-02-12	2013-02-14
ipad	145066	130541	131762	131762	131762	...	196493	172040	176328	171846	190511
iphone 4	136197	102641	117106	117106	117106	...	307536	261972	260830	242345	288033
facebook	107781	102152	101053	101053	101053	...	180402	152885	144088	137940	146059
eminem	53408	50568	53417	53417	53417	...	68160	53576	45729	35239	43558
loans	152136	142359	130295	130295	130295	...	137066	123517	128557	136123	149573

Figura 16: Muestra de las Impresiones generadas por palabra clave y día. Mercado UK campaña 2.

Del mismo modo, también se ha creado un *dataframe* indicando la posición de cada palabra clave cada día, teniendo en cuenta el número de impresiones. Ordenando las palabras claves del 1 al 20 siendo el 20 la palabra clave con el máximo número de impresiones generadas.

En este momento podemos realizar las primeras visualizaciones, mostrando la evolución del número de impresiones generadas por las palabras claves a lo largo de todos los días. Empleando estas visualizaciones es difícil comprender la evolución del mercado, ya que se muestra demasiada información, y no se aprecia correctamente. Por tanto, empleando el diccionario resumido que se ha creado se agrupan las impresiones generadas en cada mes, dando lugar a unos gráficos mucho más claros y legibles.

A continuación, se va a mostrar la evolución del mercado mes a mes, teniendo en cuenta la posición de cada palabra clave respecto al número de impresiones que genera.

Campaña 2, mercado británico.

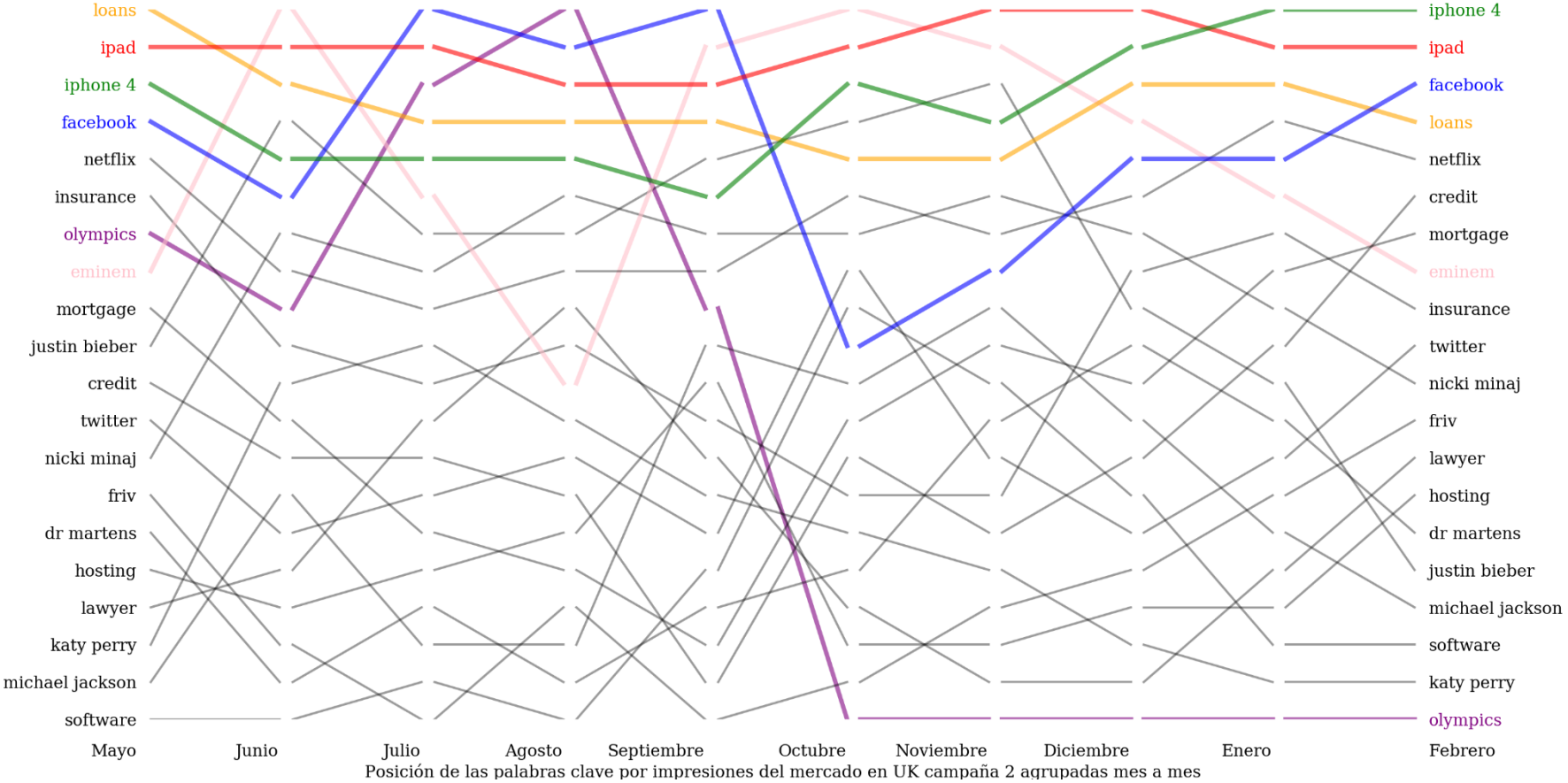

Figura 17: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado UK campaña 2.

En esta imagen podemos ver las 20 palabras claves con mayor número de impresiones totales generadas en el mercado británico en la campaña 2. Destacamos con diversos colores las 5 palabras claves más destacadas, así como un caso especial:

- *loans* (préstamo) en naranja.
- *ipad* en rojo.
- *iphone 4* en verde.
- *facebook* en azul.
- *eminem* en rosa.
- *olympics* (olimpiadas) en morado.

La palabra clave *olympics* es un caso especial, como se puede apreciar entre los meses de mayo de 2012 y agosto del mismo año, es una de las palabras claves que generan más impresiones cada mes. Sin embargo, a partir del mes septiembre reduce su cantidad de impresiones hasta ser la palabra clave que menos impresiones genera dentro de las palabras claves estudiadas.

Este caso se explica por la celebración de las olimpiadas de Londres, Reino Unido, en el mismo año que se lleva a cabo este estudio. Las olimpiadas realizaron su apertura el 27 de julio de 2012 y finalizaron el 12 de agosto del mismo año. Por tanto, la celebración de este evento coincide con la mayor generación de impresiones de la palabra clave *olympics*, y su finalización se refleja en la drástica reducción de impresiones generadas.

Concretamente, las impresiones generadas y su posición (dentro de las 20 palabras clave con más impresiones totales) mes a mes por esta palabra clave se ven en la siguiente tabla:

	Impresiones generadas	Posición
Mayo 2012	491.960	14
Junio 2012	1.945.916	12
Julio 2012	4.608.157	18
Agosto 2012	21.352.826	20
Septiembre 2012	2.188.120	12
Octubre 2012	576.734	1
Noviembre 2012	262.850	1
Diciembre 2012	153.860	1
Enero 2012	106.403	1
Febrero 2012	24.421	1

Tabla 3: Impresiones generadas y posición mes a mes de la palabra clave olympics. Mercado UK campaña 2.

Hay que destacar que los meses de mayo y febrero no están completos, en mayo hay datos recogidos de 7 días mientras que en febrero hay datos de 9 días.

Campaña 2, mercado estadounidense.

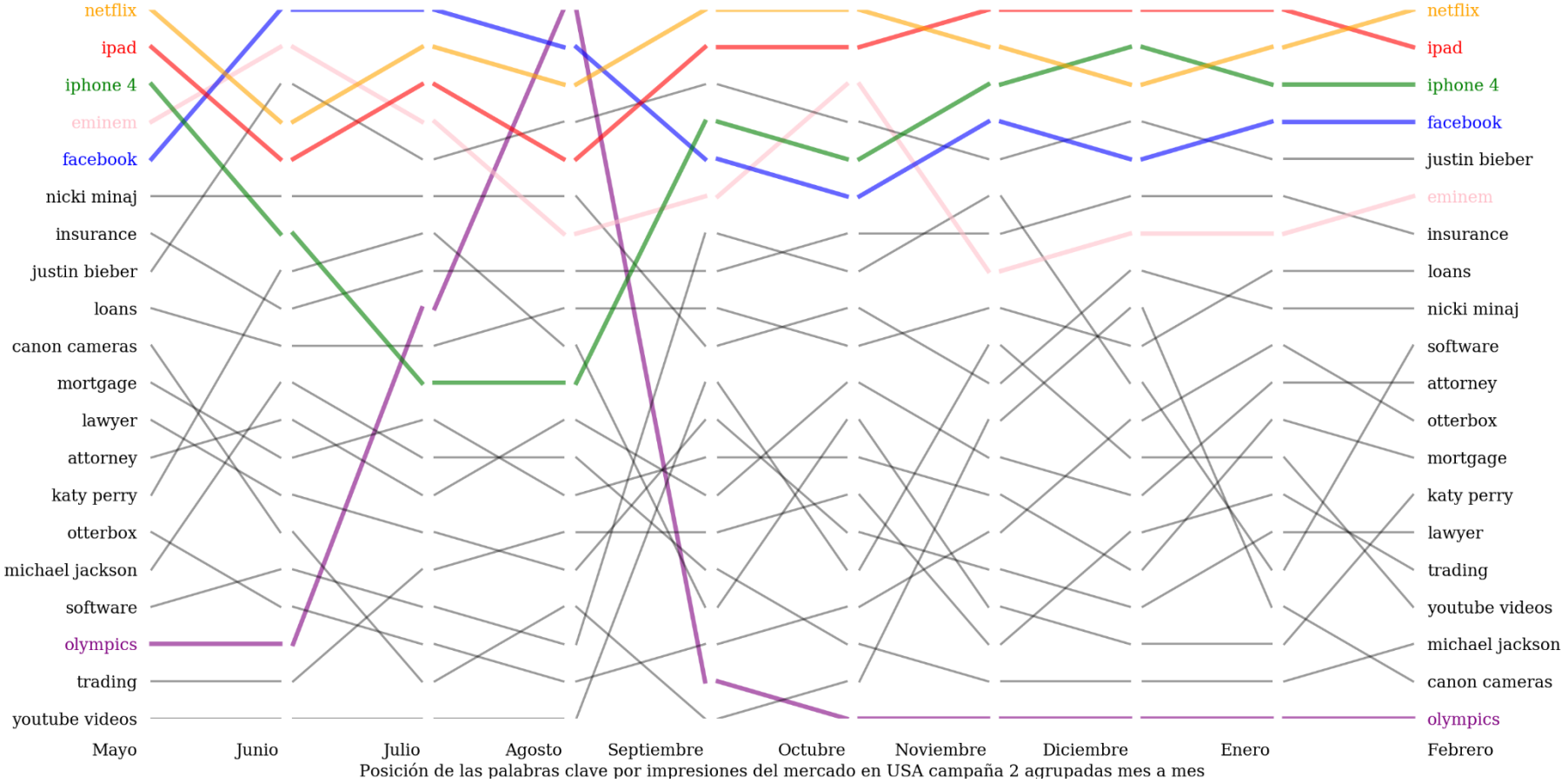

Figura 18: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado US campaña 2.

De forma análoga al mercado británico vemos las 20 palabras claves con mayor número de impresiones totales generadas en el mercado estadounidense. Se puede apreciar que ocurre lo mismo que en el caso anterior con la palabra clave *olympics*, destacando en las fechas en las que se llevó a cabo las olimpiadas.

Al igual que en el caso anterior, destacamos con distintos colores las palabras claves más relevantes. El único cambio respecto al mercado británico es la aparición de la palabra clave *netflix* en las primeras posiciones en lugar de *loans*:

- *netflix* en naranja.
- *ipad* en rojo.
- *iphone 4* en verde.
- *facebook* en azul.
- *eminem* en rosa.
- *olympics* (olimpiadas) en morado.

Campaña 2, mercado conjunto.

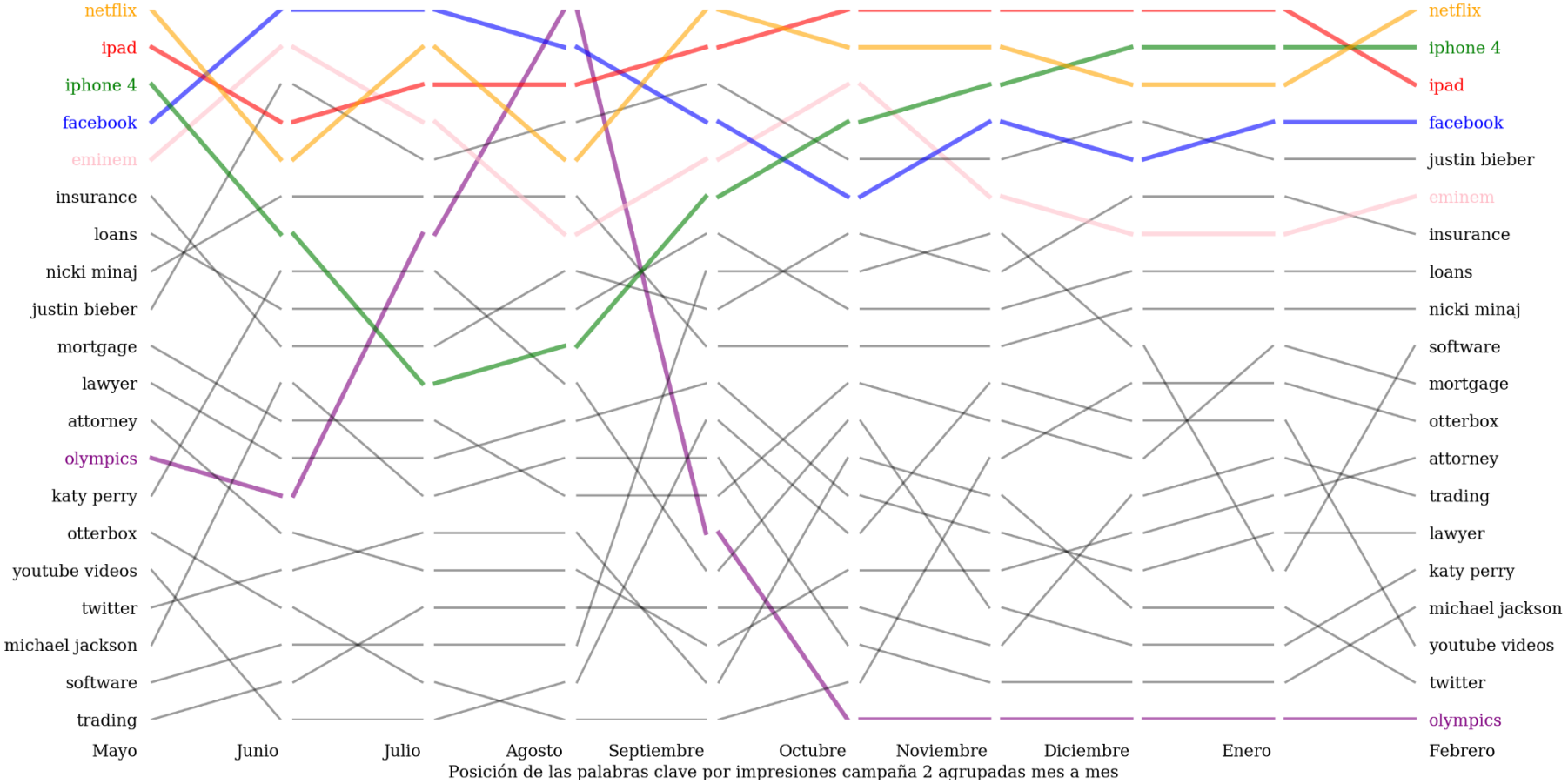

Figura 19: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado conjunto campaña 2.

En primer lugar, vamos a ver que palabras claves aparecen en cada mercado. Hay 15 palabras claves que aparecen entre las que generan más impresiones en ambos mercados, estas son: *ipad, iphone 4, facebook, eminem, loans, olympics, justin bieber, netflix, nicki minaj, insurance, mortgage, katy perry, michael jackson, lawyer* y *software*. Por otro lado, las palabras clave que generan más impresiones únicamente en el mercado británico son *credit, twitter, dr martens, hosting* y *friv*, mientras que las que generan más impresiones sólo en el mercado estadounidense son *attorney, canon cameras, otterbox, youtube videos* y *trading.*

Estudiando el mercado conjunto podemos apreciar que es muy similar al mercado estadounidense (sólo varia una palabra clave entre ambos). Esto se debe a que el mercado de Estados Unidos es mucho más grande que el mercado en Reino Unido. Mientras que el total de las impresiones generadas por todas las palabras claves en el mercado británico. es de 530.505.923, en el mercado estadounidense asciende a 2.525.243.645, es decir, casi 5 veces más impresiones generadas.

Campaña 3, mercado británico.

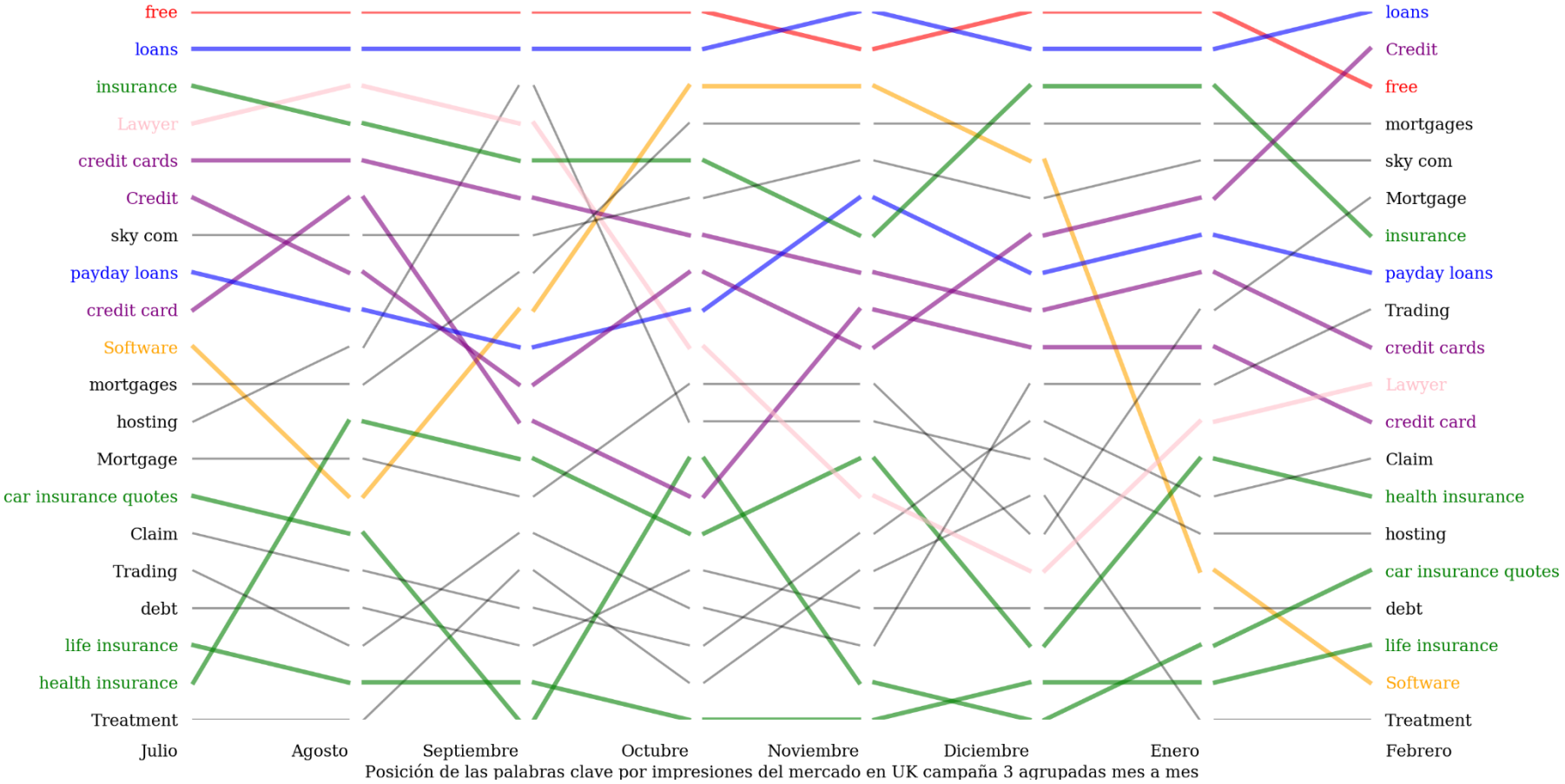

Figura 20: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado UK campaña 3.

En esta imagen podemos ver las 20 palabras claves con mayor número de impresiones totales generadas en el mercado británico en la campaña 3. Destacamos con diversos colores las 6 palabras claves más destacadas:

- *free* (gratis) en rojo.
- *loans (préstamo)* en azul. Hay dos palabras claves en las que se incluye la palabra *loans* y por tanto se muestran ambas en azul.
- *insurance* (seguro) en verde. Hay cuatro palabras claves en las que se incluye la palabra *insurance* y por tanto se muestran todas en verde.
- *Lawyer* (abogado, abogada) en rosa.
- *Credit* (crédito) en morado. Hay tres palabras claves en las que se incluye la palabra *credit* y por tanto se muestran todas en morado.
- *Software* en naranja.

En esta campaña podemos apreciar cómo se buscan sectores muy concretos a través del empleo de palabras claves muy especializadas. Por ejemplo, analizando la palabra *insurance*, vemos que hay cuatro palabras claves que incluyen esta palabra, las cuales son *insurance* que sería su versión general, *car insurance quotes*, *life insurance* y *health insurance* que son palabras claves más especializadas creando anuncios diferentes si se busca un seguro de coche, de vida o de salud.

Por otro lado, cabe destacar la palabra clave *free*, para ello vamos a mostrar la gráfica en la que aparece las impresiones generadas mes a mes de estas 20 palabras clave. Como podemos apreciar en la siguiente imagen esta palabra clave genera muchas más impresiones que cualquiera de las otras palabras claves, que apenas se pueden apreciar en la gráfica debido al valor tan elevado de la palabra clave *free*.

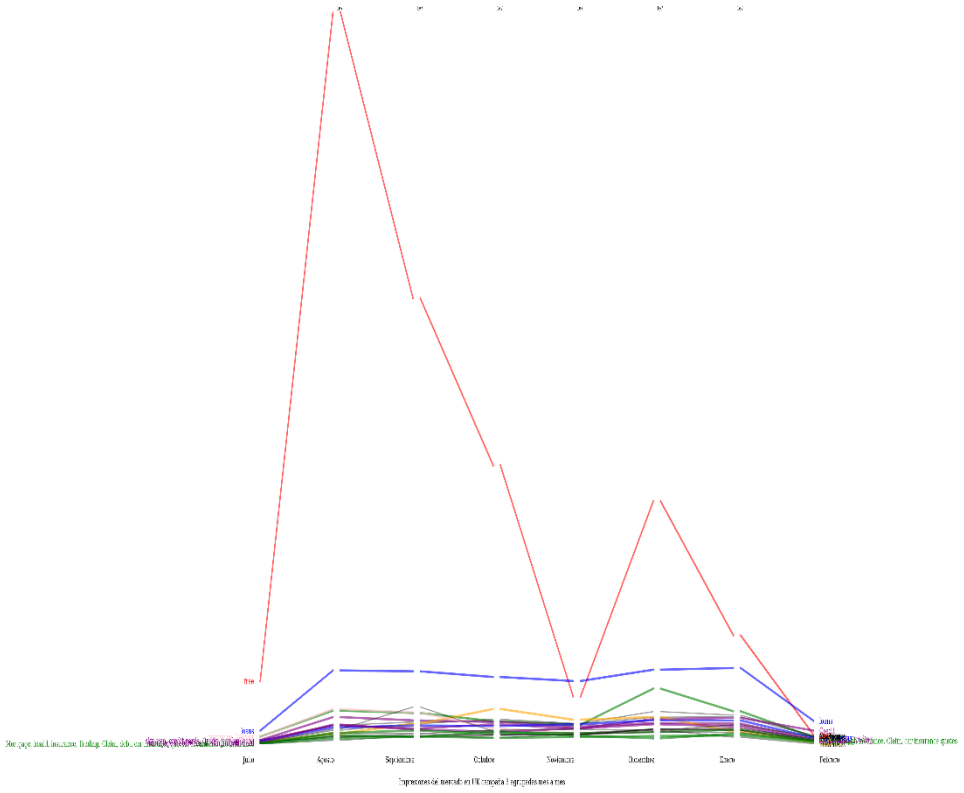

Figura 21: Impresiones generadas por las palabras mes a mes. Mercado UK campaña 3

Para observar claramente la diferencia en el número de impresiones generadas entre la palabra clave *free* y el resto, vamos a mostrar las cinco palabras claves con el mayor número de impresiones generadas total:

Palabra clave	*free*	*loans*	*insurance*	*mortgages*	*sky com*
Número de impresiones	97.125.546	23.720.383	10.976.017	8.058.482	7.925.697

Tabla 4: Palabras claves con el mayor número de impresiones generadas total.

Campaña 3, mercado estadounidense.

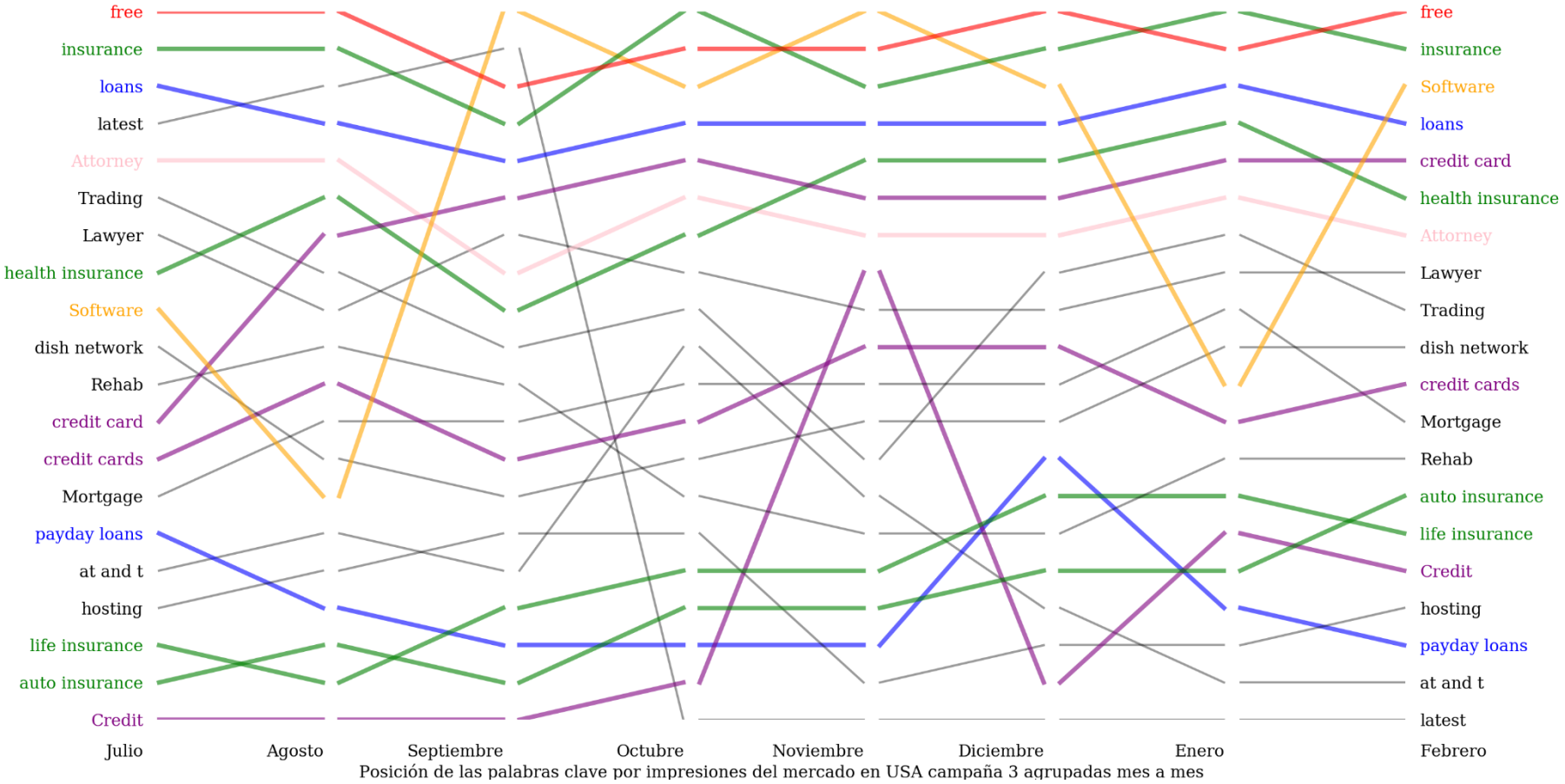

Figura 22: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado US campaña 3.

Las palabras destacadas en este caso son iguales que en el mercado británico, salvo la palabra clave *Lawyer* que ahora no se resalta en rosa, y la sustituye la palabra *Attorney* aunque su significado es el mismo (abogado o abogada).

Como vemos se sigue dando la misma especialización que en el mercado británico, incluyendo varias palabras claves con la misma palabra centrando de este modo cada anuncio a un sector concreto. Por otro lado, la palabra clave *free* no genera la diferencia de impresiones vista en el mercado británico, y aunque sigue siendo la palabra clave con más impresiones no supone una diferencia destacable con las siguientes palabras claves.

Campaña 3, mercado conjunto.

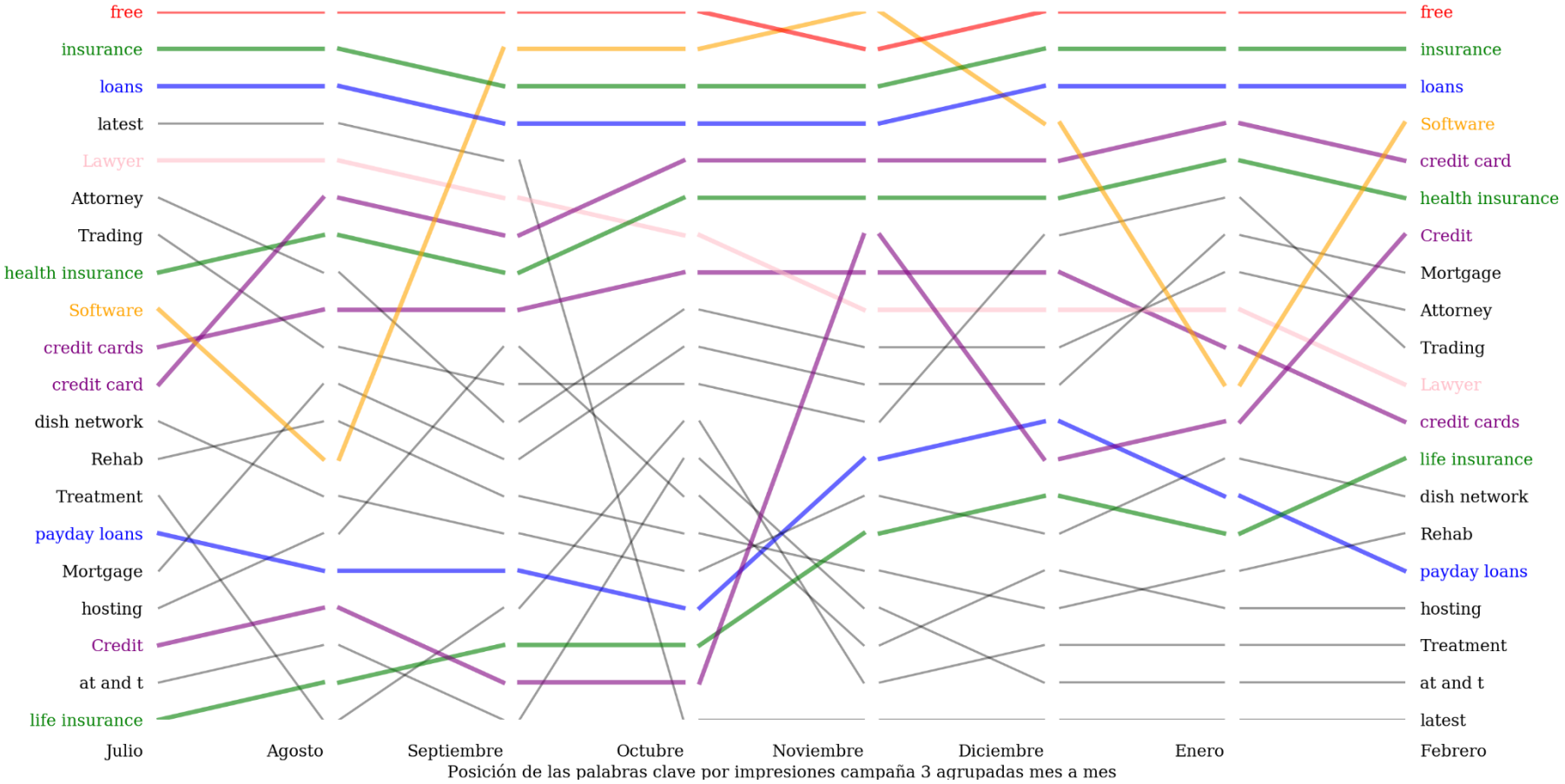

Figura 23: Posición palabras claves teniendo en cuenta el número de impresiones mes a mes. Mercado conjunto campaña 3.

Al igual que hicimos con la campaña 2, se comparan las palabras claves que generan más impresiones en cada mercado. De esta manera observamos que hay 14 palabras claves coincidentes en ambos mercados y 6 diferentes en cada uno de ellos.

- Palabras claves que estudiamos en ambos mercados: *free, insurance, Software, loans, credit card, health insurance, Lawyer, Trading, Mortgage, credit cards, life insurance, Credit, hosting* y *payday loans*
- Palabras claves estudiadas exclusivamente en el mercado británico: *mortgages, sky com, Claim, debt, car insurance quotes* y *Treatment*.
- Palabras claves estudiadas sólo en el mercado estadounidense: *Attorney, latest, dish network, Rehab, at and t* y *auto insurance*.

De forma similar a lo visto en la campaña 2, y por la misma causa, en esta campaña el mercado global también se comporta de forma muy similar al mercado estadounidense, sólo hay una palabra clave diferente entre el mercado global y el mercado estadounidense.

4.1.3 Word Cloud.

Un Word Cloud, o nube de palabras, es una representación visual de las palabras que conforman un texto. En esta representación, el tamaño de la palabra es proporcional respecto a las veces que aparece en dicho texto. En este caso, el texto estará formado por cada palabra clave teniendo en cuenta únicamente los días que ha estado activa, independientemente de la cantidad de impresiones que genera cada día.

Hemos realizados un Word Cloud por cada mercado y campaña mostrando las palabras individualmente. Hay que tener en cuenta que en las distintas campañas hay palabras que aparecen en singular y en plural, o en mayúscula y minúscula, estos casos se han tratado del mismo modo, agrupándose todas ellas como una única palabra o palabra clave, es decir, por ejemplo, las palabras *Loan, Loans, loan* y *loans* se considerarán como una única palabra.

Para realizar estas nubes de palabras hemos usado la librería *wordcloud*, que entre otras cosas, nos permite establecer palabras a no tener en cuenta a través del parámetro *stopwords* gracias al cual eliminamos palabras comunes y que no aportan ningún tipo de información como preposiciones *(at, by*, …), pronombres (*I, you*, …), etc [20].

Campaña 2, mercado británico.

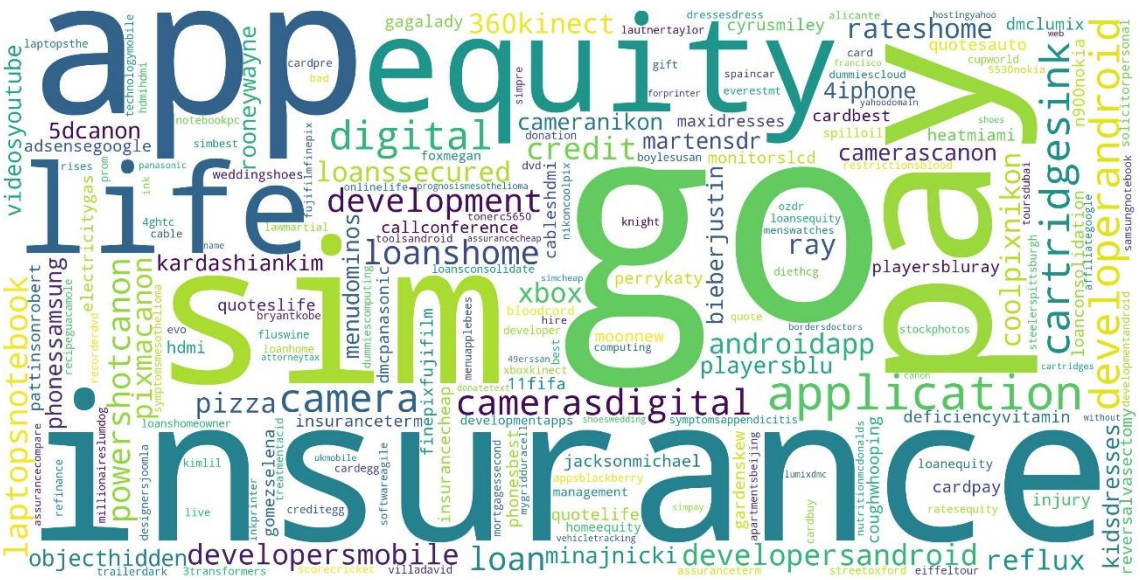

Figura 24: Word Cloud mercado británico campaña 2.

Campaña 2, mercado estadounidense.

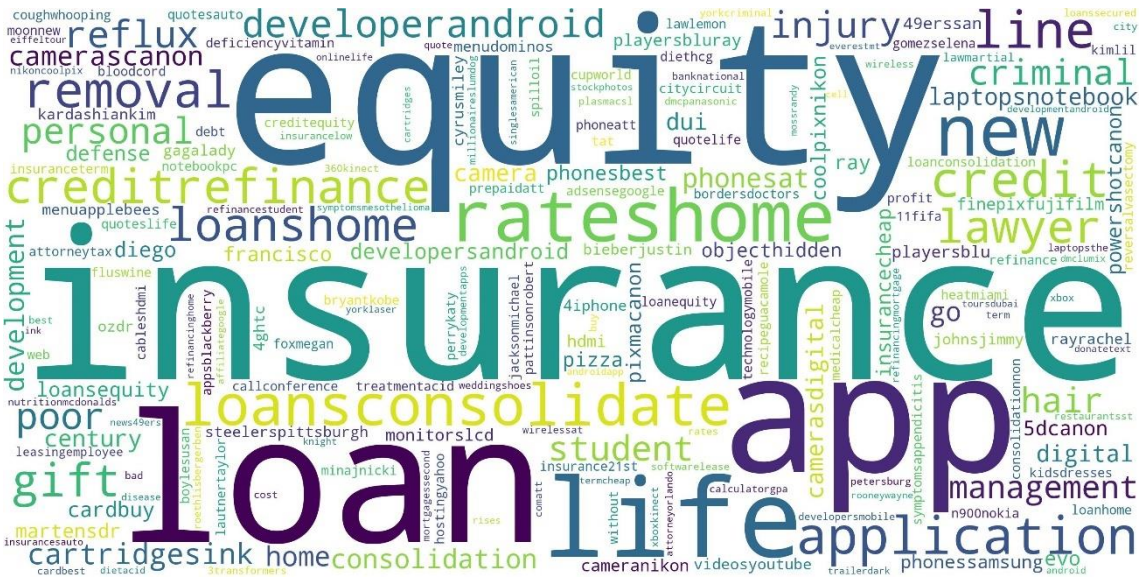

Figura 25: Word Cloud mercado estadounidense campaña 2.

Campaña 2, mercado conjunto.

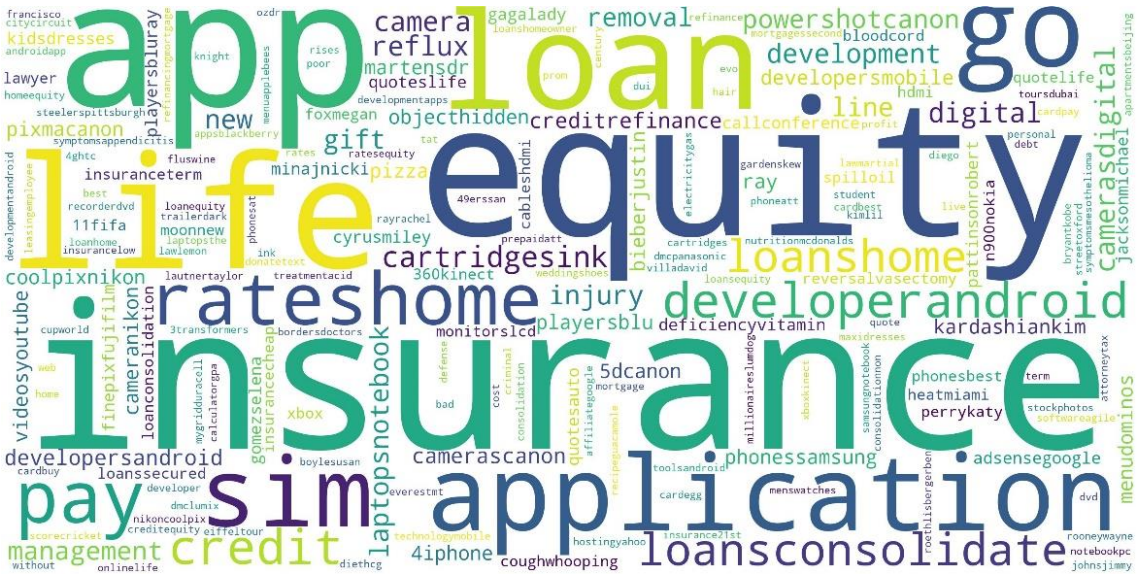

Figura 26: Word Cloud mercado conjunto campaña 2.

En los tres casos podemos apreciar en primer plano palabras que ya habíamos visto en el estudio de las impresiones como *insurance* o *loan*, algo que se podía considerar predecible. Sin embargo, también aparecen muchas palabras que no se habían visto. Esto se debe principalmente a dos factores:

- Tenemos en cuenta únicamente que una palabra clave haya estado activa un día o no, independientemente de la cantidad de impresiones que esta palabra clave.
- Hay palabras repetidas dentro de las palabras claves con el fin de buscar la especialización de los anuncios en distintos sectores. Por ejemplo, la palabra *life* que podemos ver en los tres Word Cloud aparece en 10 palabras claves dentro de esta campaña.

Podríamos hacer este mismo proceso teniendo en cuenta sólo las palabras claves, pero el resultado es poco significativo, ya que en la campaña 2 hay más de 120 palabras claves activas al menos 240 de los 241 días de los que tenemos datos disponibles. Por tanto, el resultado es un Word Cloud con todas las palabras del mismo tamaño.

Si analizamos la campaña 3, tenemos resultados muy similares, aparecen algunas palabras ya vistas en el estudio realizado sobre las impresiones generadas por las palabras claves, y otras que aparecen en varias palabras claves. Al igual que sucedía con la campaña 2, un Word Cloud de palabras claves no tiene mucho sentido ya que hay más de 200 palabras claves activas todos los días con información disponible.

4.1.4 Estudio del CTR

Las siglas CTR provienen de *Click Through Rate* y nos indica el número de clics que se obtienen en un anuncio respecto al número de impresiones que genera. Este parámetro se expresa en porcentaje y la forma de calcularlo es muy sencilla, basta con obtener el cociente entre los clics de un anuncio y las impresiones generadas (multiplicando este resultado por 100 obtendremos el porcentaje deseado).

El CTR es una métrica muy importante para medir el impacto que produce un anuncio, obviamente, el objetivo de cualquier anuncio es aumentar el CTR al máximo posible.

Para reflejar el CTR de cada uno de los mercados hemos realizado diversos gráficos en los que podemos apreciar diferente información:

- Histograma de las palabras claves activas con el que mostramos la distribución del CTR. Para ello, en primer lugar, eliminamos aquellas palabras claves que no han generado ninguna impresión y, por tanto, ningún coste a lo largo de toda la campaña.
- Diagrama de barras, en el que se muestra el valor del CTR de cada palabra clave individualmente. Debido a la cantidad de palabras claves disponibles en las campañas se ha realizado una selección:
    - Palabras claves observadas en el estudio de las impresiones generadas. Mostramos el CTR de cada una de las 20 palabras claves con mayor número de impresiones.
    - Palabras claves con mayor CTR (un total de 20 palabras clave).

Posteriormente, se ha comparado por un lado los histogramas entre los mercados de una misma campaña, y por otro lado ambas campañas globalmente.

Campaña 2, mercado británico.

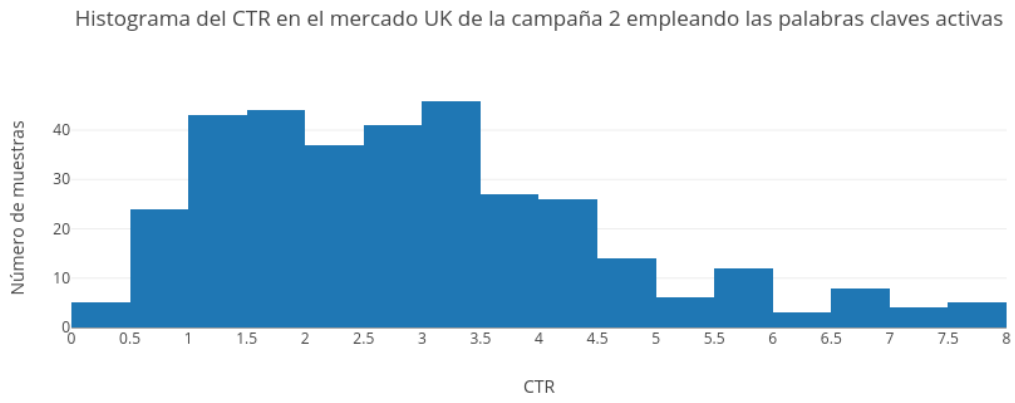

Figura 27: Histograma del CTR en el mercado británico, campaña 2.

En este gráfico vemos la distribución del valor del CTR en el mercado británico de la campaña 2. El eje X representa el valor del CTR, agrupados en grupos de 0.5, mientras que el eje Y indica el número de palabras claves existentes en cada grupo.

Podemos ver que este histograma es algo asimétrico hacia la derecha, es decir, la mayoría de las palabras claves tienen un CTR pequeño, pero hay unas pocas con un CTR mayor. No se encuentran valores atípicos en este gráfico.

En este mercado hay un total de 345 palabras claves activas, de las cuales 147, es decir, un 42.6% tienen un valor de CTR entre 1% y 3.5%, al que podemos considerar el rango medio en esta campaña. Si realizamos el cálculo del CTR de todo el mercado obtenemos un valor de 2.2%, es decir de cada 100 impresiones que genera un anuncio, se hace clic en él 2.2 veces (en media).

No hay un umbral a partir del cual podemos decir que un anuncio tiene un buen CTR, ya que depende de múltiples aspectos, como competencia en ese momento, tipo de industria a la que pertenece el producto ofertado (lo vemos al final de esta sección), etc. Aun así, si se consultan diferentes agencias de marketing digital podemos ver que a partir del 2% se considera generalmente un CTR correcto en la red de búsqueda de Google Ads.

A continuación, se muestra el CTR de las 20 palabras claves con mayor valor en esta métrica y el CTR de las 20 palabras claves que han generado mayor número de impresiones en esta campaña.

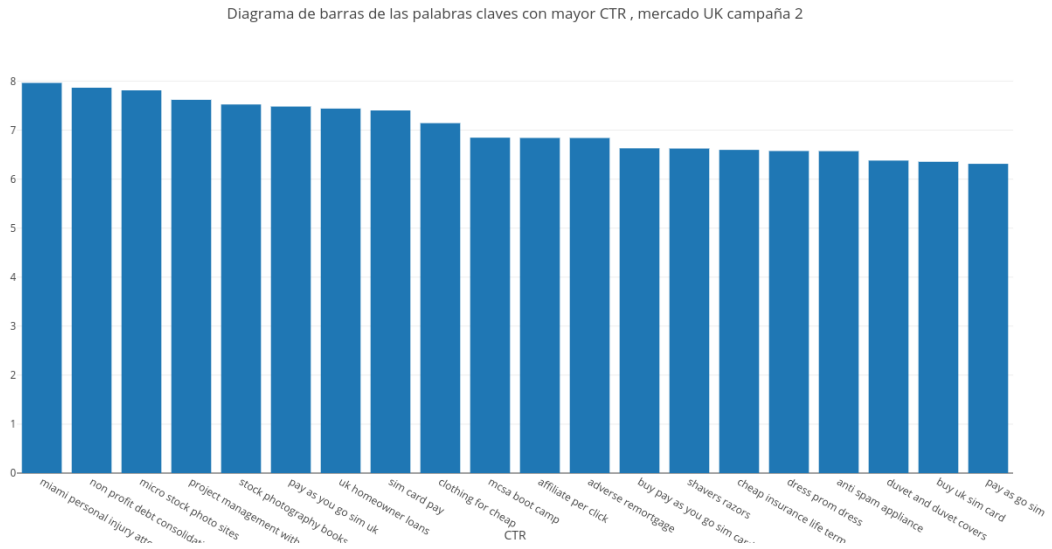

Figura 28: Valor del CTR en las palabras claves con mayor CTR. Mercado británico, campaña 2.

Ya que apenas se aprecia en la imagen, las palabras claves con mayor CTR son, ordenadas de mayor a menor: *miami personal injury attorney, non profit debt consolidation, micro stock photo sites, project management with scrum, stock photography books, pay as you go sim uk, uk homeowner loans, sim card pay, clothing for cheap, mcsa boot camp, affiliate per click, adverse remortgage, buy pay as you go sim card, shavers razors, cheap insurance life term, dress prom dress, anti spam appliance, duvet and duvet covers, buy uk sim card,* y *pay as go sim.*

Hay que destacar que estas palabras claves generan pocas impresiones, por ejemplo, la palabra clave con mayor CTR, es decir *miami personal injury attorney*, genera 135 impresiones en toda la campaña. Entre estas 20 palabras claves, aquella que más impresiones ha generado ha sido *pay as go sim* con un total 5.788, cantidad muy pequeña si tenemos en cuenta que haciendo la media entre las palabras claves activas nos da un valor superior al millón de impresiones por palabra clave.

Por tanto, se pueden considerar estos valores como anecdóticos, de poco valor informativo. Por ello, hemos procedido a realizar el diagrama de barras del CTR teniendo en cuenta las palabras claves con mayor número de impresiones generadas.

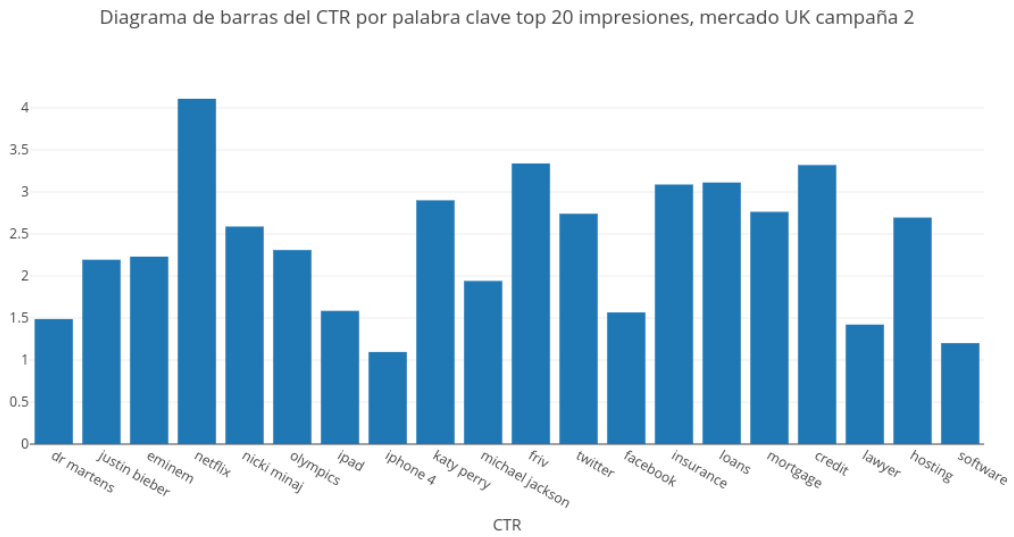

Figura 29: Valor del CTR en las palabras claves que generan más impresiones. Mercado británico, campaña 2.

En general, las palabras claves con mayor número de impresiones tienen un CTR con valor dentro del rango medio (1%-3.5%) que habíamos visto anteriormente. Concretamente sólo *Netflix* con un CTR del 4.1% está fuera de este rango de valores.

Campaña 2, mercado estadounidense.

Al igual que hemos hecho en el caso anterior, lo primero que podemos ver es la distribución del CTR a lo largo de todas las palabras claves activas en este mercado. Para ello mostramos el histograma representado en la Figura 30.

En este histograma vemos una distribución similar al mercado británico, algo asimétrico hacia la derecha. También se aprecia un rango de valores entre el que se encuentra la mayoría de las palabras claves, este rango se puede establecer para un CTR comprendido entre el 1% y el 4% e incluye a 286 de las 405 palabras claves activas, es decir un 70,6%.

Calculando el CTR medio de todo el mercado obtenemos un 2.1%, muy similar al 2.2% que obtenemos en el mercado británico. Por tanto, podemos ver, por la distribución y el valor medio del CTR que ambos mercados tienden a comportarse de forma similar en este aspecto.

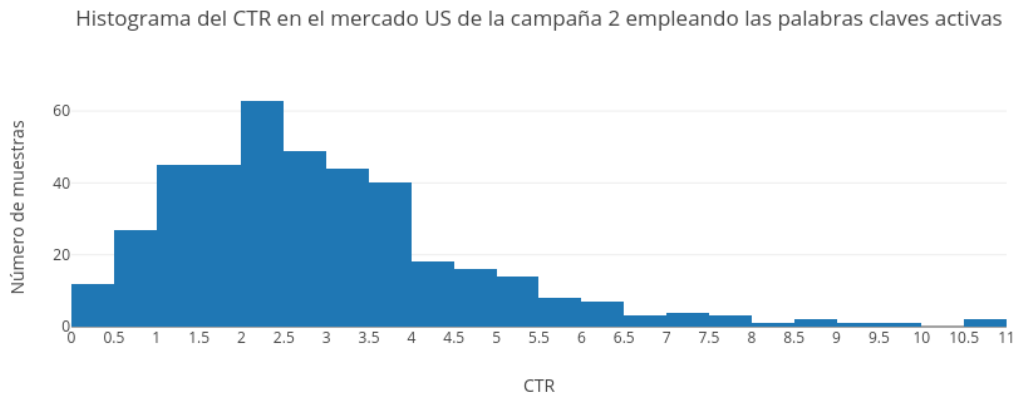

Figura 30: Histograma del CTR en el mercado estadounidense, campaña 2.

Con el objetivo de comparar ambos mercados, hemos realizado un gráfico mostrando ambos histogramas conjuntamente. Hay que tener en cuenta que el mercado británico cuenta con 345 palabras claves activas mientras que el mercado estadounidense tiene 405, por tanto, se aprecia de forma general una ligera superioridad en las barras de este segundo mercado.

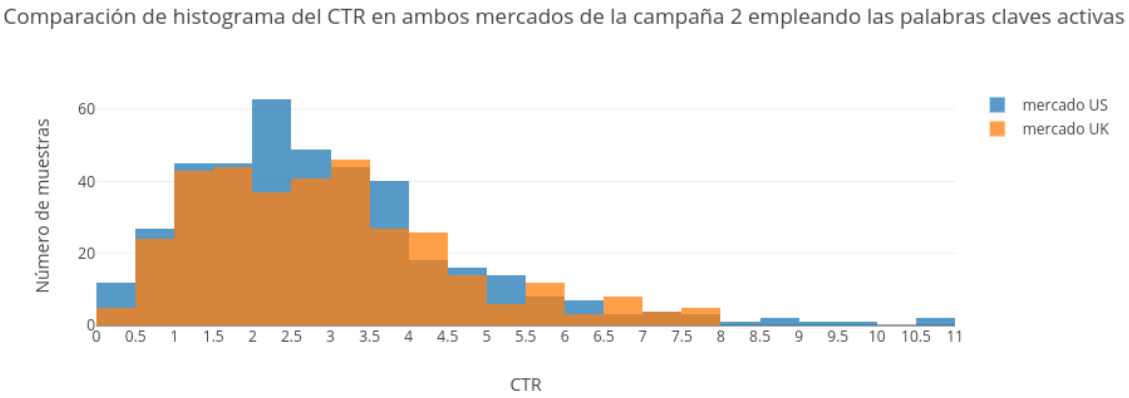

Figura 31: Comparación entre los histogramas del CTR en los mercados de la campaña 2.

Ahora nos interesa visualizar las palabras claves con el mayor CTR del mercado, y comprobar si son muestras con poca importancia en el mercado, es decir, que generan pocas impresiones, como sucedía en el caso anterior. Para ello mostramos las 20 palabras claves con el mayor CTR en la Figura 32.

Las palabras claves incluidas en esta figura, de mayor a menor CTR son: *law lemon ohio, student loan consolidate, google android application development, benchmark lending, attorney injury new personal york, compare life assurance, notebooks and laptops, phones and prices, realtime tracking device, dui florida lawyer, domain yahoo, learning sql server 2008, cost of sim card, mesothilioma, homeowner loans, domain registration yahoo, auto century insurance, line of credit home equity, buy uk sim card*, y *c5650 toner*.

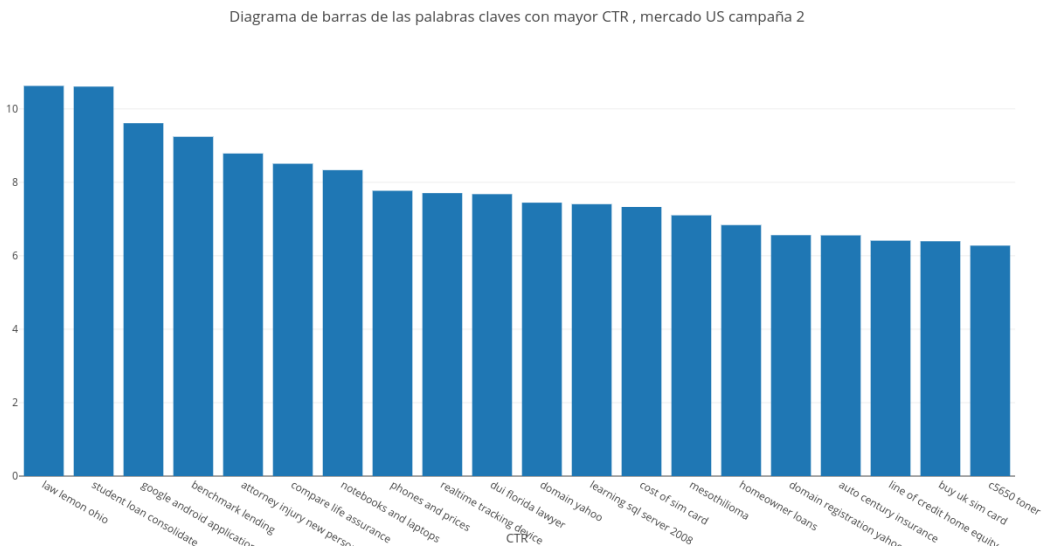

Figura 32: Valor del CTR en las palabras claves con mayor CTR. Mercado estadounidense campaña 2.

Estudiando las impresiones generadas por estas palabras claves detectamos el mismo problema visto en el mercado británico. Son palabras claves que han generado pocas impresiones en todo el mercado estadounidense ya que *line of credit home equity* es la palabra clave con más impresiones generadas en esta lista con 119.398 impresiones generadas (valor muy superior a cualquiera de las demás palabras claves, ya que la siguiente genera poco más de 10.000 impresiones). Si comparamos esos valores con la media de impresiones, que es superior a los 6 millones de impresiones por palabra clave activa, podemos ver que son palabras claves poco significativas dentro de la campaña.

Por tanto, vamos a mostrar el CTR de aquellas palabras claves que han generado un mayor número de impresiones. En la siguiente imagen podemos ver que todas estas palabras claves se encuentran en el rango medio descrito anteriormente, siendo *Netflix* la que tiene CTR, con un valor del 3.84%.

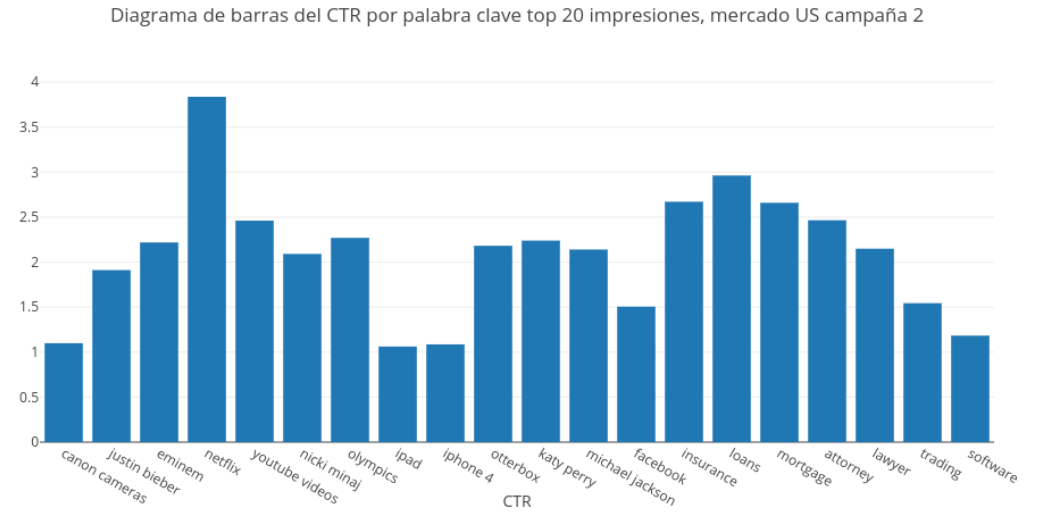

Figura 33: Valor del CTR en las palabras claves que generan más impresiones. Mercado estadounidense, campaña 2.

En este aspecto, ambos mercados también se comportan de forma similar, situándose las palabras clave que generan más impresiones entre los valores medios de la distribución del CTR del mercado, y siendo las palabras claves con mayor CTR de poca importancia.

Este mismo estudio se ha realizado para el mercado conjunto de la campaña 2 dando lugar a resultados análogos, algo lógico debido a la similitud mostrada por ambos mercados.

Campaña 3.

Por otro lado, también se ha realizado este análisis en la campaña 3, obteniendo de nuevo un comportamiento similar al visto hasta ahora:

- Histogramas asimétricos hacia la derecha, mostrando un rango medio para el CTR entre 1.5% y 4%. Como vemos, generalmente las barras del mercado estadounidense son superiores, esto se debe a que este mercado cuenta con 669 palabras claves activas mientras que el mercado británico tiene 544.

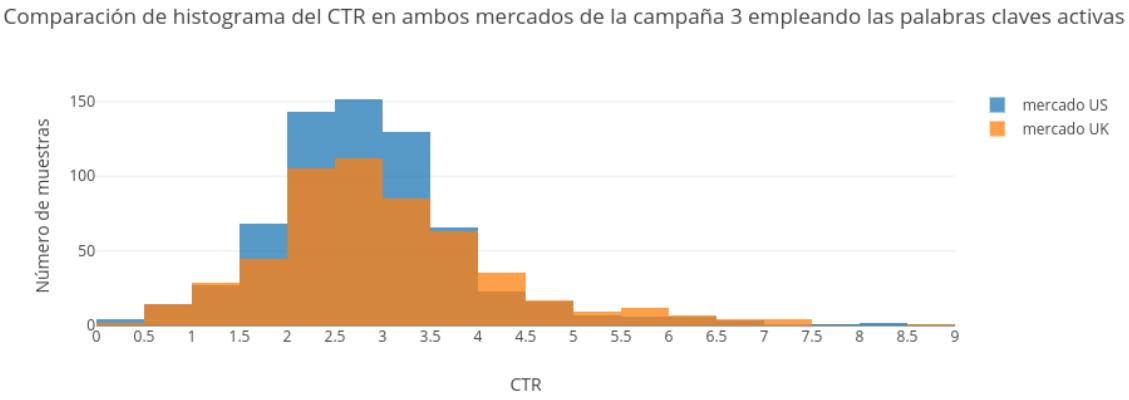

Figura 34: Comparación entre los histogramas del CTR en los mercados de la campaña 3.

- El CTR medio es algo superior que el visto en la campaña 2. Cabe destacar que en esta campaña el CTR medio difiere algo entre ambos mercados, siendo en el mercado británico 2.66% y en el mercado estadounidense del 2.38%.
- Las palabras claves con mayor CTR generan pocas impresiones y aquellas con mayor número de impresiones generadas se encuentran dentro del rango esperado (salvo un par de excepciones).

En la actualidad, el CTR medio es algo superior al visto en estas campañas (años 2012-2013), pero hay que reflejar cómo puede variar este parámetro dependiendo de la industria en la que se enfoque cada uno de los anuncios.

Gracias a un estudio realizado por la agencia de marketing digital 2mas2, podemos mostrar el siguiente gráfico donde se muestra el CTR medio dado en 2018 en las redes de búsqueda y visualización de Google AdWords.

La campaña 2 estudiada tiene palabras claves relacionadas con muchos temas, aunque entre las que más impresiones han generado destacan algunas centradas en la industria legal (*insurance, loans, mortage, etc.*) con un CTR entre el 2.5% y 3%, similar al 2.93% mostrado en este gráfico pese a la diferencia de 6 años que hay entre ambos datos.

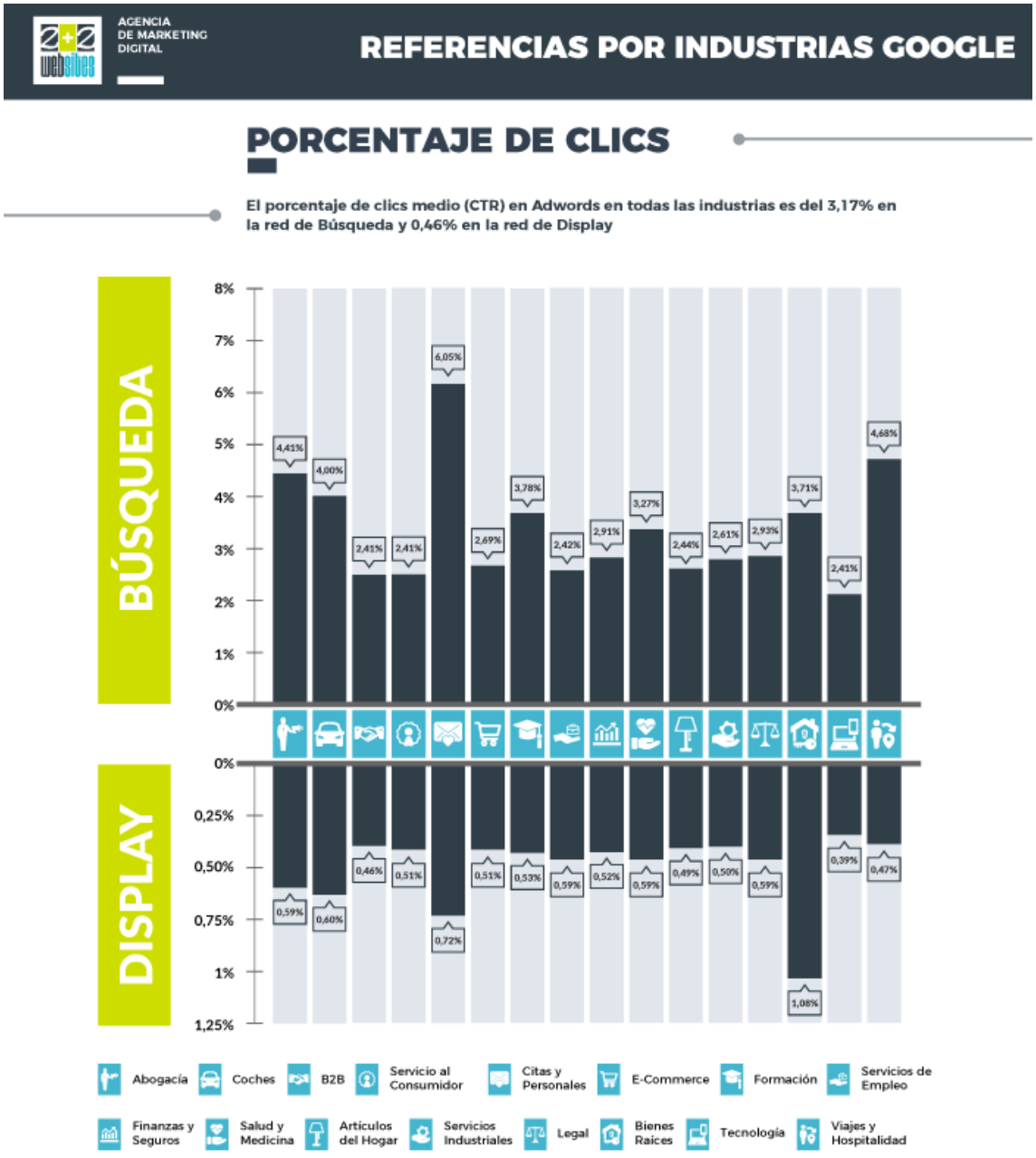

Figura 35: CTR dependiendo de cada industria año 2018 [21].

4.1.5 Estudio del CPC

Como ya hemos estudiado, Google Ads permite diferentes formas de pago, siendo el CPC (*Cost Per Click*), es decir el coste por clic, el más empleado. La forma de obtener el CPC de un anuncio es simple, hay que calcular el cociente entre el coste total invertido en él y el número de clics que ha obtenido.

El estudio realizado sobre el CPC ha seguido un proceso similar al seguido en el análisis del CTR. En primer lugar, nos centramos en mostrar y analizar el histograma de cada mercado individualmente y en su conjunto. Posteriormente visualizamos el valor del CPC para las palabras claves con mayor CPC y finalmente nos centramos en las palabras claves que más impresiones han generado.

Campaña 2, mercado británico.

Figura 36: Histograma del CPC en el mercado británico, campaña 2.

Al hacer el estudio, observamos que hay palabras claves con un CPC atípico, muy elevado respecto a la media calculada (con un valor de 4.78€). Esto se ve reflejado en el siguiente histograma, en el que se muestra el CPC agrupado de 5 en 5, con el fin de mostrar estos valores atípicos.

Apenas se aprecia en la imagen, pero si observamos con detalle podemos ver que hay palabras claves en rangos muy elevados, llegando a aparecer hasta en el rango 200-205, donde se sitúa la palabra clave *debtfreedirect*.

Para determinar los valores atípicos podemos mostrar estos datos en un boxplot, el cual, entre otras cosas, indica los límites a partir de los cuales se consideran atípicos los valores. La forma más común de establecer los límites de los valores atípicos, siendo Q1 y Q3 los cuartiles 1 y 3 respectivamente, es:

- Límite superior = Q3 + 1.5 * (Q3 - Q1)
- Límite inferior = Q1 – 1.5 * (Q3 – Q1)

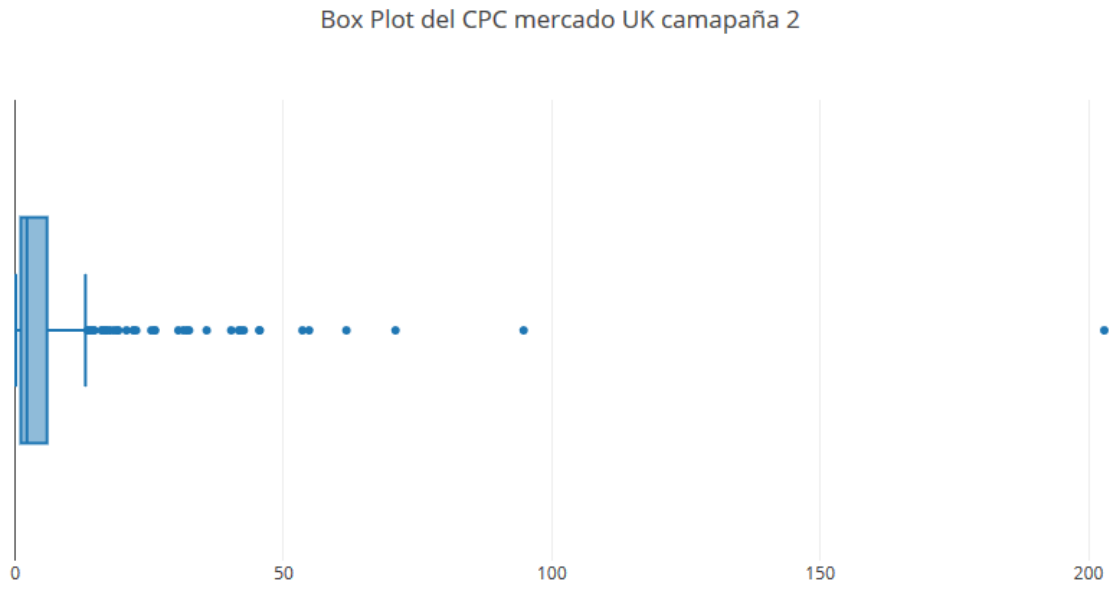

Figura 37: Boxplot CPC mercado británico, campaña 2.

Este gráfico nos muestra:

- Los límites de la caja nos indican los valores Q1 = 1.04 y Q3 = 5.92.
- La línea que hay en medio, dentro de la caja, nos indica el valor de la mediana, en este caso es 2.18.
- Bigotes, son la continuación de la caja hasta llegar a los límites de los valores atípicos, en este caso, solo tenemos valores atípicos en la parte superior de la caja. Realizando el cálculo, obtenemos que este límite se sitúa en 13.08 y, por tanto, todas las muestras con un CPC superior a este valor serán consideradas atípicas.

A continuación, mostraremos de nuevo el histograma del CPC en este mercado, pero excluyendo todas las palabras atípicas cuyo CPC se considere atípico.

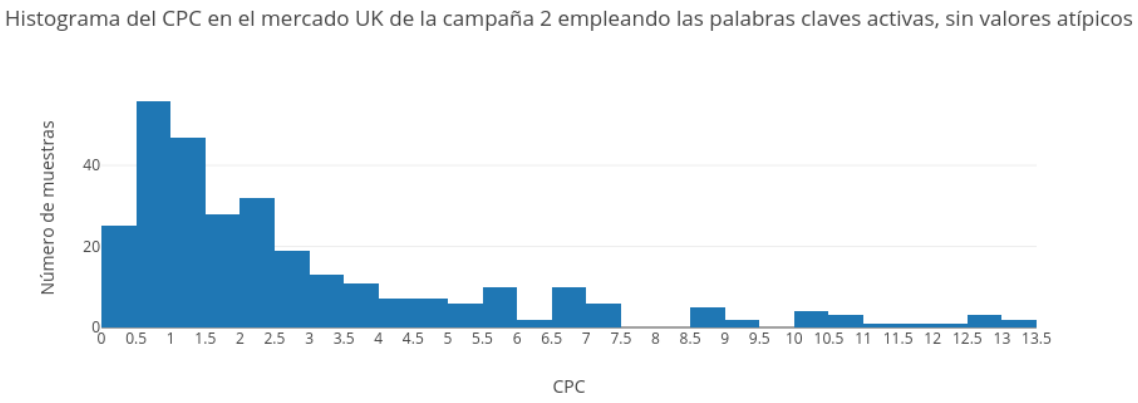

Figura 38: Histograma del CPC sin valores atípicos en el mercado británico, campaña 2.

En esta figura podemos ver que la mayoría de las palabras claves tienen un CPC inferior a la media, cuyo valor es 4.78€. Esto se debe en parte, a los ya vistos valores atípicos, por los cuales el valor medio asciende considerablemente.

A continuación, veremos cuales son las palabras claves con un CPC más elevado y analizaremos el número de impresiones que han generado en este mercado. De este modo podemos ver si son palabras claves activas de forma asidua o casos excepcionales como vimos en el CTR.

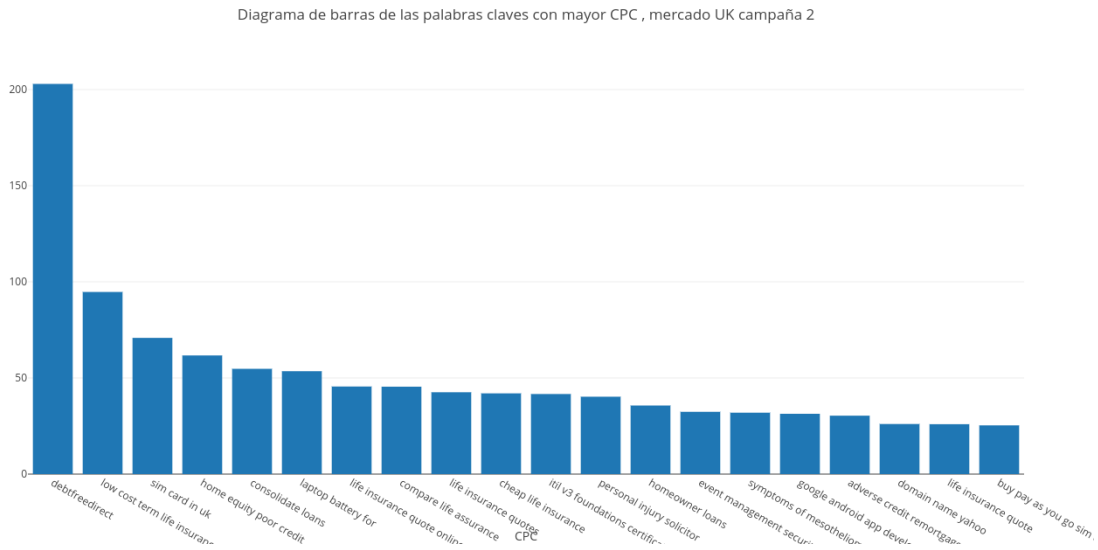

Figura 39: Valor del CPC en las palabras claves con mayor CPC. Mercado británico campaña 2.

Las palabras claves visualizadas en esta gráfica, ordenadas teniendo en cuenta su CPC, de mayor a menor son: *debtfreedirect, low cost term life insurance, sim card in uk, home equity poor credit, consolidate loans, laptop battery for, life insurance quote online, compare life assurance, life insurance quotes, cheap life insurance, itil v3 foundations certification, personal injury solicitor, homeowner loans, event management security, symptoms of mesothelioma, google android app developer, adverse credit remortgage, domain name yahoo, life insurance quote* y *buy pay as you go sim card*.

Analizando el número de impresiones generadas, observamos que las palabras clave que más impresiones ha producido han sido *life insurance quotes* y *life insurance quote*, es decir, el mismo concepto en plural y singular, teniendo en conjunto un total de 748.733 impresiones generadas, un número a tener en cuenta en este mercado. Por otro lado, el resto de estas palabras claves han generado un número de impresiones muy reducido respecto a la media generada por el mercado (13 de estas palabras clave no llegan a las 10.000 impresiones).

Finalmente, al igual que hicimos con el CTR, visualizaremos el CPC de las 20 palabras claves con más impresiones generadas en el mercado, obteniendo la siguiente figura.

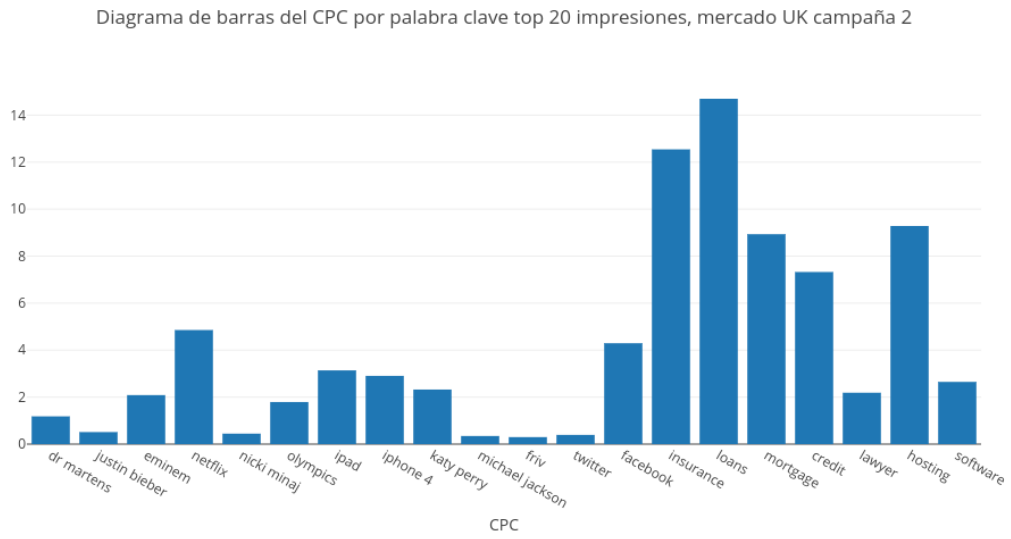

Figura 40: Valor del CPC en las palabras claves que generan más impresiones. Mercado británico, campaña 2.

En esta imagen se puede apreciar cómo las palabras claves relacionadas con el tema legal tienen un CPC mucho mayor que otras palabras claves relacionadas con diversos temas como artistas, dispositivos móviles, televisión, etc. Y esto ocurre al igual que con el CTR, es decir, el CPC puede variar mucho según la industria, el dinero dispuesto a pagar por cada empresa, etc.

Esta variación del CPC según la industria a la que pertenece cada anuncio se puede apreciar en el estudio de 2018 ya citado de la agencia de marketing digital *2mas2*. En ella se aprecia como la industria relacionada con los temas legales es la que mayor CPC tiene con una media de 6.75€

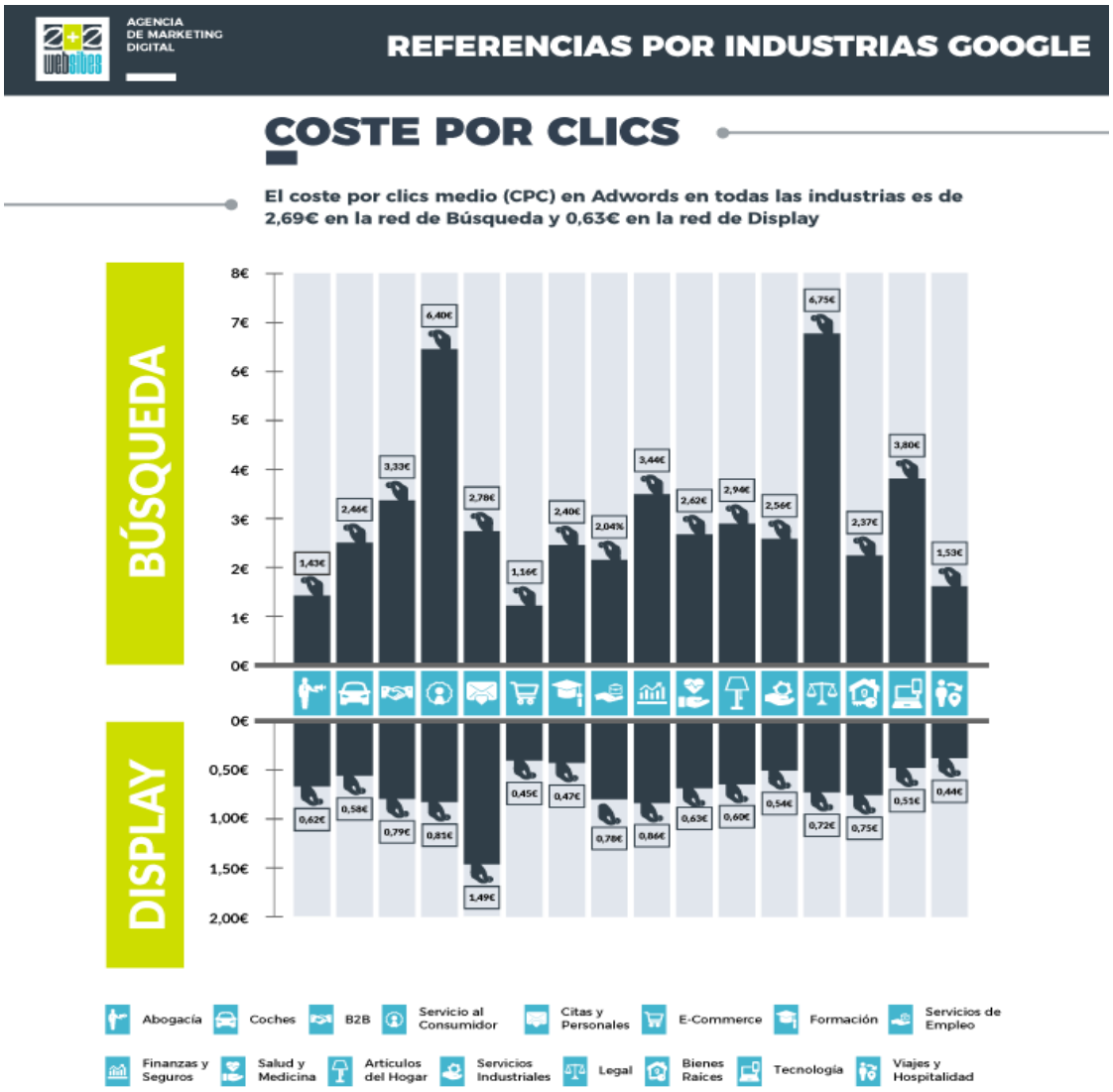

Figura 41: CPC dependiendo de cada industria año 2018 [21].

Comparando el análisis del CTR con el CPC en el mercado británico de la campaña 2, podemos observar que el CPC se comporta de forma algo más caótica, con valores atípicos y una variación muy elevada de su valor dependiendo del sector al que pertenezca cada palabra clave. Para comprobar si este comportamiento es aislado de este mercado, procedemos a realizar el mismo análisis en el mercado estadounidense de la misma campaña.

Campaña 2, mercado estadounidense.

En primer lugar, visualizaremos el histograma del CPC de todas las palabras claves activas en este mercado. Al igual que observamos en el mercado anterior, se aprecian valores atípicos, elevados respecto a la media (4.01€), en este caso estos valores atípicos llegan a aparecer en el rango 280-285 con la palabra clave *personal injury attorney colorado*.

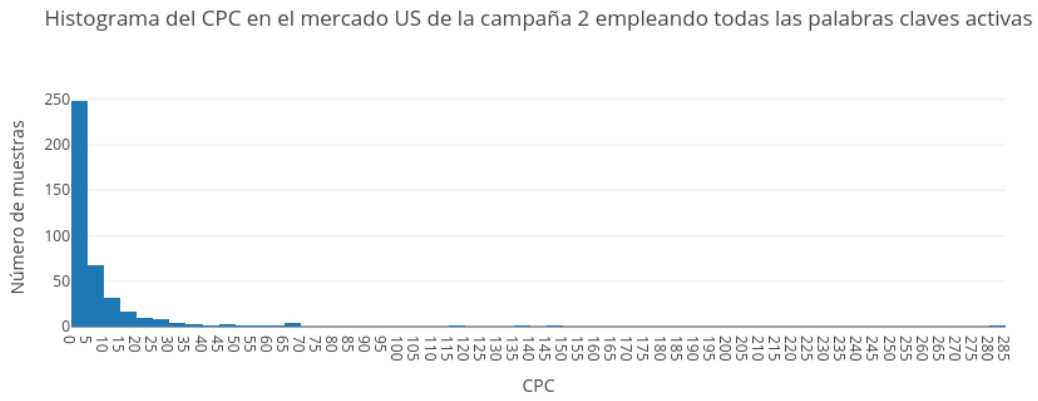

Figura 42: Histograma del CPC en el mercado estadounidense, campaña 2.

Siguiendo el procedimiento estipulado, se calcula a través de un boxplot el límite a partir del que se determinan los valores atípicos.

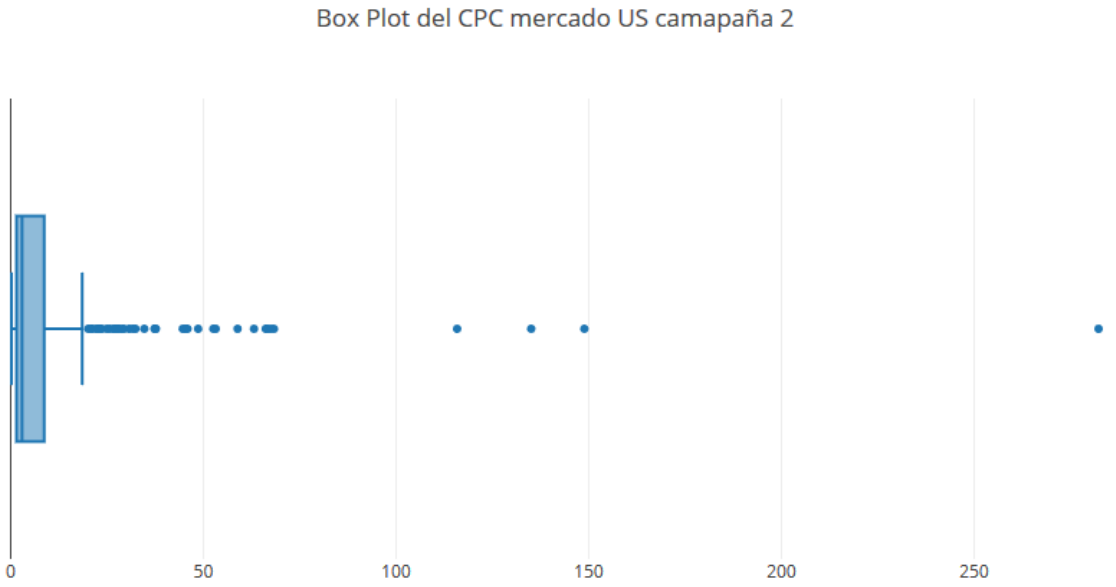

Figura 43: Boxplot CPC mercado estadounidense, campaña 2.

Este boxplot nos muestra los siguientes valores:

- Q1 = 1.53 y Q3 = 8.71. Situados en los límites de la caja.
- La mediana, mostrada con una línea en el interior de la caja y un valor de 2.96.
- El límite superior, representado por un bigote, con valor de 18.57.

Tras realizar estos cálculos, el siguiente paso es mostrar es mismo histograma, pero excluyendo los valores atípicos, es decir aquellas palabras claves cuyo CPC es superior a 18.57€.

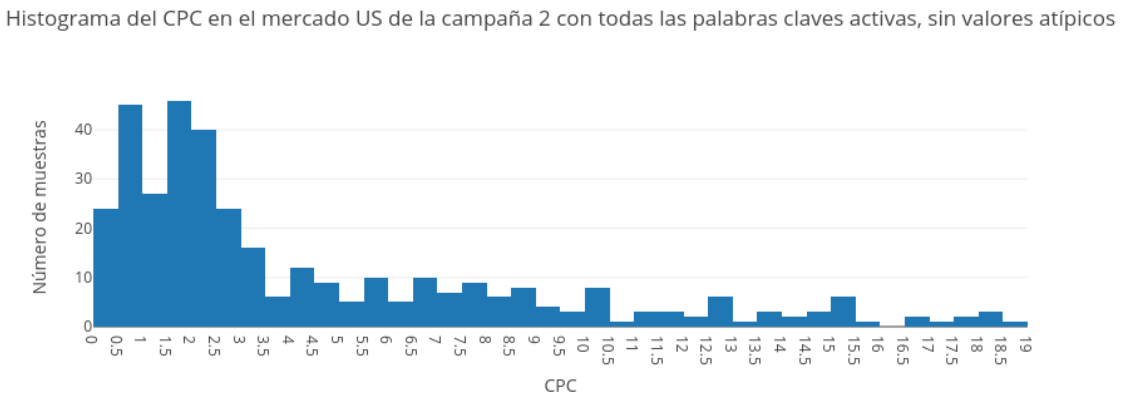

Figura 44: Histograma del CPC sin valores atípicos en el mercado estadounidense, campaña 2.

Como se aprecia en la imagen, y al igual que ocurría con el mercado británico observamos que la mayoría de las palabras claves están situadas en rangos inferiores al CPC medio de 4.01€.

En cuanto a las palabras con mayor CPC, se aprecia un comportamiento esperado, es decir, estas palabras claves generan un número de impresiones mucho menor a la media, siendo esta superior a 6 millones por palabra clave.

Por otro lado, las palabras clave que generan mayor cantidad de impresiones, también se comportan de forma similar al mercado británico, es decir, las palabras clave relacionadas con temas legales tienen un CPC superior a otras relacionadas con diversos temas.

Campaña 3.

Al realizar el mismo estudio en esta campaña observamos resultados muy similares a los obtenidos en la campaña 2. Hay que destacar que en esta campaña el CPC medio es superior, ascendiendo a 11.8€ y 16.12€ en los mercados británicos y estadounidense respectivamente. Aun así el comportamiento de esta campaña es similar al visto hasta ahora:

- Valores atípicos, muy elevados respecto a la media del mercado.
- Muchas palabras claves situadas por debajo de la media del CPC.
- Las palabras claves con mayor CPC generan pocas impresiones respecto al resto de palabras claves.
- Variación del CPC dependiendo de la industria a la que pertenezca cada anuncio.

4.1.6 Conclusión.

Todo lo visto hasta ahora nos muestra un comportamiento similar entre los mercados de una misma campaña y entre ambas campañas.

En los siguientes gráficos se muestran las comparaciones entre ambas campañas teniendo en cuanto el CTR y el CPC. Antes de analizar estos dos gráficos, hay que tener en cuenta:

- En la campaña 2 hay un total de 450 palabras activas mientras que en la campaña 3 hay 685. Esto se aprecia en todos los gráficos, pero sobre todo en la comparación del CTR, ya que el rango de ambos mercados es muy similar.
- Respecto al CPC, la campaña 3 tiene una media (15.07) bastante superior a la campaña 2 (4.15). Esto implica un límite de valores atípicos muy superiores, lo que provoca que haya más palabras claves en rangos superiores en la campaña 2. El límite en la campaña 2 es 21.31€ mientras que en la campaña 3 es 60.86€.

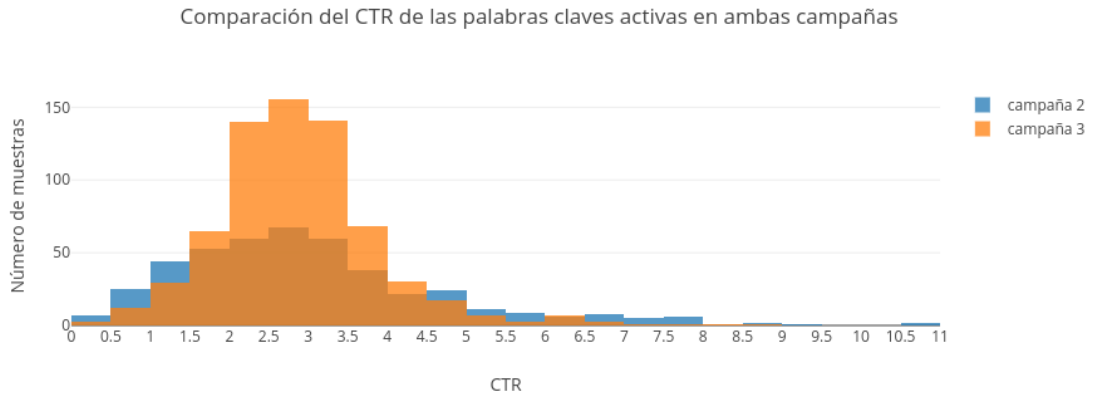

Figura 45: Comparación del CTR entre ambas campañas.

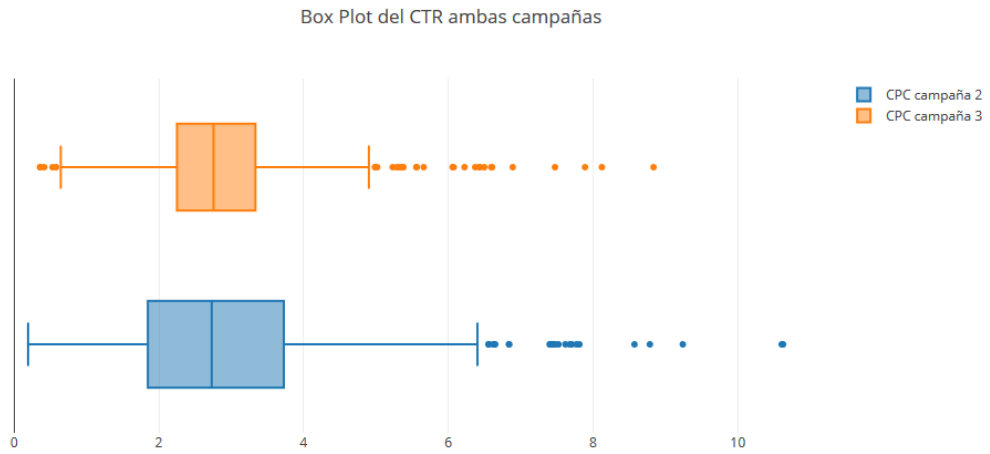

Figura 46: Boxplot CTR ambas campañas.

Figura 47: Comparación del CTR entre ambas campañas

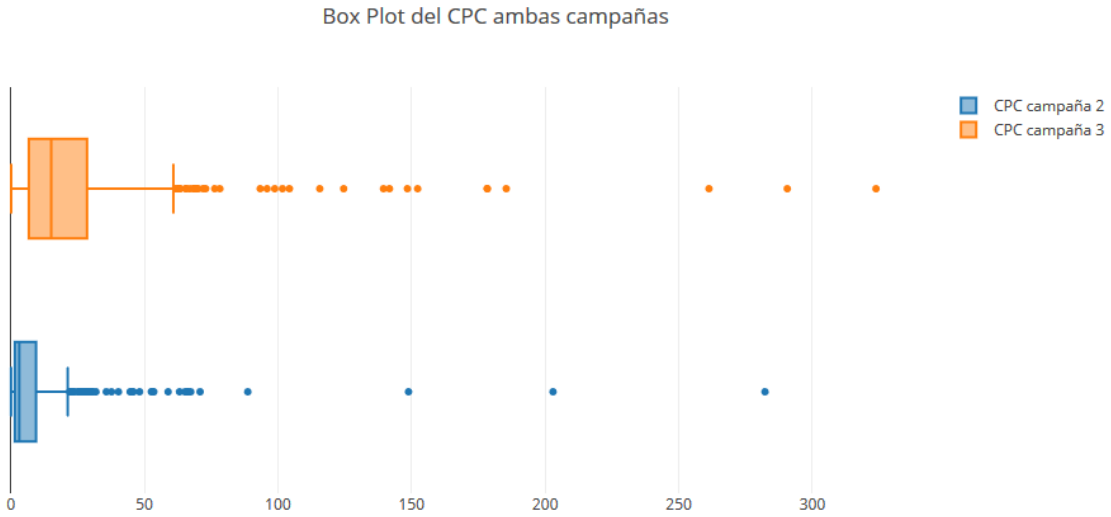

Figura 48: Boxplot CTR ambas campañas.

4.2 Criteo.

El estudio realizado sobre este dataset es muy diferente al hecho en Google Ads, ya que está totalmente enfocado a la selección de las características más importantes del conjunto de datos para emplearlas posteriormente en la generación de un modelo de aprendizaje.

En primer lugar, vamos a recordar la información que contiene este conjunto de datos. Este dataset está compuesto por anuncios recopilados por Criteo durante varios días, ordenados cronológicamente. Cada anuncio equivale a una fila, y disponemos de un total de 45.840.617 anuncios. Por otro lado, tenemos 40 columnas, la primera de ellas es la etiqueta *Label* que indica si se hace clic (valor 1) o no (valor 0) en el anuncio, el resto

de las columnas serán las que tengan las características de los anuncios, y se distribuyen en 13 columnas numéricas y 26 categóricas.

Este dataset está totalmente anonimizado, es decir, no sabemos lo que significa cada columna, ni los datos categóricos que estas contienen. Por este motivo denominaremos a las columnas categóricas como C1-C26 y a las numéricas por I1-I13.

A continuación, vamos a analizar cada una de las columnas individualmente, con el fin de determinar si cada una de ellas puede ser adecuada para la generación del modelo. Debido a la capacidad computacional disponible para la realización de este proyecto, ya que se ha empleado únicamente un ordenador personal por acotar la carga de trabajo del TFG, a partir de este punto se van a trabajar únicamente con los 10 primeros millones de anuncios.

Para ello, en primer lugar vamos a mostrar una tabla comparativa entre el dataset completo y el dataset resumido de 10 millones de registros mencionado. De este modo, observamos la semejanza entre ambos dataset y por ello la decisión de trabajar con una muestra y no con el dataset completo. La primera columna indica la columna estudiada, en la segunda y tercera columnas mostramos el porcentaje de valores vacíos que contiene cada uno de los dataset. Finalmente, las dos últimas columnas muestran la mediana en las columnas numéricas, el número de etiquetas diferentes en las columnas categóricas y el porcentaje de éxito en la columna *Label*.

En esta tabla también se va a observar el resultado de un primer filtrado, eliminaremos de nuestro modelo de aprendizaje aquellas columnas que cumplan alguno de estos dos criterios:

1. Alto porcentaje de valores vacíos. Aquellas columnas con más de un 25% de valores vacíos es eliminada. (Reflejado en la tabla en rojo)
2. Alta granularidad en las columnas categóricas. Aquellas columnas con más de 10.000 etiquetas diferentes. (Reflejado en la tabla en azul)

Columna	% valores vacíos dataset completo.	% valores vacíos muestra 10 millones.	Mediana / Etiquetas diferentes. Dataset completo	Mediana / Etiquetas diferentes. Muestra 10 millones.
Label	0%	0%	25,6%	25,1%
C1	0%	0%	1460	1396
C2	0%	0%	583	553
C3	3,4%	3,5%	10.131.227	2.594.031
C4	3,4%	3,5%	2.202.608	345.039
C5	0%	0%	305	290
C6	12,1%	12,8%	24	23
C7	0%	0%	12.517	12.048
C8	0%	0%	633	608
C9	0%	0%	3	3
C10	0%	0%	93.145	65.156
C11	0%	0%	5.683	5.309
C12	3,5%	3,5%	8.351.593	2.186.509

C13	0%	0%	3.194	3.128
C14	0%	0%	27	26
C15	0%	0%	14.992	12.750
C16	0%	0%	5.461.306	1.537.323
C17	0%	0%	10	10
C18	0%	0%	5.652	5.002
C19	44%	43,9%	2.173	2.118
C20	44%	43.9%	4	4
C21	3.4%	3,5%	7.046.547	1.902.327
C22	76,3%	76%	18	17
C23	0%	0%	15	15
C24	0%	0%	286.181	135.789
C25	44%	43,9%	105	94
C26	44%	43,9%	142.572	84.305
I1	45,4%	44,3%	1	1
I2	0%	0%	3	3
I3	21,5%	21,5%	6	6
I4	21,7%	22%	4	4
I5	2,6%	2,7%	2.813	2.763
I6	22,4%	22,8%	32	33
I7	4,3%	4,4%	3	3
I8	0%	0%	7	7
I9	4,3%	4,4%	38	38
I10	45,4%	44,3%	1	1
I11	4,3%	4,4%	1	1
I12	76,5%	77%	0	0
I13	21,7%	22%	4	4

Tabla 5: comparación entre dataset completo y muestra de 10 millones de registros.

Por tanto, tras este primer filtrado el dataset mantiene las siguientes columnas:

- Columna *Label*.
- Columnas numéricas: I2, I3, I4, I5, I6. I7, I8, I9, I11, I13. Un total de 10 columnas, ya que 3 han sido eliminadas.
- Columnas categóricas: C1, C2, C5, C6, C8, C9, C11, C13, C14, C17, C18, C23. Un total de 12 columnas, con 14 eliminadas.

Columna *Label*.

Esta columna es la etiqueta de nuestro conjunto de datos, es decir, el valor que debemos predecir posteriormente. Como hemos visto anteriormente es una columna booleana, es decir, tiene dos valores, el 0 para indicar que el anuncio no ha sido clicado y el 1 para indicar que se ha producido clic en él.

La columna *Label* está completa, no tiene ningún valor vacío. Si observamos su contenido vemos que el 25,6% de los anuncios han tenido éxito siendo clicados.

Columnas numéricas.

En estas columnas no se ha realizado ningún filtrado más, por tanto todas las columnas mencionadas anteriormente serán usadas para generar nuestro modelo. Sin embargo, sí que se ha realizado un procesado ya que se puede observar que todas las columnas tienen valores extremos muy elevados respecto a la mediana del conjunto.

Por ejemplo, si analizamos la columnas I2 podemos ver que su mediana es 3 y que el 51.3% de los registros tienen un valor entre -1 y 3, pero si observamos los extremos nos encontramos con registros cuyos valores son 22.066, 20.881, 20.712, etc. todos ellos únicos.

Por tanto, pese a desconocer la información contenida en estas columnas se han modificado todos los valores que superen los límites superiores e inferiores, es decir, los valores atípicos, y se ha asignado a estos registros la mediana de cada una de las columnas. Por otro lado, también se han rellenado los registros sin valor con la mediana.

Este proceso también se ha llevado a cabo en los datos de test, pero con los valores calculados en los datos de entrenamiento, para evitar emplear información de los datos de test en la generación del modelo. Es decir, hemos usado la mediana calculada de cada una de las columnas del dataset de entrenamiento, y empleando este valor se ha rellenado la columna correspondiente en el dataset de test.

Columnas categóricas.

En el caso de las columnas categóricas, se ha realizado un análisis columna a columna para determinar si son características interesantes para incluir en nuestro modelo de aprendizaje. Con este objetivo, se ha analizado la diversidad de etiquetas que tendrá cada columna, mostrando las 10 etiquetas más repetidas, y eliminando de este modo aquellas columnas que no posean etiquetas con un porcentaje de repetición alto.

- Columna C1: pese a tener un total de 1.396 etiquetas diferentes, en esta columna el 75,1% de los registros se reparten entre 3 etiquetas diferentes, y una de estas tres etiquetas supera el 50% del total. Por tanto, aunque no sabemos la información que contiene, consideramos que esta columna será una característica importante en nuestro modelo ya que al menos se podrá diferenciar entre los registros que tienen el valor mencionado y todos los demás.

Value	Count	Frequency (%)
05db9164	5007541	50.1%
68fd1e64	1667889	16.7%
5a9ed9b0	834720	8.3%
8cf07265	494604	4.9%
be589b51	329556	3.3%
5bfa8ab5	239929	2.4%
87552397	178444	1.8%
f473b8dc	141005	1.4%
39af2607	109225	1.1%
ae82ea21	89101	0.9%
Other values (1386)	907986	9.1%

Figura 49: Etiquetas más comunes columna C1.

- Columna C2: esta columna posee 553 etiquetas diferentes, menos de la mitad que la columna C1, sin embargo no se va a emplear en el modelo ya que posee una distribución muy general, es decir, no hay etiquetas que se repitan considerablemente.

Value	Count	Frequency (%)
38a947a1	1146635	11.5%
207b2d81	463771	4.6%
1cfdf714	420596	4.2%
38d50e09	404182	4.0%
287130e0	355155	3.6%
4f25e98b	329987	3.3%
09e68b86	293699	2.9%
421b43cd	282877	2.8%
58e67aaf	242651	2.4%
80e26c9b	214092	2.1%
Other values (543)	5846355	58.5%

Figura 50: Etiquetas más comunes columna C2.

- Columna C5: esta columna cuenta con 290 etiquetas diferentes. Pese a ello, el 67,1% de los registros tienen un mismo valor, *25c83c98*, por tanto, esta característica se emplea en el modelo.

Value	Count	Frequency (%)
25c83c98	6713418	67.1%
4cf72387	1565676	15.7%
43b19349	632557	6.3%
384874ce	326791	3.3%
30903e74	193178	1.9%
0942e0a7	125431	1.3%
f281d2a7	85369	0.9%
b0530c50	59650	0.6%
b2241560	46246	0.5%
f3474129	36014	0.4%
Other values (280)	215670	2.2%

Figura 51: Etiquetas más comunes columna C5.

- Columna C6: esta columna tan sólo posee 23 valores diferentes, por tanto, es una opción a tener en cuenta en nuestro modelo. Además, podemos observar que en tan sólo tres valores diferentes se encuentran el 79% (91,8% tras rellenar los valores vacíos) de los registros. También hay que tener en cuenta que esta columna posee el 12,8% de sus registros sin datos, y con el procedimiento descrito hasta ahora, estos datos se rellenan con la moda (tanto datos de entrenamiento como test, pero teniendo en cuenta sólo los datos de entrenamiento para calcular dicha moda), por tanto, la etiqueta más repetida, en este caso *7e0ccccf* pasará de tener un porcentaje del 39,6% al 52,4% tras el procesado. Podemos ver en la siguiente imagen etiquetas que aparecen con un 0% de frecuencia, esto se debe al redondeo (sus porcentajes reales son *e3520422* con 0,017%, *c05778d5* con 0,01%, *c76aecf6* con 0,0048%, *f1f2de2d* con 0,0035% y el resto de los valores, los 12 con menor porcentaje, sumando un total del 0,0029%).

Value	Count	Frequency (%)
7e0ccccf	3958168	39.6%
fbad5c96	2174082	21.7%
fe6b92e5	1799773	18.0%
13718bbd	308381	3.1%
6f6d9be8	281029	2.8%
3bf701e7	194634	1.9%
e3520422	1655	0.0%
c05778d5	1006	0.0%
c76aecf6	484	0.0%
f1f2de2d	347	0.0%
Other values (12)	292	0.0%
(Missing)	1280149	12.8%

Figura 52: Etiquetas más comunes columna C6.

- Columna C8: en esta columna con 608 etiquetas diferentes, nos encontramos una distribución similar a la columna C1, es decir, tenemos una columna con un porcentaje muy elevado, del 59,4%, y entre los 3 valores con mayor frecuencia tenemos un total del 83,5%. Esta columna será empleada en el modelo.

Value	Count	Frequency (%)
0b153874	5942627	59.4%
5b392875	1662064	16.6%
1f89b562	748602	7.5%
37e4aa92	415466	4.2%
062b5529	260135	2.6%
51d76abe	178378	1.8%
c8ddd494	125825	1.3%
64523cfa	95038	1.0%
6c41e35e	72709	0.7%
985e3fcb	59505	0.6%
Other values (598)	439651	4.4%

Figura 53: Etiquetas más comunes columna C8.

- Columna C9: esta columna posee sólo 3 valores diferentes, acumulándose el 89,6% de los registros en uno de ellos. Al igual que pasaba anteriormente aparece la etiqueta *a18233ea* con una frecuencia de 0% debido al redondeo, siendo su valor real 0,016%. Esta columna será usada en nuestro modelo.

Value	Count	Frequency (%)
a73ee510	8959572	89.6%
7cc72ec2	1038820	10.4%
a18233ea	1608	0.0%

Figura 54: Etiquetas más comunes columna C9.

- Columna 11: columna con 5.309 etiquetas diferentes, siendo la etiqueta con mayor frecuencia *755e4a50* con tan sólo un 2,3%. Debido a esta distribución y como vimos con C2 esta columna no será empleada en el modelo.

Value	Count	Frequency (%)
755e4a50	227886	2.3%
e51ddf94	227670	2.3%
7f8ffe57	179861	1.8%
4d8549da	155551	1.6%
a7b606c4	98464	1.0%
b7094596	92144	0.9%
8b94178b	91996	0.9%
1054ae5c	87356	0.9%
7e40f08a	86026	0.9%
5874c9c9	77902	0.8%
Other values (5299)	8675144	86.8%

Figura 55: Etiquetas más comunes columna C11.

- Columna 13: esta columna de 3.128 valores diferentes tiene una distribución similar a la anterior, la etiqueta con mayor frecuencia tiene el 2.5% del total. Por tanto, tampoco emplearemos esta columna en el modelo.

Value	Count	Frequency (%)
3516f6e6	254987	2.5%
5978055e	227886	2.3%
46f42a63	194435	1.9%
51b97b8f	155551	1.6%
1aa94af3	131567	1.3%
740c210d	124829	1.2%
025225f2	121278	1.2%
eae197fd	98765	1.0%
6e5da64f	98175	1.0%
80467802	98074	1.0%
Other values (3118)	8494453	84.9%

Figura 56: Etiquetas más comunes columna C13.

- Columna 14: columna con 26 etiquetas diferentes, agrupadas en 3 de ellas el 84,4% de los registros. Esta columna se va a emplear para generar nuestro modelo.

Value	Count	Frequency (%)
b28479f6	3565773	35.7%
07d13a8f	3360098	33.6%
1adce6ef	1513820	15.1%
64c94865	447922	4.5%
cfef1c29	318612	3.2%
051219e6	239838	2.4%
8ceecbc8	138734	1.4%
f862f261	118710	1.2%
d2dfe871	67744	0.7%
32813e21	66088	0.7%
Other values (16)	162661	1.6%

Figura 57: Etiquetas más comunes columna C14.

- Columna 17: columna que emplearemos en nuestro modelo ya que posee tan sólo 10 etiquetas diferentes. La etiqueta *af5d780c* que aparece en la siguiente imagen con un 0% de frecuencia debido al redondeo, tiene un valor real del 0.0034%.

Value	Count	Frequency (%)
e5ba7672	4586413	45.9%
07c540c4	1325962	13.3%
d4bb7bd8	1171945	11.7%
3486227d	819770	8.2%
776ce399	518427	5.2%
27c07bd6	445672	4.5%
1e88c74f	438857	4.4%
8efede7f	420119	4.2%
2005abd1	272501	2.7%
af5d780c	334	0.0%

Figura 58: Etiquetas más comunes columna C17.

- Columna 18: similar distribución a las columnas C11 y C13, por tanto no se va a emplear en el modelo. Esta columna posee un total de 5002 etiquetas siendo la que mayor frecuencia posee *e88ffc9d* con un 3,7%.

Value	Count	Frequency (%)
e88ffc9d	365460	3.7%
2804effd	282876	2.8%
891589e7	282475	2.8%
c21c3e4c	242651	2.4%
7ef5affa	201051	2.0%
5aed7436	200836	2.0%
395856b0	193490	1.9%
582152eb	170507	1.7%
5bb2ec8e	149895	1.5%
6fc84bfb	108277	1.1%
Other values (4992)	7802482	78.0%

Figura 59: Etiquetas más comunes columna C18.

- Columna 23: columna con 15 valores diferentes, será empleada en el modelo. En esta columna el 75,9% de los registros tienen un valor repartido entre 3 etiquetas diferentes.

Value	Count	Frequency (%)
32c7478e	4169943	41.7%
3a171ecb	2111565	21.1%
423fab69	1306668	13.1%
bcdee96c	708214	7.1%
be7c41b4	556754	5.6%
c7dc6720	543058	5.4%
55dd3565	239001	2.4%
dbb486d7	169153	1.7%
93bad2c0	85233	0.9%
c3dc6cef	49107	0.5%
Other values (5)	61304	0.6%

Figura 60: Etiquetas más comunes columna C23.

Finalmente, tras realizar este segundo filtrado, las columnas a emplear en nuestro modelo serán:

- Columna *Label*.
- Columnas numéricas: I2, I3, I4, I5, I6. I7, I8, I9, I11, I13.
- Columnas categóricas: C1, C5, C6, C8, C9, C14, C17, C23.

5. Modelos de aprendizaje.

En este capítulo vamos a explicar los modelos de aprendizaje empleados en este proyecto, así como los procesos realizados para poder aplicarlos. A partir de este punto sólo trabajaremos con el dataset de Criteo.

Hay que recordar que disponemos de dos dataset Criteo, uno para entrenamiento, del que tomamos los primeros 10 millones de registros, y otro de test, empleado para obtener nuestra evaluación del modelo en *Kaggle*.

Hasta ahora, hemos rellenado los valores no disponibles tanto en las columnas numéricas como en las categóricas con la moda de cada columna, es decir, con el valor más repetido. Por otro lado, también se han sustituido los valores atípicos de las columnas numéricas por la mediana. En este punto, tenemos el dataset de entrenamiento y test completo, todas las columnas tienen el 100% de los datos, empleando únicamente información del dataset de entrenamiento en este proceso.

A la hora de generar modelos de aprendizaje, empleamos la biblioteca *scikit-learn* [10], basada en el aprendizaje automático en Python. Esta biblioteca no sólo nos permitirá generar modelos de aprendizaje, ya que también proporciona herramientas para comparar y seleccionar el mejor modelo posible, preparar los datos antes de entrenar el modelo, etc.

El siguiente paso es preparar los datos categóricos para ser empleados en los modelos, ya que estos modelos son aplicaciones matemáticas, es decir, trabajan con números. Por tanto debemos modificar estas columnas y sustituirlas por números. Para ello, *scikit-learn* nos proporciona dos opciones:

- *LabelEncoder*: codifica las etiquetas entre 0 y N – 1, siendo N el número de etiquetas diferentes que hay en total. Cada vez que se repite una etiqueta se codifica con el mismo valor.
- *OneHotEncoder*: divide cada columna en nuevas columnas, una nueva columna por cada etiqueta diferente (se puede suprimir una de las nuevas columnas). Tras ello, cada columna tendrá un valor 0 para los registros que no coincidan con su etiqueta y un valor 1 cuando sí coincidan.

Podemos entenderlo mejor con un ejemplo muy simple, empleando la columna C17 del dataset Criteo. En la izquierda tenemos la columna con las etiquetas originales, a continuación, se muestra la codificación tras aplicar *LabelEncoder* y finalmente las columnas generadas con *OneHotEncoder*:

		C17_0	C17_1	C17_2	C17_3	C17_4	C17_5	C17_6	C17_7	C17_8	C17_9
e5ba7672	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
07c540c4	0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8efede7f	6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
1e88c74f	1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1e88c74f	1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figura 61: Ejemplo de Label Encoder vs OneHotEncoder.

En un primer momento se decide emplear *OneHotEncoder*, ya que *LabelEncoder* presenta un problema, cuando introducimos la columna generada de este modo en nuestro modelo estamos indicando que la etiqueta e5ba7672 (con valor 9) es nueve veces superior a la etiqueta 1e88c74f (con valor 1). Esto es un problema ya que desconocemos su significado.

Por otro lado, al aplicar *OneHotEncoder* se produce un error debido a la falta de memoria disponible. Por este motivo, se propusieron dos nuevas alternativas, que se llevaran a cabo y se comparan sus resultados:

- Emplear menos columnas categóricas, haciendo el modelo más simple. Para ello se fue reduciendo el número de columnas a emplear hasta poder trabajar sin errores. Con esta alternativa, sólo se incluirá en el modelo las columnas categóricas C9 (con 3 etiquetas diferentes) y C17 (con 10 etiquetas diferentes).
- Emplear *LabelEncoder* con las columnas categóricas disponibles. Como hemos visto esta opción es peor que *OneHotEncoder* pero nos permitirá emplear todas las columnas categóricas que hemos seleccionado en nuestro modelo de aprendizaje.

A continuación hay que seleccionar los parámetros a emplear en cada modelo, una de las opciones que nos proporciona *scikit-learn* es emplear la clase *GridSearchCV*. Esta clase nos permite comparar los resultados de un modelo al modificar sus parámetros, empleando validación cruzada. Esta clase se va a llevar a cabo junto a una normalización de los datos, que facilita una rápida convergencia y mejora la estabilidad de alguno de los procesos matemáticos que llevarán a cabo nuestros modelos.

Este escalado se lleva a cabo con la clase *StandardEscaler* de *scikit-learn*, que se encarga de centrar y escalar todos los valores de cada una de las columnas. El centrado se produce restando a cada valor de una columna la media de esta. Por otro lado, el escalado se realiza dividiendo el resultado obtenido en el centrado por la desviación estándar de la columna.

El proceso a realizar es el siguiente:

1. Dividir la muestra en datos de entrenamiento y validación para realizar la validación cruzada.
2. Normalizar los datos de entrenamiento y validación, de forma individual. Teniendo en cuenta para ello únicamente los datos de entrenamiento, es decir, normalizamos ambos dataset con los valores calculados a partir de los datos de entrenamiento.
3. Generar el modelo de aprendizaje con los parámetros determinados.

Esto se puede hacer empleando un *pipeline* en la clase *GridSearchCV* y para mostrarlo vamos a observar sus principales parámetros:

- *estimator:* implementa la interfaz del estimador. En este parámetro se incluirá una *pipeline*, es decir una tubería, en la cual se van a realizar dos pasos, en primer lugar la normalización, y en segundo lugar la estimación empleando modelo y parámetros a usar.
- *param_grid*: diccionario o lista de diccionarios con los parámetros a emplear en cada una de las iteraciones de esta clase, se realizarán todas las combinaciones posibles con los parámetros incluidos. Este parámetro dependerá exclusivamente del modelo que se emplee.
- *scoring*: método empleado en la medición del error. En nuestro caso empleamos *neg_log_loss*, ya que es el método indicado por *Kaggle* para esta competición.
- *cv*: indica la estrategia a emplear en la validación cruzada. Al introducir un número entero, especifica el número de iteraciones a realizar para cada una de las posibles combinaciones con los parámetros introducidos.

En una primera toma de contacto con los modelos de aprendizaje se emplearon varias opciones:

- *Logistic Regression*: modelo simple basado en la función logística.
- *K-Nearest-Neighbor*: clasifica valores por cercanía, es decir, con los datos más similares. El tiempo de convergencia de este modelo era demasiado alto para el estudio que se quería realizar en este proyecto, siendo una de las causas por las que no se ha llevado a cabo.
- *Gaussian Process Classifier*: basado en la aproximación de Laplace.
- *Gradient Boosting Classifier*: emplea un conjunto de modelos de predicción débiles para generar su predicción definitiva.
- *DecisionTreeClassifier*: modelo que clasifica sus muestras a través de la ramificación a partir de un nodo raíz.

Una vez vistos estos modelos de forma muy general, se ha querido profundizar en dos de ellos, la regresión logística por su sencillez y los árboles de decisión.

5.1 Regresión logística.

La regresión logística (*Logistic Regression*) es uno de los algoritmos más empleados en el aprendizaje automático, especialmente en problemas de clasificación binaria, como al que nos enfrentamos. Esto se debe a que es un algoritmo simple, de fácil interpretación, que funciona bien en muchas aplicaciones [22].

Pese a su nombre, la regresión logística no es un algoritmo empleado en problemas de regresión, en los que se quiere obtener un valor continuo, sino en problemas de clasificación que nos permiten determinar entre dos o más valores o etiquetas.

Para entender la regresión logística debemos observar, en primer lugar, la función logística, la cual se muestra en la Figura 62.

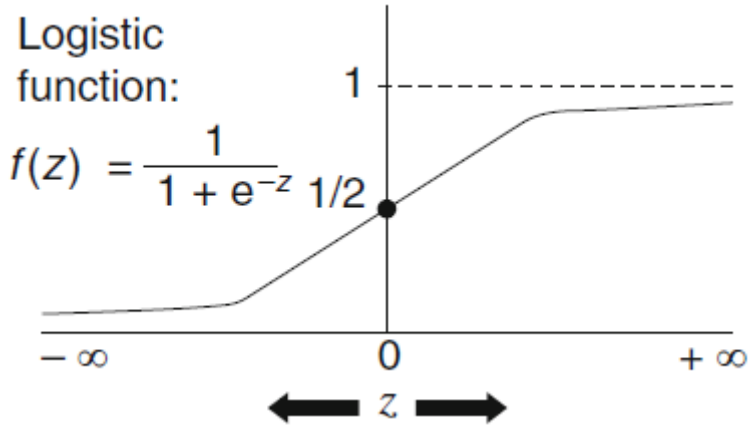

Figura 62: Función logística [22].

El resultado de esta función logística es una probabilidad, entre 0 y 1. La clasificación binaria se obtiene de una forma muy sencilla, a través de un umbral. Cuando la probabilidad obtenida es superior al umbral fijado se asigna un 1, mientras que si es inferior a este umbral será un 0. Sin embargo, debido a que nuestra forma de medir el error en este proyecto es emplear la pérdida logarítmica, que funciona con probabilidades, no será necesario realizar este paso.

Ahora es el turno de emplear esta función logística en nuestro modelo de aprendizaje. Como podemos ver en la Figura 63: Modelo regresión logística [22]. Para obtener el modelo logístico de la función logística, se escribe z como la suma lineal de β_i * X_i y α siendo i el número de variables independientes de interés, X_i estas variables de interés y β_i y α son parámetros desconocidos a determinar.

$$z = \underbrace{\alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k}$$

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

$$= \frac{1}{1 + e^{-(\alpha + \sum \beta_i X_i)}}$$

Figura 63: Modelo regresión logística [22].

Scikit-learn proporciona la clase *LogisticRegression* para trabajar con este modelo. En esta clase encontramos numerosos parámetros, los cuales sirven para modificar la forma en la que se aplica regularización en el modelo, los algoritmos de optimización, etc. Los parámetros empleados en este proyecto son:

- *penalty*: especifica la norma empleada en la regularización, observar Tabla 2: Función de error regresión lineal con y sin regularización. Sus valores pueden ser:
    - *l1:* regularización *Lasso*.
    - *l2:* regularización *Ridge*.
    - *elasticnet:* combinación de ambas regularizaciones Lasso y Ridge. El peso que tiene cada una de ellas se regula a través del parámetro *l1_ratio (l1_ratio * Lasso + (1- l1_ratio) * Ridge).*
    - *none:* sin ninguna regularización.
- *solver:* algoritmo empleado para optimizar el problema, es decir, para que su función de coste sea lo menor posible. Están disponibles las siguientes opciones:
    - *newton-cg:* cómo podemos observar en la publicación *Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente*: "*El método de Newton es uno de los algoritmos conocidos como de segundo orden, ya que hace uso de la Hessiana. Tiene como objetivo encontrar las mejores direcciones de variación de los parámetros haciendo uso de las derivadas segundas de la función de error. Para ello, se hace uso del desarrollo de Taylor de orden 2 de f, que se puede aproximar por alrededor del conjunto inicial de parámetros,* w_0" [23].
    - *lbfgs (Limited-memory Broyden-Fletcher-Goldfarb-Ahanno Algorithm):* método similar al *newton-cg* pero menos costoso. El método *newton-cg* es muy costoso debido al cálculo de la Hessiana y su inversa. Por otro lado, *lbfgs* emplea una aproximación a esta inversa en lugar de calcular ambos pasos [23].
    - *liblinear (Library for large Linear Classification):* emplea el algoritmo de descenso coordinado, que realiza una búsqueda en línea en una dirección de coordenadas (o en una dirección aleatoria en cada iteración) a partir del punto en que se encuentra en cada iteración. [24].
    - *sag (Stochastic Average Gradient):* el método SAG optimiza la suma de funciones convexas suaves, siempre que sea un número finito de funciones. Su coste es independiente al número de términos de la suma, sin embargo, al incorporar una memoria de valores gradientes anteriores, este método tiene una convergencia más rápidas que otros métodos cómo el descenso del gradiente (SG) [25].
    - *saga*: variante del algoritmo *sag* que se adapta a cualquier convexidad del problema [26].
- *tol:* tolerancia empleada en los criterios detención.
- *max_iter:* máximo número de iteraciones para que el algoritmo de optimización converja.
- *C:* inverso al parámetro de regularización. Cuanto menor sea este valor, mayor será la regularización.

Un aspecto importante a tener en cuenta es que no todos los algoritmos de optimización se pueden emplear con las normas de regularización disponibles. En la siguiente tabla vemos las combinaciones posibles:

	L1 (Lasso)	*L2 (Ridge)*	*Elasticnet*	*Sin regularización*

newton-cg		Sí		Sí
lbfgs		Sí		Sí
liblinear	Sí	Sí		
sag		Sí		Sí
saga	Sí	Sí	Sí	Sí

Tabla 6: Posibles combinaciones penalty-solver en LogisticRegression.

Ahora debemos determinar cuál es el mejor método de optimización y regulación, qué tolerancia es la adecuada, qué valor de C es el óptimo para nuestra regulación, etc. Para ello, *Scitik-learn* nos facilita la clase *GridSearchCV* que empleando validación cruzada, nos permite obtener la mejor combinación posible de los parámetros empleados en nuestro modelo. Sus principales parámetros son:

- *estimator*: modelo de aprendizaje que vamos a emplear.
- *param_grid*: diccionario con el nombre de los parámetros cómo claves junto a una lista de sus posibles valores.
- *cv*: estrategia de validación cruzada. Si le damos un valor entero, especifica el número de pliegues a emplear. Emplearemos una validación cruzada de 3 pliegues en todas las pruebas.
- *scoring*: método empleado para evaluar las predicciones obtenidas. En nuestro caso se emplea *neg_log_loss*.

En los modelo empleados en la regresión logística se han establecido el siguiente *param_grid*:

```
grid_values = [
            {'clf__C': [0.001, 0.01, 1, 10, 100],
               'clf__penalty': ['l1'],
               'clf__solver' : ['liblinear', 'saga'],
               'clf__tol' : [0.000001, 0.00001, 0.0001, 0.001, 0.01],
               'clf__max_iter' : [10, 25, 100, 250, 500]},

            {'clf__C': [0.001, 0.01, 1, 10, 100],
               'clf__penalty': ['l2'],
               'clf__solver' : ['liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga'],
               'clf__tol' : [0.000001, 0.00001, 0.0001, 0.001, 0.01],
               'clf__max_iter' : [10, 25, 100, 250, 500]}
            ]
```

Figura 64: param_grid aplicado en regresión logística.

En total se han probado 875 modelos diferentes (tanto en *LabelEncoder* como en *One Hot Encoder*), en cada uno de ellos se ha realizado validación cruzada de 3 pliegues (un total de 2.625 modelos simulados), y hemos obtenido como mejor resultado (aquel modelo con menor *log loss*) la siguiente combinación de parámetros:

- *LabelEncoder*:
  - Parámetros:
    - *penalty*: L1 (*Lasso*).
    - *solver*: liblinear.
    - *C*: 10.
    - *tol*: 0,01.
    - *max_iter*: 100.

    - Mejor resultado: 0,5353.
- *OneHotEncoder*:
    - Parámetros:
        - *penalty*: L2 (*Ridge*).
        - *solver*: newton-cg.
        - *C*: 0,01.
        - *tol*: 0,01.
        - *max_iter*: 100.
    - Mejor resultado: 0,5319.

5.2 Árboles de decisión.

Los árboles de decisión son gráficos empleados para tomar decisiones. En cada nodo en el que se produce una ramificación, o nodo de decisión, se estudia una sola característica en concreto, y a través de un umbral específico se divide en ramas diferentes. Al llegar a un nodo final obtendremos la decisión sobre a qué clase pertenece cada uno de los registros empleados en el modelo [27].

Este modelo emplea el algoritmo ID3 (inducción mediante árboles de decisión), que funciona de la siguiente manera. Siendo *S* el conjunto de datos etiquetados, al principio, el árbol de decisión sólo tiene un nodo en el que se encuentran todos los ejemplos de S. En este punto de partida, comenzamos con un modelo constante:

$$f_{ID3}^{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},y)\in\mathcal{S}} y.$$

Figura 65: Fórmula inicial algoritmo ID3 [27].

En este punto, se va dividiendo el conjunto S en subconjuntos S^+ y S^-. Para ello, se realiza un estudio de todas las características disponibles y de todos los posibles umbrales para determinar cuál es la mejor división.

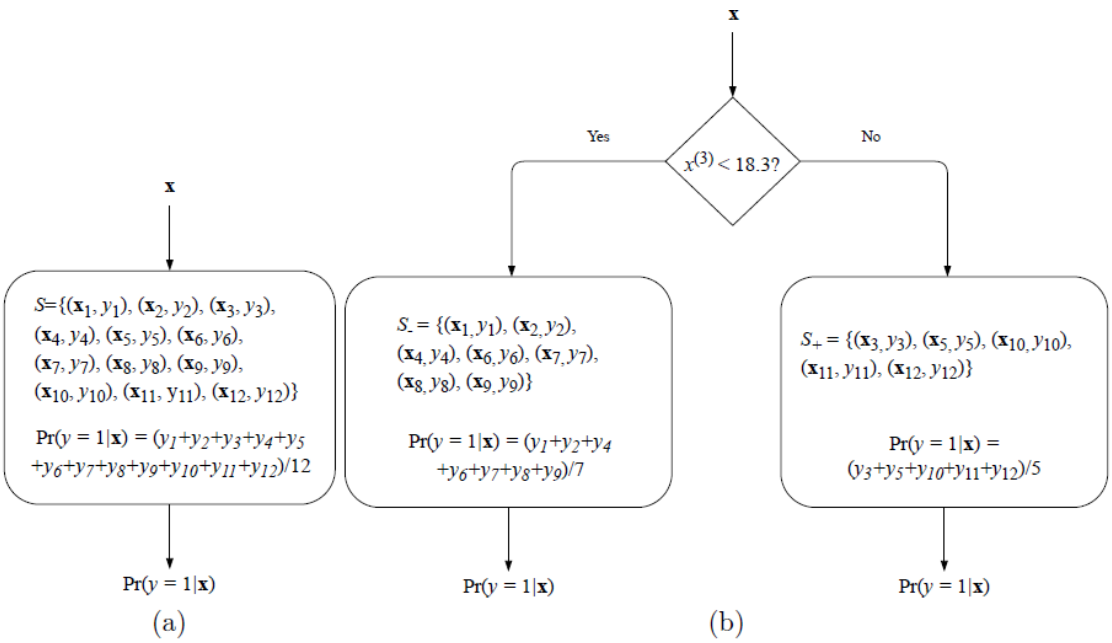

Figura 66: (a) Conjunto de datos S sin dividir. (b) Conjunto de datos S tras la primera división [27].

Para medir la calidad de cada división y así quedarnos con la mejor opción de todas las posibles, se emplea la entropía. La entropía es una medida de incertidumbre que obtiene su valor máximo cuando todos los valores de las variables aleatorias son equiprobables. La entropía de nuestro conjunto de datos S viene dada por la siguiente fórmula (al dividir en dos subconjuntos, la entropía total será la suma ponderada de la entropía de ambos conjuntos):

$$H(\mathcal{S}) = -f_{ID3}^{\mathcal{S}} \ln f_{ID3}^{\mathcal{S}} - (1 - f_{ID3}^{\mathcal{S}}) \ln(1 - f_{ID3}^{\mathcal{S}}).$$

Figura 67: Fórmula de entropía.

El algoritmo converge en un nodo final o nodo hoja cuando ocurre una de las siguientes situaciones:

- Todos los registros del nodo hoja se clasifican correctamente por su nodo anterior.
- No es posible encontrar un atributo para dividir el conjunto de datos.
- La mejor entropía de la nueva división es inferior a la que teníamos anteriormente.
- El árbol alcanza la máxima profundidad posible.

Scikit-learn nos facilita la clase *DecisionTreeClassifier*, con numerosos parámetros para adecuar el modelo de la mejor forma posible a los datos disponibles. Con la intención de obtener el mejor modelo posible, hemos realizado un *GridSearchCV*, como se hizo anteriormente, empleando los siguientes parámetros de la clase *DecisionTreeClassifier*:

- *max_depth*: máxima profundidad del árbol de decisión.
- *min_samples_split*: número de muestras necesarias para dividir un nodo interno.
- *min_samples_leaf*: número de muestras necesarias en un nodo hoja para poder dividir el nodo anterior.

El *param_grid* empleado en estos árboles de decisión ha sido el siguiente:

```
depth = np.arange(1, 21)

grid_values = {'tree__max_depth': depth,
               'tree__min_samples_split': [2, 3, 4, 5, 10, 25, 100],
               'tree__min_samples_leaf' : [1, 2, 3, 4, 10, 25, 100, 150, 250, 500]}
```

Figura 68: param_grid aplicado en los árboles de decisión.

Con este *param_grid se* examinan un total de 1.400 combinaciones diferentes de parámetros (con validación cruzada de 3 pliegues y tanto empleando *LabelEncoder* como *OneHotEncoder*). Tras realizar la prueba con estos modelos, se ha obtenido que la mejor combinación posible de parámetros es la siguiente:

- *LabelEncoder*:
  - Parámetros:
    - *max_depth*: 11.
    - *min_samples_split*: 3.
    - *min_samples_leaf*: 250.
  - Mejor resultado: 0,5093.
- *OneHotEncoder*:
  - Parámetros:
    - *max_depth*: 11.
    - *min_samples_split*: 2.
    - *min_samples_leaf*: 250.
  - Mejor resultado: 0,5121.

5.3 Modelo final.

Una vez realizada la búsqueda de los parámetros óptimos a emplear en nuestro modelo, hemos llegado a la conclusión que la mejor opción es el uso de *LabelEncoder* para tratar las columnas categóricas y posteriormente entrenar con un árbol de decisión con *max_depth* 11, *min_samples_split 3* y *min_samples_leaf* 250.

Hasta este momento sólo hemos utilizado los datos de entrenamiento para determinar el mejor modelo posible. Para ello, hemos empleado validación cruzada con el fin de poder obtener los resultados de cada posible combinación de parámetros en los diferentes modelos de aprendizaje.

Ahora hay que emplear los datos proporcionados por *Kaggle* para realizar el test. Antes de nada, debemos reentrenar nuestro modelo, ya que, como hemos mencionado anteriormente, en ningún momento se han utilizado todos los datos para entrenar nuestro modelo debido a la división entre datos de entrenamiento y validación que se lleva a cabo en la validación cruzada. Por tanto, lo primero que se debe realizar en este momento es entrenar nuestro modelo con la totalidad de los datos de entrenamiento.

Una vez realizado este entrenamiento, podemos proceder a predecir las probabilidades que tiene cada anuncio perteneciente al dataset de test de ser clicado. Con los resultados de esta predicción, se prepara un *dataframe* siguiendo las indicaciones de *Kaggle*, que contiene dos columnas:

- La primera columna será un identificador. Este identificador comienza en 60.000.000 y va aumentando de uno en uno hasta llegar al valor 66.042.134.
- En la segunda columna se debe indicar la probabilidad de que el anuncio sea clicado, es decir, de que tenga valor 1 en su etiqueta *Label*.

Este *dataframe*, será exportado en forma de archivo .csv para poder ser calificado en la competición de *Kaggle*. Tras este proceso, obtenemos un E_{out} de 0,5218 en nuestro mejor modelo. Este E_{out} viene dado por la media de los errores de cada uno de los registros de test. A continuación podemos ver una muestra del *dataframe* subido a la competición *Kaggle*:

	Id	Predicted
0	60000000	0.131512
1	60000001	0.226426
2	60000002	0.417347
3	60000003	0.130210
4	60000004	0.262832

Figura 69: Ejemplo dataframe predicción final.

Ya lo mostramos en el capítulo 3 de este proyecto, pero para entender este error vamos a recordar los valores que este proporciona dependiendo de las predicciones:

- Etiqueta 1 y predicción 0.8: 0.2231
- Etiqueta 1 y predicción 0.6: 0.5108
- Etiqueta 1 y predicción 0.5: 0.693
- Etiqueta 1 y predicción 0.4: 0.9163
- Etiqueta 1 y predicción 0.2: 1.609

Hay que tener en cuenta las limitaciones computacionales que nos hemos encontrado a lo largo de este proyecto, siendo las más significativas:

- No se ha podido emplear el total de los datos de entrenamiento disponibles. Se han empleado 10 millones de registros, lo que equivale a un 21,8% del dataset.
- No se ha podido emplear la técnica *OneHotEncoder* en todas las columnas categóricas. Lo que ha derivado en el uso de esta técnica en sólo 2 columnas o el empleo de *LabelEncoder*.

Esos aspectos lógicamente han tenido su impacto en los resultados obtenidos. Los datos, como ya se ha comentado, se han obtenido de una competición de *Kaggle* ya finalizada. Como hemos mencionado, el error cometido, E_{out}, (calculado como *log loss*) es 0,52. Para poner en perspectiva este valor, resumimos a continuación algunos resultados de dicha competición: una predicción aleatoria daba un E_{out} de 1, el empleo de

regresión logística (pero utilizando todos los datos para entrenar) daba un error de 0,48 y el ganador de la competición logró un error de 0,44.

6. Conclusiones y líneas futuras.

En este último capítulo vamos a proceder a mostrar las conclusiones extraídas a lo largo del desarrollo de este Trabajo Fin de Grado, así como posibles líneas futuras a seguir en él.

6.1 Conclusiones.

El principal objetivo fijado al comienzo de este proyecto ha sido la creación de un modelo de aprendizaje capaz de predecir la probabilidad de éxito en los anuncios publicitados en plataformas digitales.

Este objetivo se ha cumplido en varias fases como hemos podido ir observando en los capítulos 4 y 5 de este proyecto. Es verdad que no se ha obtenido el resultado esperado al comienzo de este trabajo, sin embargo, sí se han podido aplicar todas las ideas estudiadas. Una de las posibles causas de este resultado es debido a los recursos disponibles, ya que no se han podido aplicar las ideas iniciales sobre todos los datos que se deseaba y hemos tenido que emplear un dataset resumido del inicial.

La primera conclusión que podemos extraer es la importancia que tiene conocer los datos con los que se trabaja, ya que al disponer de un dataset totalmente anonimizado no hemos podido analizar la importancia de las características en el conjunto de datos, ni realizar todo el preprocesado que nos hubiera gustado hacer, como puede ser un rellenado de los datos más acorde con cada característica disponible frente al proceso más general que se ha llevado a cabo aquí.

También se ha observado cómo modelos sencillos cumplen perfectamente la función que buscamos en cada modelo. Por tanto, no es necesario recurrir a modelos muy complejos, cuando otros más sencillos generan unos resultados similares e incluso mejores.

La conclusión que podemos extraer de este TFG es observar las grandes ventajas y beneficios que puede originar la aplicación del aprendizaje automático en las empresas, en concreto, aquellas que empleen el marketing online, algo fundamental en la actualidad, permitiéndoles entre otras cosas analizar el mercado para ajustar sus anuncios con el fin de ahorrar creando los mejores anuncios posibles, o encontrando los mejores posibles clientes.

6.2 Líneas futuras.

Una vez cumplido el principal objetivo de este TFG, tenemos la posibilidad de afrontar diversas líneas futuras con las que fundamentalmente intentar conseguir un resultado mejor al visto hasta el momento. Alguna de las opciones que se pueden plantear para mejorar estos resultados son:

- Realizar un modelo más completo, empleando todos los datos ofrecidos por *Kaggle*, tanto la totalidad de los registros disponibles como de las características de estos. Para ello necesitaríamos una capacidad computacional superior a la empleada hasta este momento. Para ello, podríamos emplear *Google Cloud* o *Amazon Web Services* que nos permitirán tener una capacidad de cálculo superior en la nube.
- Analizar otros modelos de aprendizaje en profundidad con el fin de poder aplicarlos como alternativa a los empleados hasta el momento.

Por otro lado, otro camino que se podría seguir es aplicar estas mismas fases en otros dataset, en los que podamos conocer las características de los anuncios que contienen.

Referencias

[1] *Diccionario de la Real Academia Española.* Página web oficial: http://www.rae.es/

[2] Antonio Checa Godoy. *Historia de la publicidad.* Netbiblo, 2008.

[3] Juan José Castaño y Susana Jurado. *Marketing digital (Comercio electrónico).* Editex, 2016.

[4] *Centro de asistencia Google.* Página web oficial: https://support.google.com/

[5] *Una compañía transformadora, comprometida con la excelencia.* Página web oficial: https://www.criteo.com/es/

[6] *Proyect Jupyter.* Página web oficial: https://jupyter.org/

[7] *Pandas.* Página web oficial: *https*://pandas.pydata.org/

[8] *Numpy.* Página web oficial: http://www.numpy.org/

[9] *Matplotlib.* Página oficial: https://matplotlib.org/

[10] *Scikit-learn*. Página oficial: https://scikit-learn.org/stable/

[11] Andrew Goodman. *Google Adwords*. McGraw Hill, 2005.

[12] *Google Ads Fundamentals.* Página oficial: https://academy.exceedlms.com/student/path/3132-google-ads-fundamentals

[13] *Twitter Ads.* Página oficial: https://ads.twitter.com/

[14] A. Moreno, E. Armengol. *Aprendizaje automático*. Edicions UPC, 1994.

[15] Y. S. Abu-Mostafa, M. Magdon-Ismail. *Learning from data*. Amlbook.com, 2012.

[16] Jason Broenlee. *A gentle introduction to probability scoring methods in Python.* Página en la que se puede encontrar el artículo: https://machinelearningmastery.com. Julio 2019.

[17] Artículo *Overfiting and regulariztion.* Se puede consultar en: https://machinelearningmedium.com. Julio 2019.

[18] Artículo *Memorizing is not Learning.* Se puede consultar en la web: https://hackernoon.com. Julio 2019.

[19] Tutorial *A complete tutorial on ridge and lasso regression in Python*. Se puede realizar accediendo a la página web https://www.analyticsvidhya.com. Julio 2019.

[20] Librería w*ordcloud*. Se puede consultar en http://amueller.github.io/word_cloud/index.html

[21] *CTR y CPC medio en Google Adwords por industria 2018*. Estudio realizado por la agencia de marketing digital 2+2websites. Se puede consultar en la página web: https://2mas2websites.com. Agosto 2019.

[22] David G. Kleinbaum, Mitchel Klein. *Logistic Regression*. Springer, 2010.

[23] Fernando Sancho Caparrini. Artículo. *Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente*. Se puede consultar en el departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Sevilla. Publicación disponible en: http://www.cs.us.es/~fsancho/?e=165. Agosto 2019

[24] *Liblinear*. Página oficial: https://www.csie.ntu.edu.tw/~cjlin/liblinear/#document

[25] Mark Schmidt, Nicolas Le Roux, Francis Bach. *Minimizing Finite Sums with the Stochastic Average Gradient*. 2015.

[26] Aaron Defazio, Francis Cach, Simon Lacoste-Julien. *SAGA: A Fast Incremental Gradient Method WithSupport for Non-Strongly Convex CompositeObjectives*. 2014.

[27] Andriy Burkov. *The hundred-page machine learning book*.