UNIVERSITAT POLITÈCNICA DE VALÈNCIA

MIARFID Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

# Predicción del *target* en anuncios políticos de Facebook

**TRABAJO FIN DE MASTER**

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

*Autor:* Marcos Esteve Casademunt

*Tutores:* Carlos David Martínez Hinarejos
Tomás Baviera Puig

Curso 2019-2020

# Resum

L'augment d'usuaris a la xarxa ha afavorit el desenvolupament de nous models de negoci on empreses com Facebook, Google o Twitter permeten introduir anuncis a les seves plataformes realitzant per a això un pagament. Aquest tipus de pagament sol estar determinat en funció del nombre d'impressions, el nombre de clics en un determinat objecte o el nombre de transaccions realitzades.

Les plataformes de creació d'anuncis permeten al anunciant determinar els segments sobre els quals anirà dirigit aquesta publicitat; per exemple, un anunciant podria especificar el sexe, l'edat, la localització, els gustos i un altre tipus de segments sobre els quals vol destacar el seu producte.

En aquest treball s'han d'aplicar diferents tècniques d'aprenentatge automàtic per tal de predir la segmentació en anuncis polítics de Facebook des d'una perspectiva multimodal (text i imatge). S'aprofundirà en l'ús de tècniques de reconeixement de patrons clàssiques, com poden ser l'ús de les bosses de paraules i algoritmes clàssics com les màquines de vectors suport, per a la classificació textual. A més, s'aprofundeix en l'ús de tècniques de *deep learning*, com les xarxes convolucionals per al tractament d'imatges i les xarxes recurrents per al tractament de la informació textual.

Finalment, per la multimodalitat de les dades, és necessari aprofundir en tècniques de combinació de classificadors que permetin, a partir del text i les imatges proporcionades pels anunciants, obtenir les prediccions referents al *target*.

**Paraules clau:** segmentació en anuncis polítics, aprenentatge automàtic, aprenentatge profund, LSTM, CNN, Facebook

# Resumen

El aumento de usuarios en la red ha favorecido el desarrollo de nuevos modelos de negocio donde empresas como Facebook, Google o Twitter permiten introducir anuncios en sus plataformas realizando para ello un pago. Este tipo de pago suele estar determinado en función del número de impresiones, el número de clics en un determinado objeto o el número de transacciones realizadas.

Las plataformas de creación de anuncios permiten al anunciante determinar los segmentos sobre los que irá dirigida dicha publicidad; por ejemplo, un anunciante podría especificar el sexo, la edad, la localización, los gustos y otro tipo de segmentos sobre los que quiere destacar su producto.

En este trabajo se aplicarán distintas técnicas de aprendizaje automático con el fin de predecir la segmentación en anuncios políticos de Facebook desde una perspectiva multimodal (texto e imagen). Se profundizará en el uso de técnicas de reconocimiento de patrones clásicas, como pueden ser el uso de las bolsas de palabras, y algoritmos clásicos, como las máquinas de vectores soporte, para la clasificación textual. Además, se profundizará en el uso de técnicas de *deep learning*, como las redes convolucionales para el tratamiento de imágenes y las redes recurrentes para el tratamiento de la información textual.

Por último, dada la multimodalidad de los datos es necesario profundizar en técnicas de combinación de clasificadores que permitan, a partir del texto y las imágenes proporcionadas por los anunciantes, obtener las predicciones referentes al *target*.

**Palabras clave:** segmentación en anuncios políticos, aprendizaje automático, aprendizaje profundo, LSTM, CNN, Facebook

# Abstract

The increase in the number of users on the network has favoured the development of new business models where companies such as Facebook, Google or Twitter allow you to place ads on their platforms by making a payment. This type of payment is usually determined by the number of impressions, the number of clicks on a certain object, or the number of transactions made.

Ad creation platforms allow the creator of the ad to determine the segments on which the ad will be directed; for example, a creator could specify the sex, age, location, tastes and other types of segments on which he wants to highlight his product.

In this work, different automatic learning techniques will be applied in order to predict the segmentation in Facebook political advertisements from a multimodal perspective (text and image). We will deepen in the use of classic pattern recognition techniques, such as the use of word bags, and classic algorithms, such as Support Vector Machines, for text classification. In addition, we will use Deep Learning techniques, such as convolutional networks for the treatment of images and recurrent networks for the treatment of textual information.

Finally, given the multimodality of the data, it has been necessary to go deeper into techniques of combining classifiers that allow, from the text and the images provided by the advertisers, to obtain the predictions referring to the target.

**Key words:** political ad targeting, machine learning, deep learning, LSTM, CNN, Facebook

# Índice general

# Índice de figuras



# Índice de tablas

# Agradecimientos

Quiero aprovechar estas líneas para agradecer a todas aquellas personas que, de manera directa o indirecta han hecho este camino más fácil.

En primer lugar me gustaría agradecer a mis tutores Carlos y Tomás por haber contado conmigo para el desarrollo de este proyecto, así como por toda la ayuda recibida. Además, agradecer a todo el equipo docente del MIARFID por haberse adaptado con tanta velocidad y excelencia a la situación vivida por la pandemia del COVID-19.

I change to English these lines to thank Datamaran, specially the tech team, for the possibility to be part of their data scientist team; allowing me to learn a lot of technologies needed in the working world as well as to develop some crazy ideas.

También agradecer a aquellas amistades que me han acompañado durante la carrera así como las nuevas incorporaciones. Las reuniones de los viernes o las "ciber-cervezas" durante el periodo de confinamiento permitió descansar y desconectar del estrés de la semana.

Por último, agradecer a mis padres por apoyarme en todas las decisiones que he tomado. Vosotros me habéis enseñado todos los valores que hoy me definen como persona.

A todos vosotros, gracias.

*It's difficult to make predictions, especially about the future* Niels Bohr

# CAPÍTULO 1
# Introducción

## 1.1 Motivación

En la última década, gracias a un incremento sustancial en la capacidad de cómputo de los ordenadores, el aumento de la cantidad de datos disponibles y el avance en numerosas técnicas algorítmicas relacionadas con el aprendizaje a partir de datos, se han conseguido avances en numerosas áreas como el procesamiento del lenguaje natural [1] o la visión por computador [2].

Gracias a los avances anteriormente comentados, se han desarrollado nuevas áreas de investigación relacionadas con el aprendizaje automático. Una de estas áreas que ha ido tomando importancia, sobre todo en el sector del marketing, ha sido la predicción del *target* al cual va dirigido un anuncio. Esta área es de especial interés a la hora de realizar ventas en cualquier medio. Especialmente en internet se convierte en una pieza clave, ya que determinar correctamente el *target* al cual va dirigido un determinado anuncio puede ahorrar costes, así como incrementar el número de ventas.

## 1.2 Objetivos

El objetivo principal de este trabajo consiste en la construcción de un conjunto de sistemas basados en aprendizaje automático que permitan predecir el perfil al que va dirigido un determinado anuncio político en Facebook. En nuestro caso se tratará de predecir ciertos atributos socio-demográficos, como el sexo, la edad y la localización, a partir de las imágenes y el texto proporcionado por el anunciante. Además, se tratará de predecir otros atributos como el alcance del anuncio o el tema tratado. Se han planteado los siguientes subobjetivos:

- Explorar distintas técnicas para la clasificación del texto de los anuncios
- Explorar distintas técnicas relacionadas con la visión por computador para la clasificación de las imágenes de los anuncios
- Combinar las salidas de la clasificación de las imágenes y el texto en un único modelo, explorando por tanto técnicas de combinación de clasificadores

## 1.3 Estructura de la memoria

El presente documento se estructura en un total de 8 capítulos. A continuación se detalla cada uno de ellos.

En el **capítulo 2** se profundizará en el estado del arte de las distintas tecnologías utilizadas. Se comentarán técnicas clásicas de representación y aprendizaje, así como los últimos avances en el campo del *deep learning* aplicados a datos textuales e imágenes. Además, se detallará el funcionamiento de las plataformas de *targeting* para anuncios *online* en sitios web como Facebook o Google.

En el **capítulo 3** se analizará el *dataset* proporcionado por el grupo de investigación MediaFlows de la Universitat de València. Además, se proporcionarán algunos estadísticos y gráficas para poder entender correctamente la distribución de los datos.

En el **capítulo 4** se realizará la propuesta de solución y se describirán las distintas tecnologías empleadas. Posteriormente, en el **capítulo 5** se profundizará en el desarrollo de la solución y se expondrá la experimentación realizada.

Por último, en el **capítulo 6** se comentarán las conclusiones obtenidas, en el **capítulo 7** se comentarán posibles futuros trabajos y en el **capítulo 8** se detallará la relación del presente trabajo con los estudios cursados.

# CAPÍTULO 2
# Estado del arte

El objetivo de este capítulo consiste en realizar una revisión del estado del arte en la utilización de tecnologías de reconocimiento de formas. Además, dada la multimodalidad del trabajo, se comentarán los distintos avances realizados tanto en el campo del procesamiento del lenguaje natural como en el campo de la visión por computador. Por último, se comentarán distintos aspectos importantes en el uso de marketing en medios *online*, así como las distintas plataformas publicitarias que existen.

## 2.1 Reconocimiento de formas

En la última década, gracias a una mejora sustancial en la velocidad de los procesadores, así como la proliferación de las unidades de procesamiento gráfico (GPU), se han podido abordar problemas de gran complejidad computacional. Gracias a esto, ha sido posible realizar avances en campos como la medicina gracias a la aplicación de redes neuronales convolucionales a diversas tareas, como podría ser la detección de neumonía a partir de imagen de rayos X[1]. El uso de estas técnicas podría facilitar la detección precoz de neumonía, siendo de vital importancia a la hora de detectar si un paciente puede o no tener el virus COVID-19. Además, el uso de imagen biomédica también se está empleando en la detección de cáncer o alzheimer.

Además, con la explosión de internet, las redes sociales y demás contenidos multimedia en línea, ha surgido un alto interés por parte de las empresas en el uso de técnicas de reconocimiento de formas, para, de esta forma, ofrecer experiencias personalizadas a sus clientes a partir de las acciones que realizan. Más recientemente se ha comenzado a comercializar productos cuyo *core* o núcleo central es el reconocimiento de patrones, como podría ser el habla en el caso de los altavoces "inteligentes" de Google o Amazon.

El reconocimiento de formas, por tanto, consiste en la aplicación de un conjunto de técnicas computacionales para encontrar patrones en distintos aspectos perceptivos como podría ser la visión, el habla, la escritura, etc.

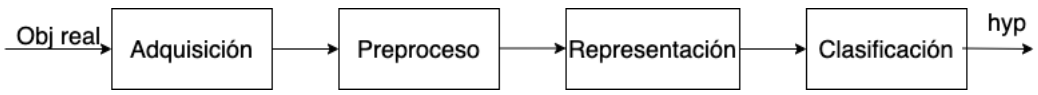


**Figura 2.1:** Arquitectura de un sistema de reconocimiento de formas

Un sistema de reconocimiento de formas o patrones, tal y como puede observarse en la figura 2.1, consiste en un sistema formado por distintos módulos interconectados. En

[1] https://github.com/BIMCV-CSUSP/BIMCV-COVID-19/tree/master/padchest-covid

primer lugar, se dispondría de un objeto en el mundo real. Sobre este objeto se realizaría un proceso de adquisición mediante algún tipo de sensor, como pudiera ser un micrófono, una cámara, etc. Tras el proceso de adquisición se obtiene un objeto digitalizado, el cual se pasará por un preproceso y una extracción de características para obtener una representación que tenga el menor ruido posible. Una vez realizada esta representación se pasaría a un módulo donde, o bien se trataría de aprender la etiqueta de clase a la que pertenece, tratándose por tanto de un problema de clasificación, o bien se trataría de predecir un valor, en cuyo caso se trataría de un problema de regresión [3].

### 2.1.1. Representaciones clásicas

Previo a la fase de entrenamiento del modelo de aprendizaje automático es necesario, en muchas ocasiones, realizar una representación de los datos que permita maximizar la capacidad discriminativa del objeto representado, minimizando a su vez el ruido. En esta sección se van a detallar algunas de las técnicas clásicas que se utilizaban a la hora de representar tanto la información textual como la información contenida en imágenes.

#### Representación textual

A la hora de representar la información textual, la técnica más usada consiste en representar cada documento como un vector de cuentas, conocido como bolsa de palabras o *Bag Of Words* en inglés. El vector está compuesto por valores numéricos donde, para cada posición del vector, se indica la frecuencia de aparición de un *token* en ese documento. Un *token* constituye, por tanto, la unidad mínima de información semántica. Por ejemplo, supongamos las siguientes documentos "se irá con el calor"(Documento 1), "se irá con desinfectante"(Documento 2). Podemos construir una bolsa de palabras, tal y como podemos observar en la tabla 2.1, donde se considera a un *token* como toda aquella secuencia de letras separada por blancos.

**Tabla 2.1:** Ejemplo de representación textual por bolsa de palabras

| | calor | con | desinfectante | el | irá | se |
|---|---|---|---|---|---|---|
| Documento 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Documento 2 | 0 | 1 | 1 | 0 | 1 | 1 |

En ocasiones, a la hora de clasificar tipos de documentos, ciertas secuencias de palabras no son discriminativas, ya que aparecen en muchos de ellos, con el fin de atenuar aquellos *tokens* con presencia en muchos documentos existe una variante de la representación por bolsa de palabras conocida como TF-IDF.

$$tf(t,d) = 1 + \log\ f(t,d)\ si\ f(t,d) > 0$$

$$idf(t,d) = \log(\frac{N}{\sum_{f(t,d)>0} 1})$$

$$tfidf(t,d) = tf(t,d) + idf(t,d)$$

Donde $N$ es el número total de documentos, y *f(t,d)* es la frecuencia de aparición del termino $t$ en el documento $d$.

Uno de los principales problemas que surge al tratar con representaciones del tipo bolsa de palabras es la pérdida de contexto. Este problema surge debido a que un único *token* no es capaz de captar las relaciones con otros *tokens* y, por tanto, se pierde información acerca de la estructura del discurso. Una de las formas de solucionar este problema

consiste en utilizar secuencias de *n-tokens* (n-gramas). La representación por bolsa de n-gramas permite capturar de una forma más adecuada las relaciones entre *tokens*, aunque puede presentar problemas de dimensionalidad y dispersión de los datos.

#### Representación en imágenes

A la hora de realizar la representación de las imágenes, históricamente se realizaban dos aproximaciones. Por una parte, las representaciones globales donde se podría considerar, por ejemplo, el histograma de grises de una imagen. Por otra parte, se podrían utilizar representaciones locales, donde, por ejemplo, se podría representar una cara como la combinación de cejas, ojos, nariz, boca, barbilla y contorno; gracias a este tipo de técnicas se podría representar una imagen como una combinación de partes. Además, una buena representación local debe ser invariante a traslaciones o incluso a oclusiones parciales.

### 2.1.2. Algoritmos de aprendizaje clásicos

Una vez disponemos de los datos preprocesados y se ha realizado la extracción de características de los datos, es necesario aplicar técnicas de aprendizaje automático que permitan obtener las predicciones. En el trabajo desarrollado, se ha profundizado en dos técnicas de aprendizaje supervisado: las máquinas de vectores soporte y la regresión logística.

#### *Support Vector Machines*

Las máquinas de vectores soporte (SVM) [4] es un clasificador que determina fronteras discriminantes lineales. Esta técnica hace uso de los denominados vectores soporte para determinar los hiperplanos separadores entre las muestras, buscando la maximización de la distancia entre clases. En la figura 2.2 se aclaran algunos conceptos relacionados con este tipo de algoritmos.

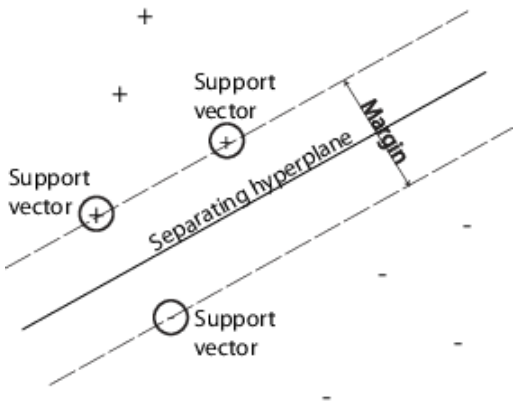


**Figura 2.2:** Frontera de decisión determinada por una SVM

A la hora de desarrollar soluciones prácticas, en muchas ocasiones no es posible encontrar una máquina de vectores soporte de margen máximo. Para solventar este problema y obtener modelos que obtengan mejores generalizaciones surge una extension del modelo conocido como *Soft Margin SVM* [3]. Este nuevo modelo introduce un conjunto de variables de holgura conocidas como **C**. Esta nueva variable constituirá por tanto un hiperparámetro a decidir.

Además, en ocasiones los datos no son linealmente separables y, por tanto, la precisión obtenida por el modelo no es adecuada. Una forma de solventar este problema radica en, tal y como se puede observar en la figura 2.3, introducir una función Kernel [3]. Una función Kernel es una función matemática que trata de, dado un conjunto de datos que no es linealmente separable, transformarlos a una distribución espacial donde sí lo sean. Esta función Kernel también formará parte de los hiperparámetros a decidir.

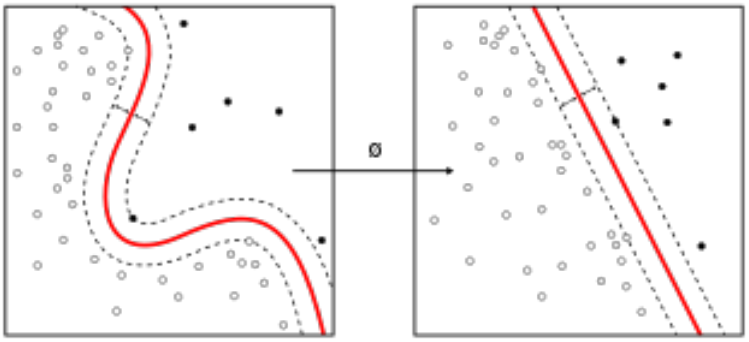


**Figura 2.3:** Aplicación de una función Kernel a un espacio no linealmente separable

#### Regresión logística

Otro de los modelos clásicos ampliamente utilizados, especialmente con datos textuales, es la regresión logística [3]. Se trata de un modelo muy similar a la regresión lineal pero utilizando la función logit, siendo $p$ un valor entre 0 y 1:

$$\text{logit(p)} = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p)$$

El uso de las funciones logit permite, en ocasiones, obtener mejores fronteras de decisión, obteniendo de esta forma mejores generalizaciones.

### 2.1.3. *Deep learning*

Las redes neuronales profundas o *deep learning*, ha ido ganando gran relevancia en los últimos años gracias a la mejora sustancial en las capacidades de cómputo aportadas por las tarjetas gráficas (GPU), así como, la disponibilidad de grandes volúmenes de datos y el desarrollo de numerosas técnicas algorítmicas [5]. El uso de estas técnicas ha permitido obtener resultados del actual estado del arte en numerosas tareas como podría ser ImageNET[2], donde se disponía de un conjunto de millones de imágenes que debían ser clasificadas en 1000 categorías distintas. Además, se podría destacar que la clave del éxito de este tipo de técnicas radica en no solo realizar la tarea de clasificación de los datos, sino también aprender la representación de los mismos. De esta forma, se consigue aprender sistemas *end-to-end* donde a partir de las muestras se aprende la distribución de las clases a las que pertenece, y donde la red aprende tanto la fase de representación como la clasificación al mismo tiempo.

#### Procesamiento del lenguaje natural

En el ámbito del procesamiento del lenguaje natural, el uso de redes neuronales tomó gran importancia con la aparición de los *word embeddings* [6]. Este tipo de representaciones permiten, a diferencia de las representaciones vectoriales por bolsa de palabras,

[2] http://image-net.org/index

capturar en vectores de una dimensionalidad reducida (habitualmente entre 50 y 300 dimensiones) las relaciones semánticas y sintácticas entre diversos términos.

Además, se desarrollaron arquitecturas que permiten lidiar con secuencias, conocidas como redes recurrentes[2]. En la figura 2.4 se puede apreciar el funcionamiento de una red recurrente, donde, dada una secuencia compuesta por *tokens* $(x_1, x_2, ..., x_i, ..., x_t)$, la red empleará el estado de la red codificado en la etapa anterior $w_{i-1}$ y la entrada en la etapa actual $x_i$ para codificar el estado actual de la red $w_i$. Algunas de las arquitecturas recurrentes más conocidas son las LSTM[7] o las GRU [8].

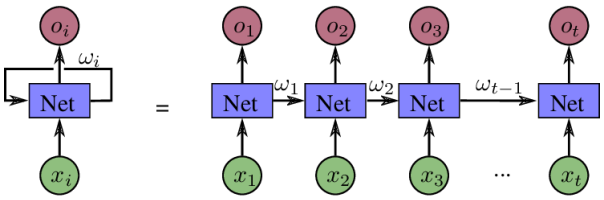


**Figura 2.4:** Red recurrente desplegada

Por otra parte, en 2017 fue presentada la arquitectura Transformer [9]. Tal y como podemos observar en la figura 2.5, se trata de una arquitectura cuya principal diferencia con las arquitecturas basadas en redes recurrentes radica en el uso de modelos de atención y redes *feed-forward*. Esta arquitectura ha demostrado obtener resultados del actual estado del arte en tareas de traducción automática. También se ha visto que obtiene buenas generalizaciones creando modelos de lenguaje o realizando análisis sintáctico de oraciones.

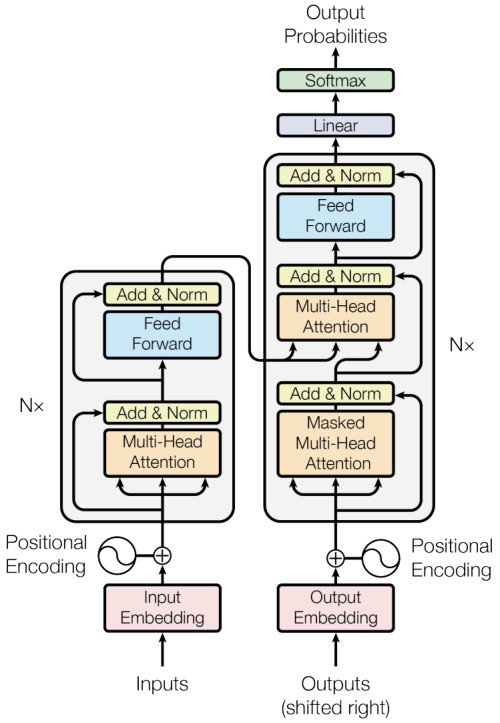


**Figura 2.5:** Arquitectura Transformer

Con el transcurso de los años, y gracias a la evolución de esta tecnología, se han conseguido modelos más elaborados como pueden ser BERT [10] o GPT [11] . Estos modelos han conseguido sembrar el actual estado del arte para numerosas tareas, como puede ser el *Question Answering* [10]. Además estos modelos preentrenados sobre el lenguaje permiten ser la columna vertebral de modelos de clasificación permitiendo, de esta forma, lo que se conoce como *transfer learning* [12].

### Visión por computador

Por otra parte, en el ámbito de la visión por computador también se han desarrollado numerosos avances en el uso de arquitecturas de redes neuronales profundas. Estos avances han sido posible gracias a la aplicación de las redes neuronales convolucionales [2]. Este tipo de redes se componen, principalmente, de dos tipos de capas: las capas convolucionales y las capas de *pooling*.

Por una parte, las capas convolucionales (fig. 2.6) donde se realiza la multiplicación de la imagen de entrada por una matriz denominada Kernel. Habitualmente el tamaño del Kernel es de una dimensionalidad más reducida que la imagen de entrada y, por tanto, las imágenes resultantes tendrán una dimensionalidad inferior.

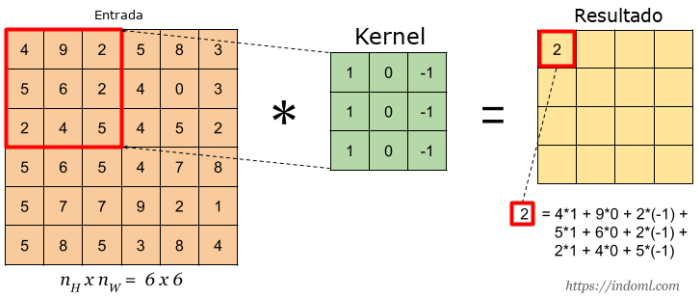


**Figura 2.6:** Ejemplo de aplicación de operación de convolución

Por otra parte, encontramos las capas de *pooling* (fig. 2.7). Este tipo de capas se encargan de reducir el tamaño de la imagen, obteniendo representaciones de más alto nivel. Existen dos populares versiones de la operación *pooling*:

- *Max-pooling*: consiste en calcular el máximo de los valores vistos por la ventana
- *Average-pooling*: consiste en calcular la media de los valores vistos por la ventana

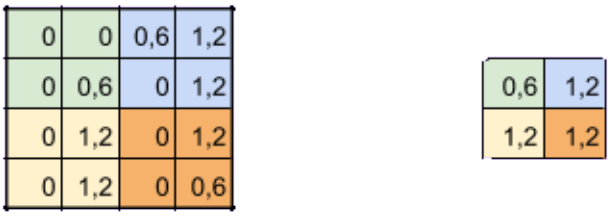

**Figura 2.7:** Ejemplo de aplicación de la operación *max-pooling* con un tamaño de ventana de 2x2

Además, en los últimos años se han ido desarrollando una gran cantidad de arquitecturas y técnicas distintas con el fin de obtener prestaciones del actual estado del arte en numerosas tareas.

Por una parte, podríamos destacar la arquitectura VGG [13]. Esta arquitectura emplea una combinación de capas convolucionales y capas *pooling* agrupadas en 5 bloques. En la figura 2.8 se detallan las distintas arquitecturas propuestas por los autores para la tarea ImageNet.

| ConvNet Configuration | | | | | |
|---|---|---|---|---|---|
| A | A-LRN | B | C | D | E |
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64<br>**LRN** | conv3-64<br>**conv3-64** | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128<br>**conv3-128** | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | | | | | |
| conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256<br>**conv1-256** | conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

**Figura 2.8:** Arquitecturas VGG para la tarea ImageNet (convY: convolucion de YxY, FC: *Fully Connected*)

Por otra parte, uno de los grandes avances en el campo de la visión por computador se produjo con la inclusión de conexiones residuales [14]. Este tipo de conexiones permiten al gradiente pasar directamente desde la imagen origen a cualquier mapa convolucional 2.9.

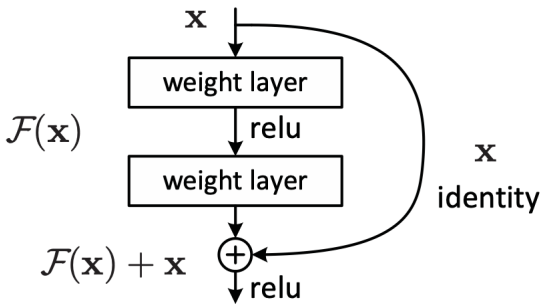


**Figura 2.9:** Conexión residual

### 2.1.4. Combinación de clasificadores

En los últimos años ha surgido un gran interés por la aplicación de técnicas de *ensemble* o combinación de clasificadores. Este tipo de técnicas ha tenido gran auge debido a que, habitualmente, el uso de clasificadores individuales obtienen bajas prestaciones, pero la combinación de múltiples sistemas puede obtener clasificadores con altas prestaciones.

Una de las técnicas más utilizadas consiste en aplicar un esquema de votación [15]. Existen dos tipos de votación. Por una parte, la votación *hard* o mayoritaria, donde la clase asignada viene determinada por la clase que digan la mayoría de los clasificadores. Por otra parte, la votación *soft*, donde la clase asignada se determina haciendo el cálculo del argumento máximo de las sumas de las probabilidades predichas por cada clase. Además, destacar que la votación *soft* suele ser adecuada cuando los clasificadores están bien calibrados.

Otra de las técnicas ampliamente utilizada a la hora de combinar clasificadores consiste en hacer uso de un metaclasificador que aprenda a partir de las salidas de los clasificadores anteriores. Esta técnica se le conoce como *Stacking* [16], busca construir un metamodelo que aprenda cuándo un determinado modelo se va a comportar mejor para darle de esta forma mayor prioridad. Un esquema de la aplicación de esta técnica se puede apreciar en la figura 2.10.

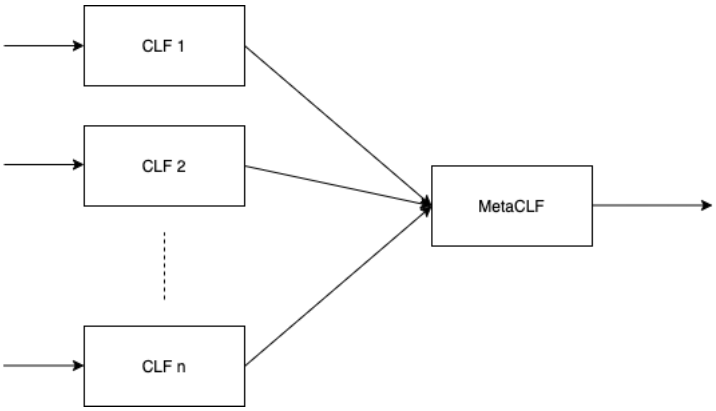


**Figura 2.10:** Esquema de la técnica *Stacking*

Con la aparición de las redes neuronales las técnicas de combinación de clasificadores tomaron especial importancia, sobre todo en tareas multimodales. Para ello, simplemente se deben diseñar los distintos módulos y conectarlos entre sí, entrenando de esta forma un sistema *end-to-end*. Será el propio proceso de aprendizaje, mediante el algoritmo *back-*

*propagation* [5], el que se encargue de aprender la mejor combinación para los distintos módulos. Un ejemplo de esta técnica se puede apreciar en la figura 2.11, presentada en [17]. El objetivo de la arquitectura presentada consistía en detectar el sexo de los autores a partir de las imágenes y el texto que publicaban en Twitter. Diseñaron un módulo textual basado en *word embeddings* y redes neuronales recurrentes. Para el módulo basado en imágenes empleaban redes neuronales convolucionales. Estos dos módulos se combinaban mediante capas de concatenación y capas de *pooling* para, de esta forma, poder entrenar un sistema *end-to-end* que, a partir de las imágenes y el texto de los usuarios, predijera si eran hombres o mujeres.

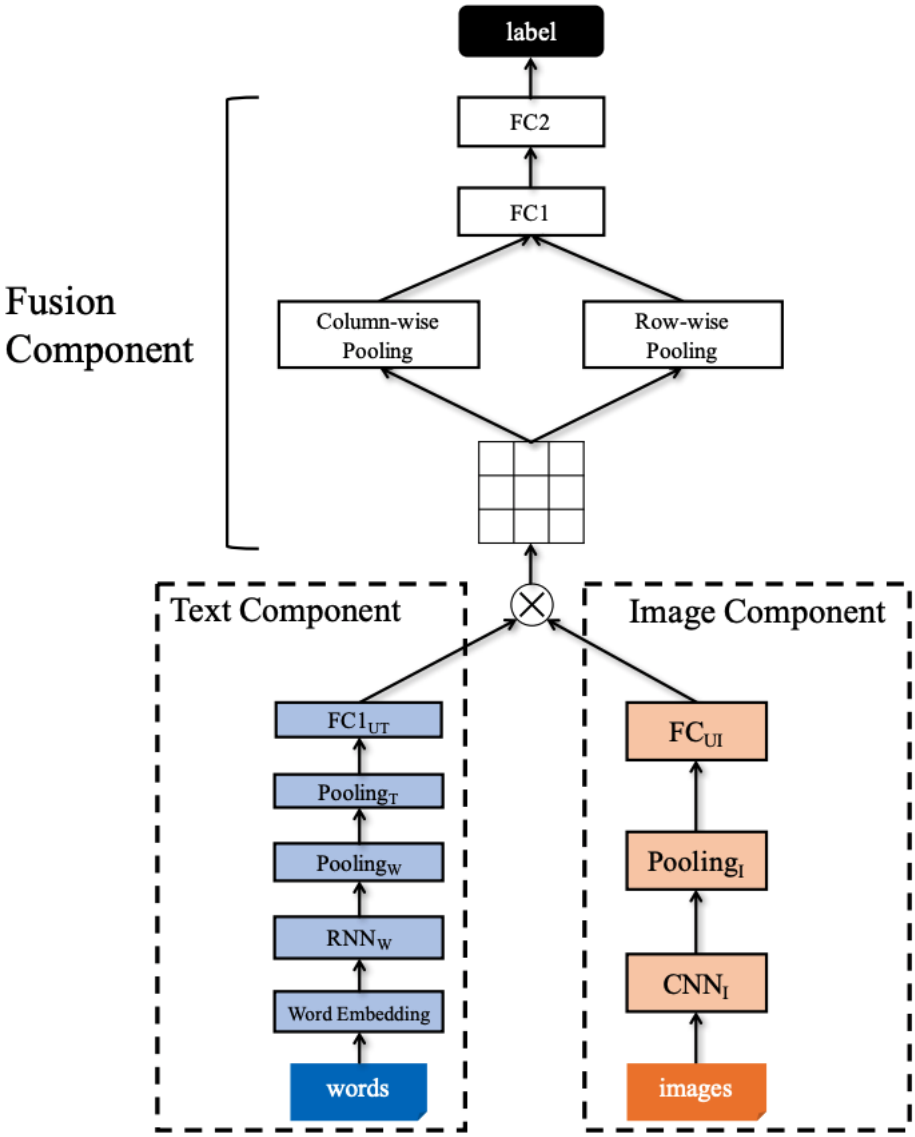


**Figura 2.11:** Esquema de la aplicación de técnicas de combinación de clasificadores multimodales utilizando redes neuronales

## 2.2 Marketing en medios *online*

La publicidad se ha convertido en uno de los pilares principales de internet. Plataformas como Facebook y Google generan gran parte de su facturación gracias a este tipo de sistemas. Concretamente, en 2019 los sistemas de anuncios en internet generaron única-

mente en el mercado estadounidense 124.6 billones de dolares[3]. Estas plataformas permiten a los anunciantes posicionar un anuncio en sus sitios webs o aplicaciones a partir de un coste, el cual puede definirse según diversos criterios. Los criterios más habituales son[4]:

- Coste por venta (CPV), este sistema únicamente cobra si el usuario que accede a la aplicación acaba comprando el producto.
- Coste por click (CPC), este sistema cobra por cada click que se realice sobre el anuncio. Actualmente es el sistema más usado por herramientas como Google Adwords o Adsense.
- Coste por mil impresiones (CPM), este sistema permite establecer una cantidad a pagar por cada mil impresiones del anuncio en la plataforma.
- Coste por acción (CPA), en este tipo de sistemas el anunciante únicamente paga cada vez que un usuario realiza una determinada acción, bien sea descargarse una aplicación, rellenar un formulario o realizar una compra.
- Coste por *lead* (CPL), en este tipo de sistemas el anunciante únicamente pagará si entra a un determinado sitio y rellena un formulario interesándose por un producto. Es similar al coste por acción.

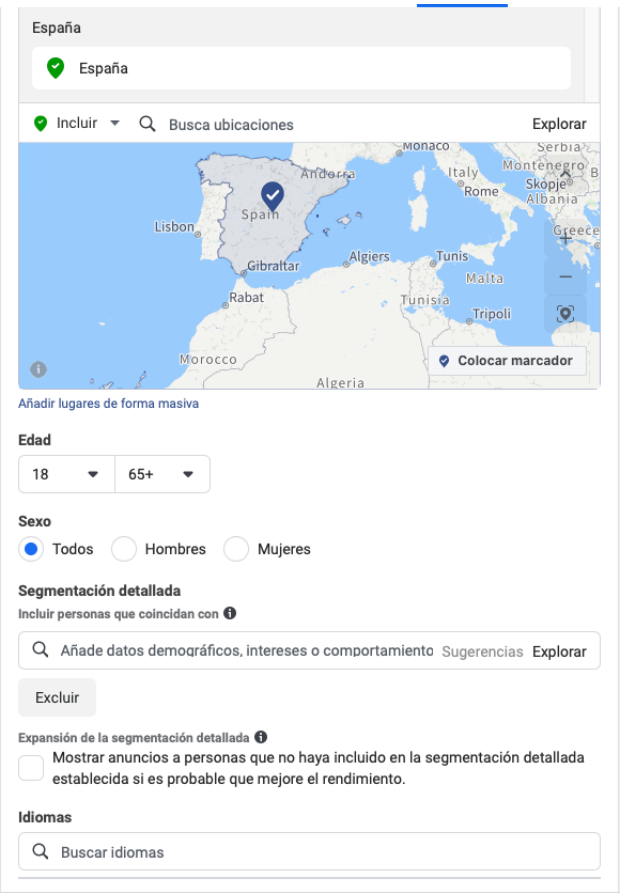


**Figura 2.12:** Interfaz utilizada por Facebook para la segmentación de los anuncios

[3] https://www.statista.com/statistics/275883/online-advertising-revenue-in-the-us-by-half-year/
[4] https://www.funkymk.com/formas-de-pago-en-una-campana-de-marketing-online/

En la actualidad, las dos plataformas más avanzadas y ampliamente utilizadas son las proporcionadas por Google y Facebook. Estas plataformas permiten determinar distintos segmentos sobre los que irá dirigido el anuncio, estableciendo, por ejemplo, si queremos que el anuncio vaya dirigido a jóvenes o adultos, la localización, el sexo de las personas o incluso sus gustos. En la figura 2.12 se puede apreciar la interfaz utilizada por Facebook para la segmentación de anuncios según las características comentadas anteriormente.

# CAPÍTULO 3
# *Dataset*

Para la realización de este trabajo es necesario la obtención de un corpus multimodal de anuncios políticos en Facebook. Este *dataset* ha sido obtenido por los investigadores del proyecto *polcom in Facebook Ads* en el grupo de investigación Mediaflows de la Universitat de València. Los autores emplearon la biblioteca de anuncios de Facebook[1] para la construcción del *dataset*.

Se trata de un *dataset* formado por los anuncios políticos publicados por los principales partidos políticos españoles en las elecciones del 28 de abril de 2019 y las elecciones del 10 de noviembre de 2019. A continuación se detallará el análisis exploratorio realizado sobre los datos proporcionados.

## 3.1 Análisis exploratorio

**Tabla 3.1:** Distribución del número de anuncios por partido político y periodo de elecciones

| Partido | Elecciones | Anuncios | Imágenes | Vídeos |
|---|---|---|---|---|
| **Ciudadanos** | **28A** | 6098 | 1016 | 5078 |
| | **10N** | 2473 | 1394 | 1079 |
| **IU** | **28A** | 13 | 7 | 6 |
| | **10N** | 5 | 1 | 4 |
| **PSOE** | **28A** | 336 | 0 | 735 |
| | **10N** | 285 | 113 | 172 |
| **PP** | **28A** | 3609 | 2126 | 1413 |
| | **10N** | 908 | 251 | 657 |
| **Podemos** | **28A** | 379 | 155 | 224 |
| | **10N** | 545 | 65 | 480 |
| **VOX** | **28A** | 0 | 0 | 0 |
| | **10N** | 44 | 6 | 38 |

Como se puede observar en la tabla 3.1, el partido político que realizó un mayor uso de las campañas de Facebook Ads fue Ciudadanos, mientras que el partido que menos uso hizo de las campañas publicitarias en Facebook fue Izquierda Unida. Cabe destacar que VOX no realizó ninguna campaña publicitaria durante las elecciones del 28 de abril de 2019. Además, en la tabla se puede apreciar cuál es la distribución de los datos con respecto a la multimodalidad, donde, como se puede observar, los partidos políticos suelen hacer un mayor uso de vídeos.

[1] https://www.facebook.com/ads/library/

Por otra parte, el número de datos únicos en el *dataset* es reducido. Tal y como se puede observar en la tabla 3.2, la mayoría de los partidos políticos suelen utilizar el mismo contenido para dirigirse a distintos *targets*.

**Tabla 3.2:** Distribución del contenido único de los anuncios por partido político y periodo de elecciones

| Partido | Elecciones | Textos únicos | Imágenes únicos | Vídeos únicos |
|---|---|---|---|---|
| **Ciudadanos** | **28A** | 197 | 31 | 88 |
| | **10N** | 38 | 21 | 23 |
| **IU** | **28A** | 12 | 7 | 5 |
| | **10N** | 2 | 1 | 1 |
| **PSOE** | **28A** | 42 | 0 | 186 |
| | **10N** | 175 | 22 | 133 |
| **PP** | **28A** | 273 | 320 | 343 |
| | **10N** | 5 | 5 | 14 |
| **Podemos** | **28A** | 143 | 45 | 83 |
| | **10N** | 37 | 5 | 24 |
| **VOX** | **28A** | 0 | 0 | 0 |
| | **10N** | 15 | 1 | 10 |

En las figuras 3.1 y 3.2 se muestran dos ejemplos de anuncios políticos extraídos de la biblioteca de Facebook. En ambos anuncios se pueden apreciar a qué segmentos van dirigidos, así como el contenido de los mismos.

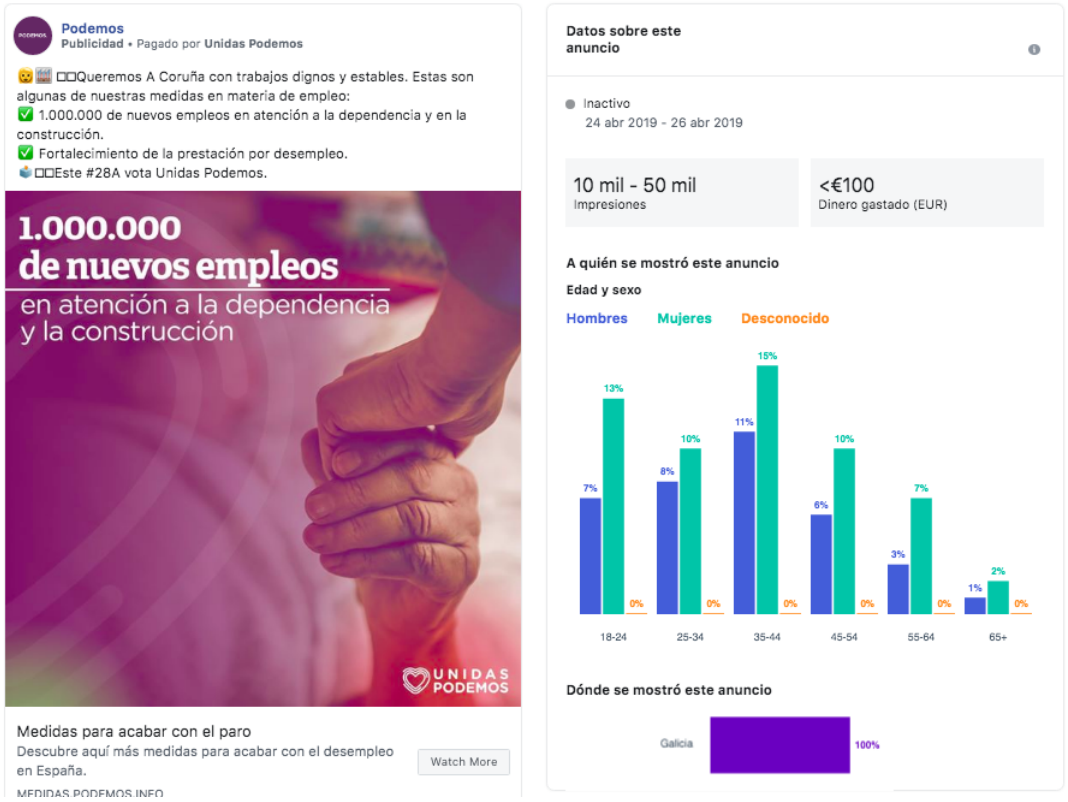


**Figura 3.1:** Ejemplo de anuncio político creado por el partido político Podemos

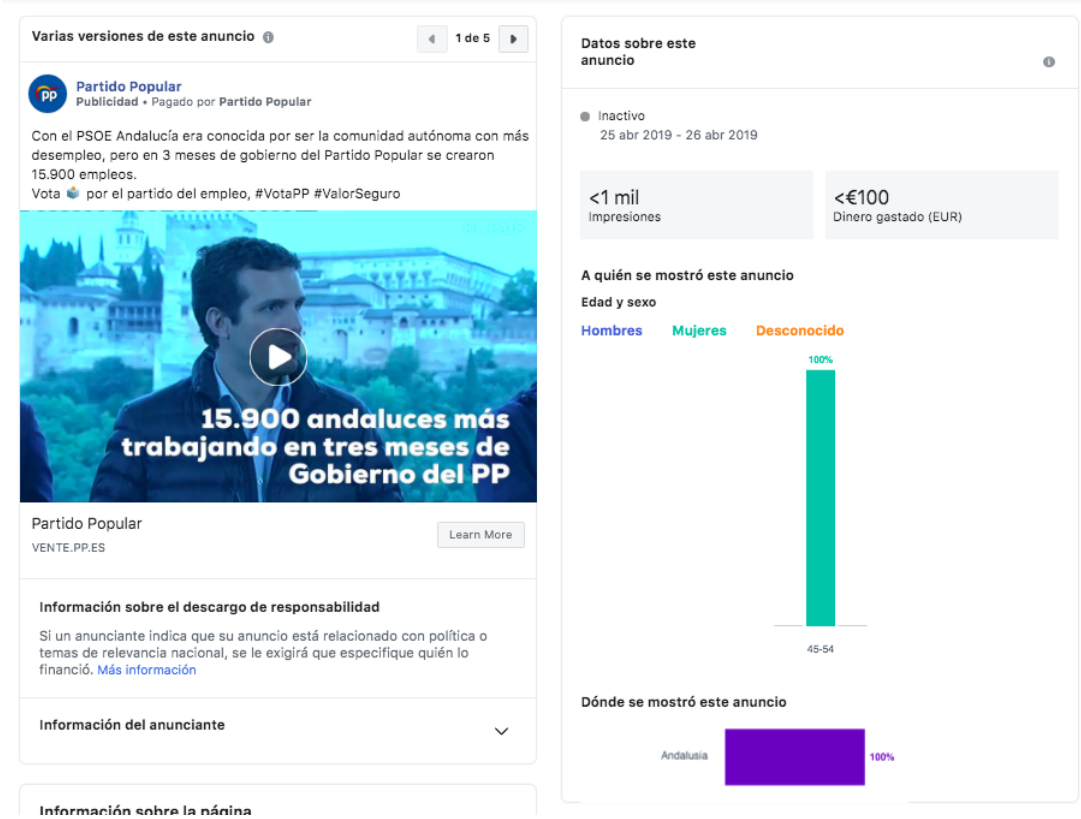


**Figura 3.2:** Ejemplo de anuncio político creado por Partido Popular

### 3.1.1. Distribución por clases

El objetivo de este apartado radica en determinar cuál es el público objetivo mayoritario en los anuncios del *dataset* proporcionado. Para ello se analizará cuáles son las clases mayoritarias para la edad, sexo y distribución geográfica. Además, en última instancia, se analizará cuáles son los temas más habituales en los anuncios, el gasto económico habitual por campaña publicitaría y cúal es el impacto de las campañas en términos de número de impresiones.

#### Sexo

En cuanto al sexo de los anuncios existen tres posibles *targets*: mujer, hombre o desconocido. En la figura 3.3 se puede apreciar la distribución de los anuncios según el sexo mayoritario al que van dirigidos. Tal y como se observa, la mayoría de anuncios tienen una mayor visibilidad por mujeres (67.29 %) frente a los hombres, con un 32.71 % y el sexo desconocido, con 0 %.

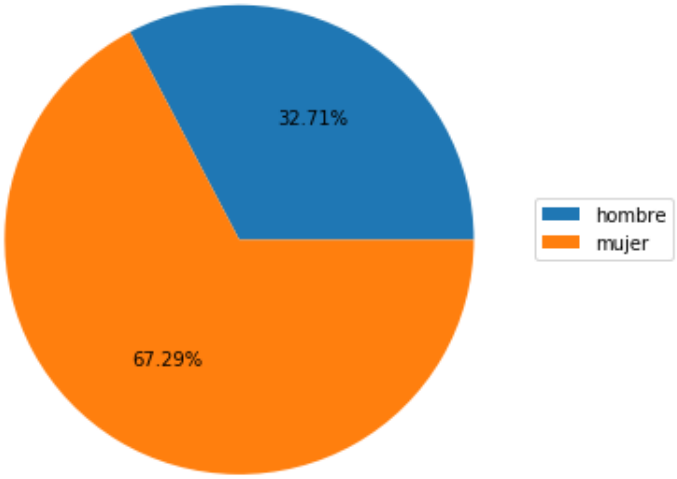


**Figura 3.3:** Distribución del *target* de género (máximo) en los anuncios

### Edad

En cuanto a la edad, la biblioteca de anuncios proporciona 6 rangos distintos: 18-24, 25-34, 35-44, 45-54, 55-64, 65+. Siguiendo la misma estrategia y etiquetando las muestras dependiendo del rango que presenta el porcentaje mayoritario, obtenemos la distribución de la figura 3.4.

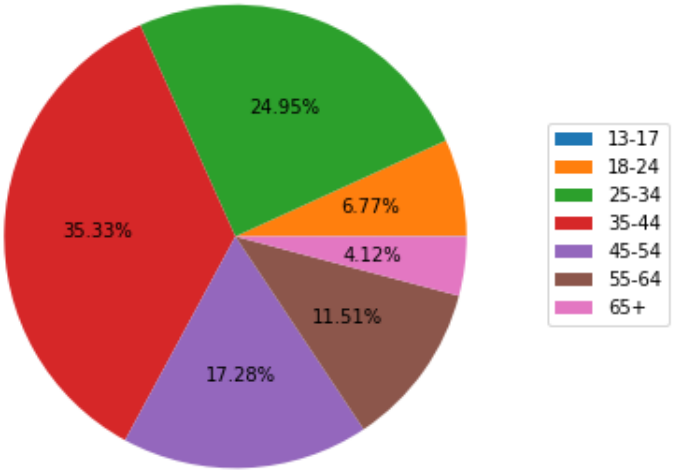


**Figura 3.4:** Distribución del *target* de edad (máximo) en los anuncios

Tal y como puede observarse, la franja con un mayor número de anuncios se encuentra entre los 35 y 44 años, seguida de la franja comprendida entre los 25 y 34 años. Además, destacar que la franja entre los 13 y 17 años no tiene representación en la gráfica, ya que los menores no pueden votar en España.

### Distribución geográfica

La biblioteca de anuncios de Facebook proporciona una descripción de cuáles son las comunidades autónomas alcanzadas por el anuncio. Facebook es capaz, por tanto,

de determinar cuál es el porcentaje de usuarios que han visto los anuncios en cada una de las 17 comunidades que componen España. A continuación, en la figura 3.5 se puede apreciar cuál es el impacto de los periodos electorales en cada comunidad autónoma. Para ello se ha considerado, de nuevo, el máximo de los porcentajes por anuncio.

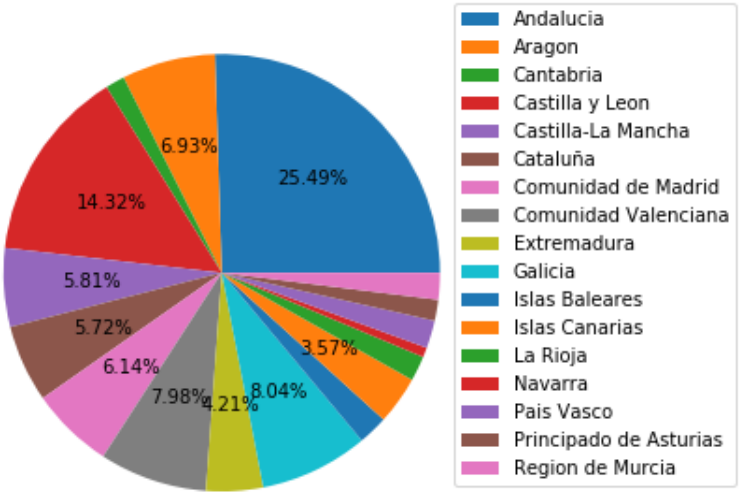


**Figura 3.5:** Distribución del *target* referente a las comunidades autónomas en los anuncios

Tal y como puede observarse, las dos comunidades autónomas con un mayor porcentaje de impacto en los anuncios son las comunidades con un mayor territorio: Andalucía (25.49 %) y Castilla y León (14.32 %). Por otra parte, las comunidades con un menor impacto son Cantabria (1.43 %) y Navarra (0.72 %).

### Impresiones

Los anuncios políticos del *dataset* poseen una etiqueta que indica, mediante un rango, el número de impresiones de un anuncio, es decir, el número estimado de veces que un anuncio se ha mostrado en la red social. Esta etiqueta está comprendida entre los siguientes 8 rangos: <1000, 1K-5K, 5K-10K, 10K-50K, 50K-100K, 100K-200K, 200K-500K y >1M.

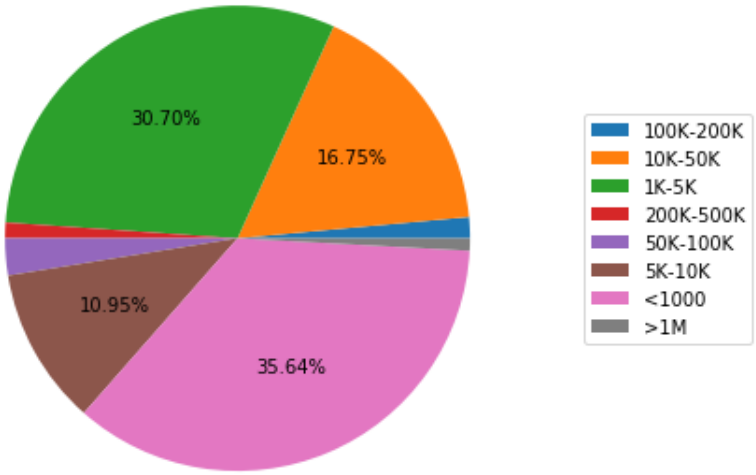


**Figura 3.6:** Distribución de los anuncios por rango de impresiones

La figura 3.6 muestra la distribución de los anuncios según el rango de impresiones en la plataforma. Tal y como se puede observar, el *dataset* no está equilibrado. Las clases que hacen referencia a un número inferior a mil impresiones, así como entre 1000 y 5000 impresiones, son las que poseen un mayor número de muestras, mientras que las clases que referencian a un número de impresiones elevado, como puede ser 200K-500K y más de un millón de impresiones, poseen una representación inferior al 3 %.

### Gasto económico

Por otra parte, la biblioteca de anuncios también proporciona información acerca del gasto realizado por los anunciantes por campaña publicitaria. Para ello proporciona 10 rangos: <100, 100-499, 500-999, 1K-5K, 5K-10K, 10K- 50K, 50K-100K, 100K-200K, 200K-500K y >1M.

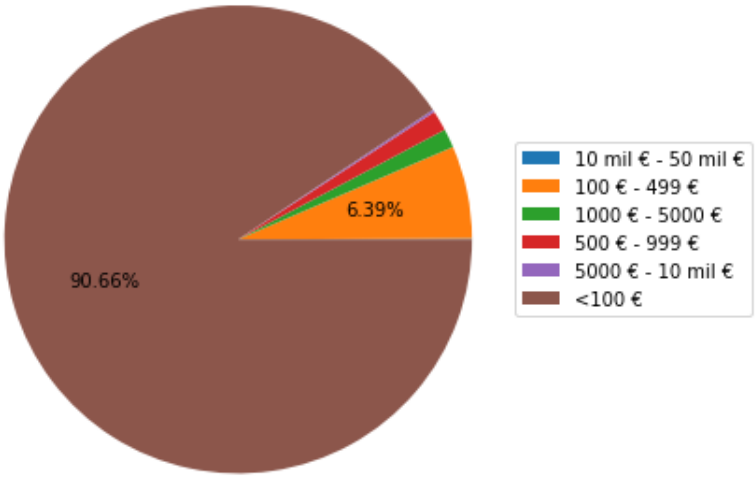


**Figura 3.7:** Distribución de los anuncios según el dinero invertido en la campaña publicitaria

La figura 3.7 muestra la distribución de los anuncios según el dinero invertido en la campaña publicitaria. Tal y como se puede observar, se trata de una distribución de los anuncios muy desigual, donde la inversión habitual realizada por los anunciantes suele ser inferior a 100 euros por anuncio.

### Temas políticos

Para el análisis de los temas políticos más habituales se ha hecho uso de la anotación proporcionada por el grupo de investigación. Consiste en una anotación en 15 clases distintas donde se analiza cual es el tema predominante en el anuncio. Para la anotación, los expertos tuvieron en cuenta el texto escrito, el texto de la imagen y el contenido del vídeo.

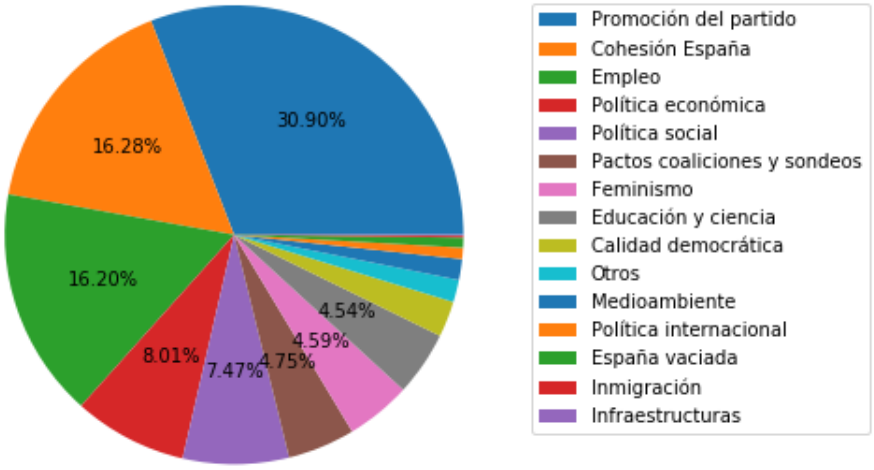


**Figura 3.8:** Distribución de los temas en el *dataset*

Como se puede observar en la figura 3.8, el tema predominante en los anuncios de las elecciones generales del 2019 es la promoción del propio partido con un 30.90 %. Seguidamente se puede observar como los temas cohesión de España y empleo, con un 16.28 % y 16.20 % respectivamente, también tienen una gran importancia en los anuncios. Por contra, los temas relacionados con la infraestructura, la inmigración o la pérdida de la población en los pueblos no son de gran importancia para los partidos políticos.

# CAPÍTULO 4
# Propuesta de solución

El objetivo de este capítulo consiste en detallar la propuesta de diseño de los sistemas multimodales para la predicción de las distintas características demográficas en los anuncios políticos publicados en Facebook. Además, se comentarán las distintas tecnologías empleadas en el proceso de tratamiento y explotación de los datos proporcionados.

## 4.1 Diseño de la propuesta

Para resolver el conjunto de problemas propuestos se ha optado por diseñar un conjunto de sistemas multimodales donde se combinen las salidas tanto de los módulos textuales como los módulos visuales basado en imágenes, para obtener las distintas predicciones. En la figura 4.1 se puede observar el esquema de un sistema de combinación de clasificadores multimodal.

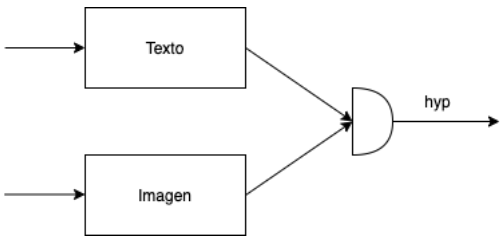


**Figura 4.1:** Esquema de la aplicación de combinación de clasificadores multimodal

### 4.1.1. Reconocimiento textual

Para el reconocimiento textual se propone realizar una comparación de tres sistemas distintos basados en tecnologías clásicas y técnicas de *deep learning*.

En primer lugar, se realizará una representación del tipo *Bag Of Words* comentada en el apartado 2.1.1 y se explorarán distintos hiperparámetros, así como distintos modelos de aprendizaje como los descritos en la sección 2.1.2.

Seguidamente se hará uso de redes neuronales recurrentes y *word embeddings* y se estudiará la contribución de distintos hiperparámetros como el tamaño de las redes LSTM o el tamaño de los *embeddings*.

### 4.1.2. Reconocimiento en imágenes y vídeo

Por otra parte, a la hora de construir los sistemas de reconocimiento para imagen se ha optado por emplear técnicas de *deep learning*. Se explorarán distintas arquitecturas como VGG [13] o Resnet [14]. Se explorará el uso de modelos preentrenados sobre grandes colecciones de datos, como pudiera ser el caso de la base de datos ImageNet[1].

## 4.2 Tecnología empleada

Para el desarrollo del proyecto se ha empleado el lenguaje de programación Python, debido a su sencillez y a la alta integración que dispone con numerosas librerías para el análisis y explotación de datos. A continuación se detallarán algunas de las más relevantes.

#### Pandas

Pandas[18] [19] es una librería ampliamente utilizada en la industria para manipular y analizar colecciones de datos. En particular, la librería provee herramientas para crear y manipular tablas numéricas y series temporales. Para ello, la librería crea un objeto *DataFrame*, sobre el cual se permite realizar agrupaciones y operaciones sobre los datos contenidos.

#### Scikit learn

Scikit-learn[20] es un *toolkit* desarollado sobre las librerías Numpy, Scipy y matplotlib centrado principalmente en la construcción de modelos, de forma sencilla, para tareas de clasificación o regresión. Dispone de una gran variedad de algoritmos supervisados como pueden ser las máquinas de vectores soporte, la regresión logística y perceptrones multicapa, entre otros. También dispone de algoritmos de aprendizaje no supervisado como podría ser K-means[21].

Además, dispone de una gran cantidad de técnicas de selección de modelos como pueden ser la validación cruzada y la búsqueda en *grid*, así como técnicas de reducción de dimensionalidad como PCA y LDA.

#### NLTK y Spacy

NLTK[2] y Spacy[3] son dos librerías ampliamente utilizadas para tareas de procesamiento de lenguaje natural. La principal diferencia entre ellas radica en el público al cual van dirigido. Mientras que NLTK está pensado para un entorno más académico, Spacy está pensada para un entorno más industrial, por lo que muchas de las funciones están optimizadas para su eficiencia. Además, dispone de una interfaz más intuitiva, con lo que se facilita la tarea al programador.

#### Tensorflow-Keras

Tensorflow[22] es un *framework* diseñado para realizar computación numérica, pensado especialmente para realizar investigación en el campo del aprendizaje automático

[1] http://www.image-net.org
[2] https://www.nltk.org
[3] https://spacy.io

y *deep learning*. La librería está pensada para representar la computación como un grafo donde los nodos representan las operaciones matemáticas y las aristas representan *arrays* multidimensionales (tensores). Estos tensores son comunicados entre los nodos con el objetivo de realizar el cómputo.

Por encima de Tensorflow se situa Keras[4]. Esta librería ofrece una API de alto nivel con la que se puede prototipar soluciones de *deep learning* de una forma sencilla y rápida. Además, Keras ofrece implementaciones de las principales redes recurrentes y convolucionales utilizadas habitualmente, tanto en investigación como en la industria. A partir de la versión 2.0 de Tensorflow, Keras viene integrado dentro de la librería.

### Seaborn

Seaborn[5] es una librería construida sobre Matplotlib[23] y pensada para ser utilizada en tareas de visualización de datos. Proporciona una interfaz de alto nivel que permite realizar complejas visualizaciones con pocas líneas de código.

[4] https://keras.io
[5] https://seaborn.pydata.org

# CAPÍTULO 5
# Desarrollo de la solución

En este capítulo se detallará la construcción de los sistemas, haciendo especial hincapié en las distintas partes que lo componen. Además, se detallará el preproceso seguido tanto para la información textual como para las imágenes. Por último, se expondrán los resultados obtenidos para cada una de las modalidades así como en la tarea combinada.

## 5.1 Particiones

A la hora de evaluar los sistemas se ha optado por utilizar una partición *hold out* donde el 80 % se emplea para entrenamiento y el 20 % para test.

Para los sistemas textuales se ha empleado un total de 11756 muestras para entrenamiento y 2939 muestras para test. El tamaño del vocabulario es aproximadamente 3800 *tokens* y la longitud máxima de un anuncio después de ser tokenizado es de 319 *tokens*.

Para los sistemas basados en imágenes, así como el sistema multimodal, se ha utilizado un conjunto más reducido, ya que no todos los anuncios disponen de imágenes. Concretamente se han utilizado 4147 muestras para entrenamiento y 987 muestras para test. Las imágenes del *dataset* pueden tener distintas resoluciones. Con el fin de unificar los formatos se han re-escalado todas a 224x224 píxeles y se han respetado los 3 canales rgb.

## 5.2 Métricas empleadas

A la hora de evaluar los sistemas desarrollados es necesario disponer de un conjunto de métricas que permitan medir la bondad de los modelos obtenidos.

### 5.2.1. *Accuracy*

El *Accuracy* o exactitud trata de modelar la bondad del modelo realizando el cociente entre el número de aciertos del sistema entre el total de muestras del conjunto de test. Formalmente se puede definir como:

$$Accuracy = \frac{|muestrascorrectas|}{|muestras|}$$

### 5.2.2. Precisión

La precisión se puede definir como el porcentaje de documentos recuperados que son relevantes. Formalmente se puede definir como:

$$Precision = \frac{|\{muestrascorrectas\} \cap \{muestraspredichas\}|}{|muestrasrecuperadas|}$$

### 5.2.3. *Recall*

El *recall* o cobertura se define como el porcentaje de documentos relevantes que han sido recuperados. Formalmente se puede definir como:

$$Recall = \frac{|\{muestrascorrectas\} \cap \{muestraspredichas\}|}{|muestrascorrectas|}$$

### 5.2.4. F $\beta$

La F$\beta$ consiste, tal y como se detalla en la ecuación 5.1, en combinar la precisión y el *recall* mediante un factor $\beta$. Habitualmente se emplea $\beta = 1$, obteniendo la métrica conocida como F1.

$$F\beta = (1 + \beta^2) * \frac{Precision * Recall}{(Precision * \beta^2) + Recall} \tag{5.1}$$

### 5.2.5. Divergencia de Kullback-Leibler

La divergencia de Kullback-Leibler[24] es una métrica desarrollada por los investigadores Solomon Kullback y Richard Leibler en 1951. Formalmente, tal y como se define en la ecuación 5.2, se trata de una medida para calcular la diferencia entre dos distribuciones de probabilidad P y Q. Generalmente, P representa la verdadera distribución de probabilidad mientras que Q se trata de la distribución de probabilidad obtenida por un determinado modelo.

$$D_{KL}(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)} \tag{5.2}$$

A diferencia de las medidas expuestas anteriormente, la divergencia de Kullback-Leibler es una métrica que debe ser minimizada, ya que, cuanto menor sea el valor mejor es la aproximación entre las distribuciones de probabilidad P y Q.

## 5.3 Predicción del tema

Tal y como se comenta en el apartado 3.1.1, la base de datos contiene un campo en el que se indica el contenido principal del anuncio. Esta codificación fue realizada manualmente por un equipo de dos codificadoras. El objetivo, por tanto, será la construcción de un modelo multimodal capaz de, utilizando las imágenes y el texto de los anuncios, determinar cuál es el tema predominante. A continuación se detallarán las arquitecturas de los distintos módulos, así como los resultados obtenidos.

### 5.3.1. Predicción textual

A la hora de realizar la predicción del tema empleando el texto se ha realizado un preproceso, tokenizando los textos mediante nltk y se han evaluado tres modelos distintos:

- Máquinas de vectores soporte sobre una representación *bag of words* ponderada por TF-IDF y realizando una busqueda en *grid* para determinar los mejores parámetros referentes al tamaño del n-grama, mínima y máxima frecuencia de aparición de un término, kernel empleado y el parámetro de regularización o variable de holgura C.
- Regresión logística sobre una representación *bag of words* ponderada por TF-IDF y realizando una busqueda en *grid* para determinar los mejores parámetros referentes al tamaño del n-grama, mínima y máxima frecuencia de aparición de un termino y el parámetro de regularización C.
- Redes neuronales recurrentes y *word embeddings* realizando una busqueda en *grid* para explorar distintos tamaños en la representación basada en *embeddings*, así como variando el número de unidades LSTM empleadas.

En la tabla 5.1 se puede observar como tanto el algoritmo basado en máquinas de vectores soporte como el algoritmo basado en regresión logística obtienen prestaciones muy similares.

**Tabla 5.1:** Resultados de la tarea de clasificación temática empleando características textuales y algoritmos clásicos sobre un conjunto de test formado por 2939 muestras

| Modelo | n_gram | min_df | max_df | kernel | C | *Accuracy* | F1 macro |
|---|---|---|---|---|---|---|---|
| **SVM** | (1,2) | 2 | 0.2 | linear | 100 | [0.97, 0.99] | 0.95 |
| **Logistic** | (1,2) | 2 | 0.2 | - | 1000 | [0.97, 0.99] | 0.94 |

Por otra parte, los resultados obtenidos con el uso de *word embeddings* y redes recurrentes han demostrado obtener unas prestaciones muy similares a las técnicas clásicas propuestas en la tabla 5.1.

### 5.3.2. Predicción usando imágenes

Para la predicción temática utilizando imágenes se ha hecho uso de modelos de aprendizaje profundo preentrenados sobre un gran conjunto de datos y adaptados utilizando las imágenes proporcionadas por el *dataset*. Los modelos que se han empleado son VGG-16 y Resnet-50, ambos preentrenados sobre ImageNET.

En la fase de *fine-tuning* de los modelos se ha empleado *data augmentation* es decir, se han realizado transformaciones sobre las imágenes para evitar de esta forma soluciones sobre-ajustadas. Más concretamente, el *data augmentation* que se ha empleado consiste en aplicar desplazamiento horizontal y vertical, zoom, rotación y *flip* horizontal sobre las imágenes. Además, se ha empleado *learning rate annealing*, esta técnica consiste en ir disminuyendo el factor de aprendizaje de la red con el fin de encontrar mejores generalizaciones, evitando de esta forma caer en mínimos locales.

En la tabla 5.2 se pueden apreciar los resultados obtenidos al realizar una evaluación mediante partición *hold-out* 80 % para entrenamiento y 20 % para test. Además, ambos sistemas han sido evaluados tras 50 *epochs*. Los mejores resultados se obtienen mediante la arquitectura VGG con 16 capas preentrenada sobre la base de datos ImageNET.

**Tabla 5.2:** Resultados obtenidos en la predicción del tema empleando redes convolucionales (50 *epochs*)

| Modelo | *Accuracy* | F1 macro |
|---|---|---|
| **VGG** | [0.96, 0.98] | 0.94 |
| **Resnet** | [0.87, 0.90] | 0.84 |

### 5.3.3. Predicción combinada

Para el modelo multimodal se ha optado por emplear una combinación de redes neuronales. La figura 5.1 muestra el esquema seguido para la predicción utilizando texto e imagen. Para ello se utilizan módulos basados en redes recurrentes y redes convolucionales basadas en VGG. Dichos módulos se concatenan para obtener un tensor único. Será el propio proceso de *back propagation* [5] el que se encargue de aprender cuál es la mejor combinación de las características de ambas modalidades.

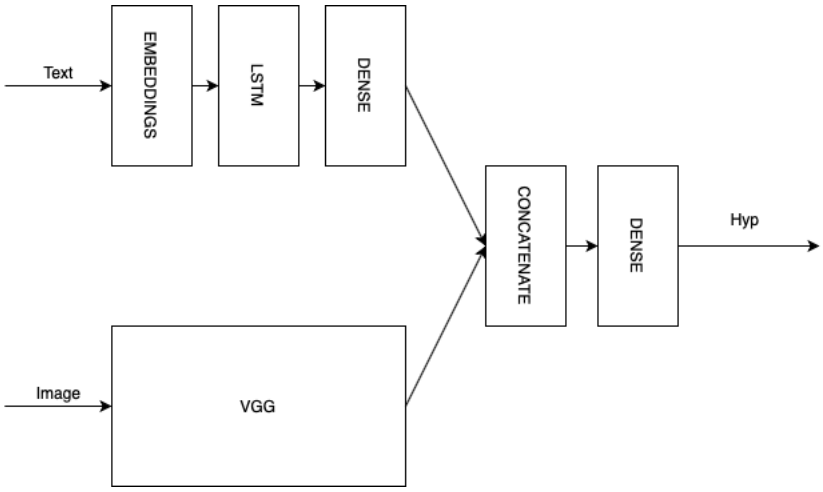


**Figura 5.1:** Esquema multimodal para la clasificación usando texto e imagen

Siguiendo el esquema comentado anteriormente y entrenando en una partición *hold-out* 80 % para entrenamiento y 20 % para test se ha obtenido cerca de un **99 %** de precisión con intervalo de confianza al 95 % de [98.4 %, 99.6 %] y una F1-macro asociada del **96 %**. Como podemos observar, aunque no es significativa, existe una pequeña mejoría en el sistema respecto al módulo textual y al módulo basado en imágenes.

## 5.4 Predicción del alcance

Como se ha comentado en el apartado 3.1.1, la biblioteca de anuncios de Facebook proporciona información acerca del alcance de los anuncios. Disponer de un sistema capaz de predecir cuál va a ser el alcance de un anuncio puede resultar de especial interés para los anunciantes, ya que habitualmente es una de las métricas que quieren maximizar. En los siguientes puntos se detallarán los distintos módulos desarrollados para finalizar con la construcción de un sistema que aproveche la información multimodal basada en texto e imágenes, así como demás información que podría proporcionar el usuario como podría ser el rango de edad, el sexo, la localización o el dinero que pretende gastar en la campaña publicitaria.

### 5.4.1. Predicción textual

A la hora de realizar la predicción del alcance empleando el texto se ha seguido una estrategía similar a la descrita en la predicción del tema, realizando un preproceso tokenizando los textos mediante nltk y evaluando tres modelos distintos. Además, se ha explorado la mejora introducida al utilizar *word embeddings* preentrenados sobre grandes volúmenes de datos, como puedan ser los proporcionados por la librería FastText [1] para el idioma español.

En la tabla 5.3 se puede observar como tanto el algoritmo basado en máquinas de vectores soporte como el algoritmo basado en regresión logística obtienen prestaciones muy similares, con unas prestaciones reducidas en términos de *accuracy* y F1-macro.

**Tabla 5.3:** Resultados de la tarea de clasificación del alcance empleando características textuales y algoritmos clásicos sobre un conjunto de test formado por 2939 muestras

| Modelo | n_gram | min_df | max_df | kernel | C | *Accuracy* | F1 macro |
|---|---|---|---|---|---|---|---|
| **SVM** | (1,2) | 1 | 0.3 | linear | 10 | [0.51, 5.55] | 0.33 |
| **Logistic** | (1,2) | 1 | 0.4 | - | 10000 | [0.51, 5.55] | 0.34 |

Por último, las pruebas realizadas con *word embeddings* y redes neuronales recurrentes, así como la aplicación de *word embeddings* pre-entrenados sobre un gran volumen de texto y proporcionados por la librería Fasttext, han demostrado obtener prestaciones muy similares a las obtenidas con los algoritmos y representaciones clásicas.

### 5.4.2. Predicción usando imágenes

A la hora de realizar la predicción utilizando las imágenes se ha empleado una estrategia similar a la empleada en la predicción temática. Concretamente se han utilizado los modelos VGG-16 y Resnet-50 comentados anteriormente y se han aplicado técnicas de *data augmentation*. En la tabla 5.4 se puede apreciar los resultados obtenidos en la predicción del alcance utilizando únicamente imágenes y empleando el 20 % de los datos para test.

**Tabla 5.4:** Resultados obtenidos en la predicción del alcance empleando redes convolucionales (50 *epochs*)

| Modelo | *Accuracy* | F1 macro |
|---|---|---|
| **VGG** | [0.52, 0.58] | 0.34 |
| **Resnet** | [0.47, 0.53] | 0.31 |

A la vista de los resultados expuestos se observa cómo el modelo que obtiene unas mejores prestaciones tras 50 *epochs* es la arquitectura VGG-16. Además, durante el entrenamiento, se ha observado una clara tendencia a obtener mejores generalizaciones al utilizar esta arquitectura.

### 5.4.3. Predicción combinada

En la construcción del modelo combinado (fig. 5.2) se ha empleado de nuevo la estructura propuesta anteriormente (fig. 5.1) y se ha añadido la información del coste de

[1] https://fasttext.cc

la campaña publicitaría, la segmentación referente a la localización y la segmentación referente al sexo y edad.

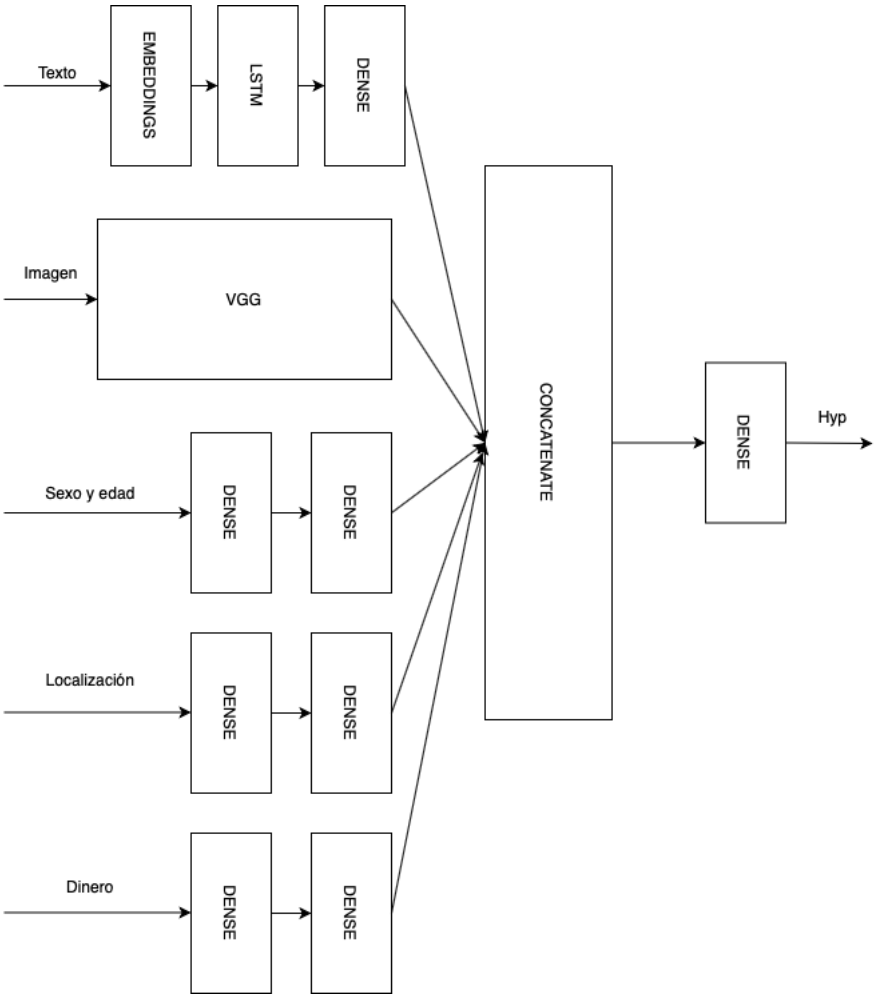


**Figura 5.2:** Esquema para la predicción combinada en la predicción del alcance de un anuncio

Para el entrenamiento de la arquitectura se han añadido capas *Dropout* [25] ya que han permitido obtener mejores generalizaciones evitando, a su vez, el sobre-entrenamiento. Además, se han descongelado los pesos del módulo basado en la arquitectura VGG con el objetivo de que sean adaptados a la tarea.

Una vez definida la arquitectura se ha realizado un entrenamiento a 75 *epochs* obteniendo una *accuracy* cercana al **65 %** (64.8 %) con un intervalo de confianza al 95 % de [61.9 %, 67.8 %].

## 5.5 Predicción de la segmentación por localización

A la hora de segmentar los anuncios, las plataformas permiten a los anunciantes determinar a qué comunidades autónomas quieren apuntar. Resulta por tanto interesante construir un sistema qué, a partir del contenido, prediga cuál va a ser el impacto sobre las distintas comunidades autónomas del estado español. Para ello, se ha propuesto un sistema basado en redes neuronales con una función de pérdida basada en la medida Kullback-Leibler presentada en la sección 5.2.5. De esta forma se calcula la divergencia entre las distribuciones de probabilidad con el fin de miminizar la distancia entre ellas.

El sistema propuesto se basa en el descrito en la figura 5.1. La capa de salida constará de 18 unidades, una por cada comunidad autónoma más una extra para cuando el impacto sea fuera de España. La capa utilizará una función de activación softmax [26]. De esta forma, cada neurona representará el porcentaje del impacto de una determinada campaña publicitaria en una comunidad autónoma. Además, se ha empleado el optimizador Adam [27], ya que ha demostrado obtener mejores generalizaciones con un número inferior de iteraciones.

Inicializando el sistema de forma aleatoria y comprobando la divergencia de Kullback-Leibler sobre el conjunto de test se obtiene un valor de **4.0989**. Tras realizar un entrenamiento con alrededor de 400 *epochs* sobre una partición *hold-out* 80 % para entrenamiento y 20 % para test, se consigue un *accuracy* en la clase mayoritaria del **55.0 %** (intervalo al 95 % [51.9 %, 58.1 %]) con una divergencia de Kullback-Leibler de **1.4087**.

## 5.6 Predicción de la segmentación por edad y sexo

A la hora de segmentar los anuncios, las plataformas permiten a los anunciantes determinar a qué grupos de edad y sexo quieren apuntar. Para predecir estas características se ha propuesto un sistema basado en redes neuronales, similar al propuesto para la predicción de la segmentación por localización. El objetivo de este sistema consiste en determinar para cada segmento de edad y cada segmento sexual cuál va a ser el impacto de la campaña publicitaria.

Para ello, de nuevo, se ha seguido una estrategia similar a la descrita en la sección anterior. La capa de salida de la red consistirá en un vector de 13 componentes formado por las combinaciones posibles entre los 6 rangos posibles de edad y los dos posibles sexos, y una componente extra para cuando no se conoce el segmento de sexo y edad. Además, se ha seguido utilizando el optimizador Adam y la función de perdida de Kullback-Leibler.

Inicializando el sistema de forma aleatoria y comprobando la divergencia de Kullback-Leibler sobre el conjunto de test se obtiene un valor de **5.1000**. Tras realizar un entrenamiento con alrededor de 400 *epochs* sobre una partición *hold-out* 80 % para entrenamiento y 20 % para test, se consigue un *accuracy* en la clase mayoritaria del **41.9 %** (intervalo al 95 % [38.9 %, 45.0 %]) con una divergencia de Kullback-Leibler de **0.5886**.

# CAPÍTULO 6
# Conclusiones

En la actualidad, la industria de la publicidad y en especial la publicidad *online* es una de las más rentables. Esto hace que grandes empresas como Google o Facebook basen en gran medida sus modelos de negocio en la venta de anuncios para sus plataformas.

La existencia de herramientas de *targeting* capaces de determinar los segmentos a los que debería de ir dirigido un determinado anuncio a partir de su contenido puede ser de especial interés para anunciantes, ya que permitiría optimizar campañas de publicidad, permitiendo ahorrar costes y apuntar a segmentos más adecuados. Una vez determinado el objetivo principal del trabajo se definen 4 tareas distintas a trabajar sobre un *dataset* preparado por el grupo de investigación Mediaflows de la Universitat de València y relacionado con anuncios políticos de las elecciones del 28 de abril de 2019 y las elecciones del 10 de noviembre de 2019 en España.

Para la realización de los experimentos, y dada la multimodalidad del *dataset*, se ha propuesto sistemas multimodales capaces de realizar las predicciones a partir de las imágenes y el texto de los anunciantes. Además, se ha tratado de enriquecer los sistemas aportando la segmentación propuesta por los anunciantes y referente a la localización, sexo, edad y gasto económico.

Los resultados obtenidos resultan competitivos en la predicción temática y aceptables en la predicción del alcance. En cambio, las prestaciones son reducidas en cuanto a la predicción de los segmentos de localización y de edad-sexo.

Resumiendo, en este trabajo se han explorado distintas técnicas de representación de los datos, como es el caso de las bolsas de palabras o los *word embeddings* para informacion textual. Se han explorado distintos algoritmos de aprendizaje clásicos como las máquinas de vectores soporte y la regresión logística. Se ha experimentado con distintas técnicas de *deep learning* como las redes recurrentes o las redes convolucionales. Por último, se ha propuesto una arquitectura multimodal capaz de mezclar la información textual y las imágenes para realizar las predicciones. Concluir, por tanto, que se han cumplido los objetivos propuestos al inicio del trabajo, por lo que el balance general del proyecto es positivo.

Además, con el fin de poder replicar los experimentos realizados, se ha dejado en Github[1] el código desarrollado.

Por último, a nivel profesional este trabajo ha permitido obtener conocimientos más sólidos tanto teóricos como en el uso de *toolkits* para el desarrollo de modelos predictivos. Estos conocimientos adquiridos son de especial interés en el tejido productivo donde, por ejemplo, librerías como Tensorflow o scikit learn son ampliamente utilizadas por empre-

[1] https://github.com/marescas/TFM

sas como Google, Facebook o Amazon para el análisis y la extracción de información sobre grandes volúmenes de datos.

# CAPÍTULO 7
# Trabajo futuro

Ante el amplio alcance de este proyecto se ha decidido centrarse en la predicción del tema tratado, así como el alcance desde una perspectiva multimodal (texto e imagen). Además, en última instancia se ha intentado aproximar las distribuciones del impacto de los anuncios en distintos segmentos como son la edad, el sexo o la localización. Existen numerosas mejoras a este sistema, así como otras áreas que pueden ser investigadas. A continuación se detallan algunas de ellas.

En primer lugar, al igual que se han empleado modelos preentrenados para la clasificación de imágenes, se puede hacer uso de técnicas basadas en la adaptación de modelos de lenguaje preentrenados utilizando técnicas de Transformers como BERT. Concretamente, se puede hacer uso de un modelo BERT preentrenado sobre un corpus de más de 4GB de texto en español. Este modelo recibe el nombre de BETO [28].

Por otra parte, se podría realizar un análisis de los vídeos complementando de esta forma a los modelos basados en la predicción utilizando imágenes. Por ejemplo, se podría hacer uso de la técnica descrita en [29], obteniendo de esta forma *embeddings* de los vídeos, los cuales pueden ser utilizados posteriormente para las tareas de clasificación propuestas.

Como se ha comentado, la utilización de *word embeddings* preentrenados sobre un gran volumen de datos no ha permitido mejorar las prestaciones de los modelos. Esto puede deberse a que los *embeddings* son demasiado genéricos y no consiguen captar correctamente la relaciones semánticas y sintácticas presentes en la comunicación política. Una posible mejora podría pasar por compilar un gran *corpus* de comunicación política y entrenar unos *embeddings* para, de esta forma, intentar mejorar las prestaciones de los sistemas.

Por último, dado que los resultados obtenidos en la predicción de la segmentación sexo-edad y localización obtuvieron bajas prestaciones, se podría explorar nuevas arquitecturas de *deep learning* con el fin de obtener modelos más robustos.

# CAPÍTULO 8
# Relación con los estudios cursados

Para la realización de este trabajo han sido necesarios gran parte de los conocimientos impartidos en el máster, especialmente de las ramas relacionadas con el reconocimiento de formas y la lingüística computacional.

Con respecto a la rama de reconocimiento de formas, la asignatura Reconocimiento de Formas y Aprendizaje Computacional (RFA) ha permitido establecer las bases teóricas necesarias para entender los elementos que subyacen a un sistema de reconocimiento de formas. La asignatura de Redes Neuronales Artificiales (RNA) ha aportado los conocimientos teóricos y prácticos necesarios para desarrollar aplicaciones de redes neuronales para la clasificación y regresión en las distintas tareas desarrolladas en este trabajo. Por último, la asignatura Aplicaciones de Reconocimiento de Formas (ARF) ha contribuido en el aprendizaje, desde una perspectiva totalmente práctica, de sistemas de reconocimiento de formas.

Por otra parte, con respecto a la rama de lingüística computacional, la asignatura Lingüística Computacional (LC) ha permitido establecer las bases teóricas necesarias para el entendimiento de las aplicaciones relacionadas con la lingüística. Además, la asignatura Aplicaciones de la Lingüística Computacional (ALC) ha permitido abordar desde una perspectiva práctica problemas de clasificación de texto.

Destacar también la asignatura Visión Por Computador (VPC). Esta asignatura ha sido de especial importancia, ya que ha permitido aprender conceptos y arquitecturas del actual estado del arte para tareas de clasificación con imágenes.

Por último, con el desarrollo de este trabajo se han trabajado también numerosas competencias transversales como podrían ser ”aprendizaje permanente”, ”diseño y proyecto”, ”planificación y gestión del tiempo” o ”comunicación efectiva”.

# Bibliografía


[1] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[3] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[12] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?," in *China National Conference on Chinese Computational Linguistics*, pp. 194–206, Springer, 2019.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[15] J. H. Orallo, M. J. R. Quintana, and C. F. Ramírez, *Introducción a la Minería de Datos*. Pearson Educación, 2004.

[16] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[17] T. Takahashi, T. Tahara, K. Nagatani, Y. Miura, T. Taniguchi, and T. Ohkuma, "Text and image synergy with feature cross technique for gender identification," *Working Notes Papers of the CLEF*, 2018.

[18] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[19] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[23] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[24] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] J. Cañete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish pre-trained bert model and evaluation data," in *to appear in PML4DC at ICLR 2020*, 2020.

[29] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei, "Learning temporal embeddings for complex video analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4471–4479, 2015.