Analítica de datos con Python para marketing digital

Desarrollo de KPIs en negocios *online*

Joseba Ruiz Díez
Ubaldo Hervás Olvera

ANAYA MULTIMEDIA

Índice de contenidos

Operadores binarios y comparaciones

Algo muy relacionado con los *booleans* son los operadores. Además del uso de == o de !=, es importante conocer muchos más que llegarás a ver a lo largo del libro, que aparecen en la tabla 2.2.

Tabla 2.2. Tabla con operaciones más simples y que se pueden aplicar tanto en *integers*, *floats* e incluso *strings*, aunque se debe tener en cuenta el tipo de *output* que se va a generar en función del tipo de dato y el tipo de operación que se va a ejecutar.

OPERACIÓN	DESCRIPCIÓN	EJEMPLO; RESULTADO
`a - b`	Resta a y b	`5 - 3; 2`
`a * b`	Multiplica a y b	`5 * 3; 15`
`a/b`	Divide a y b	`5/3; 1,67`
`a//b`	Divide a y b y solo se queda con el resultado del número entero	`5/3; 1`
`a ** b`	Eleva a a la potencia de b	`5 ** 3; 125`
`a & b`	Para *booleans*, devuelve `True` si a y b son `True`	`True & False; False`
`a | b`	Para *booleans*, por defecto, devuelve `True` si a o b son `True`	`True | False; True`
`a % b`	Para *floats* o *integers*, devuelve el resto de dividir a por b	`10 % 2; 0`
`a <= b`	Devuelve `True` si a es menor o igual a b	`a (5) <= b (6); True`
`a >= b`	Devuelve `True` si a es mayor o igual a b	`a (5) >= b (6); False`
`a is b`	Devuelve `True` si a y b hacen referencia al mismo objeto de Python, que no específicamente al valor	`a is b; False`
`a is not b`	Devuelve `True` si a y b hacen referencia al mismo objeto de Python, que no específicamente al valor	`a is b; True`

Para entender este último punto de `is` e `is not`, suele ser recomendable poner un ejemplo (aunque más adelante, en el libro, se explicará qué son los objetos en Python):

```
In [15]: a = [1, 2, 3, 4, 5] #esto hace referencia a un objeto de tipo lista
que está dentro de la variable 'a'
b = a #aquí asignamos la variable 'a' a una segunda variable llamada 'b'
a is b
Out[15]: True
```


Como se puede ver, devuelve `True`, lo mismo que devolvería si usáramos el símbolo ==; sin embargo, si ejecutamos el siguiente código:

```
In [16]: c = list(a) #ejecutamos la función list() que, por su naturaleza,
siempre crea una lista nueva de Python. Cambia el objeto sin cambiar los
valores que hay dentro.
a is not c Out
Out[16]: True
```

Aún albergando los mismos valores, si se ejecuta el código anterior que estipula que a y c no son lo mismo, devuelve `True`. De hecho, si a continuación ejecutamos el siguiente código, se podrá ver que si se aplica ==, también devuelve `True`, puesto que el símbolo de == hace referencia al valor y los operadores `is` e `is not` hacen referencia a la igualdad del objeto:

```
In [17]: c == a
Out[17]: True
```

Listas

Además de *strings*, *booleans*, *floats* y demás, también existen otros tipos de datos fundamentales, como, por ejemplo, las listas (que ya se han mostrado previamente en la línea `In [16]`). Las listas son un tipo de objeto en sí mismo que puede albergar todo tipo de objetos como, por ejemplo, otros *strings*, *integers*, *floats*, *booleans* e, incluso, otras listas. Las listas, como se ve a continuación, se declaran con el símbolo [ ]:

```
In [18]: product_revenue = [15000, 15832.90, 22830, 11098, 9870.12]
product_revenue
Out[18]:  [15000, 15832.90, 22830, 11098, 9870.12]
```

En esta lista, tenemos tanto *integers* como *floats* y hay que tener presente que la variable `product_revenue` es una variable de tipo `list`, la cual, como el resto de objetos, tiene sus propias normas, en las cuales se profundizará en las siguientes páginas cuando se ahonde en los siguientes tipos de datos.

Tuplas

```
In [19]: service_revenue = (9000, 2000, 12000, 989.90, 1501)
service_revenue
type(service_revenue)
Out[19]:  tuple
```

Como se puede ver en el código anterior, las tuplas son objetos que también pueden albergar objetos de distinta índole. Aparte de que el símbolo para declarar una tupla es el paréntesis, ( ), y no el corchete, [ ], como sucede con las listas, ¿qué diferencia puede llegar a haber entre ambas para que haya la necesidad de que coexistan?

CARACTERÍSTICA	MYSQL WORKBENCH	GOOGLE BIGQUERY
Coste	Gratis para descargar y usar.	Modelo de precios basado en el consumo (almacenamiento y procesamiento de consultas).

¿Qué es una operación CRUD? En esencia, se trata de las siglas de las cuatro operaciones básicas utilizadas en bases de datos. La sencillez de estas cuatro opciones denotan la simplicidad inherente a SQL en contraposición a todo lo que Google BigQuery puede desarrollar. A continuación, las siglas y la definición de CRUD:

- *Create*: Crear o añadir registros a la base de datos.
- *Read*: Leer, consultar o recuperar registros presentes en la base de datos.
- *Update*: Actualizar datos dentro de la base de datos.
- *Delete*: Eliminar registros dentro de la base de datos.

La misión de este libro es simple: hacer que el mundo de marketing digital y la ciencia de datos estén más unidas y, debido a que el entorno Google es el más utilizado en el mundo del marketing, se utilizará para los siguientes casos la interfaz de Google BigQuery, aunque eso no debe dar a entender al lector de que el resto de soluciones no sean funcionales y perfectamente válidas para el negocio digital.

Ejecutando las primeras *queries*

La curva de aprendizaje de SQL es muchísimo más simple que en Python o cualquier otro lenguaje (de hecho, el lector debe recordar que SQL no es un lenguaje de programación como sí es Python, JavaScript o Java) y es por ello que es uno de los más cómodos de aprender puesto que, con un poco de práctica, ya se pueden generar buenas consultas que bien pueden responder a preguntas básicas de negocio.

En el contexto de realizar consultas sobre una base de datos de ventas de vehículos de ocasión la cuestión es muy simple; sin embargo, cuando se trabajan datos que provienen de Google Analytics 4, el asunto se complica. ¿Por qué? El motivo es que para responder a preguntas de negocio con Google Analytics 4 y Google BigQuery debe haber un conocimiento profundo sobre la naturaleza del dato (¿qué es una sesión?, ¿qué es una sesión con interacción?, ¿qué diferencia existe entre las dimensiones de ámbito usuario y sesión?, ¿qué significa `user_pseudo_id`?) y, sobre todo, para interpretar el resultado de las consultas.

Sobre este asunto se profundizará un poco más en el próximo capítulo sobre marketing digital, un capítulo que se recomienda leer a aquellos perfiles más técnicos, puesto que uno de los grandes errores de los científicos de datos que aterrizan en proyectos de marketing digital es su desconocimiento sobre el ámbito web y *app*.

Google BigQuery y Google Trends Sandbox

La configuración de Google BigQuery no es, inicialmente, algo excesivamente complejo, más aún si solamente se quiere practicar con SQL; sin embargo, si es la primera vez y se va a trabajar en un proyecto profesional, se recomienda trabajar codo con codo con un desarrollador al lado en la configuración inicial. No obstante, a continuación, se va a mostrar los primeros pasos y cómo acceder a la zona de pruebas de Google Trends para poder practicar.

El primer paso sería visitar `https://console.cloud.google.com/bigquery` desde el navegador de tu ordenador. Si es tu primera vez y no tienes una cuenta de Google conectada a Google BigQuery, lo primero que aparecerá será algo como lo de la figura 4.1.

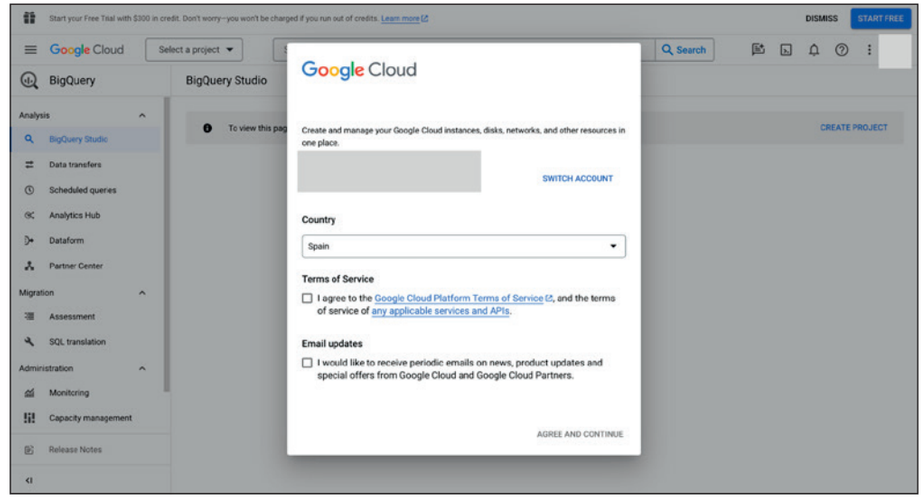

Figura 4.1. Interfaz de Google BigQuery.

Después de seleccionar el país y aceptar los términos de condiciones, se deberá hacer clic en Create Project y aparecerá la siguiente pantalla, en la que sencillamente debes seleccionar el nombre del proyecto, el nombre de la organización y una localización. Después, debes hacer clic en Create, como se ve en la figura 4.2.

- **Empírica:** A distinción de la anterior, esta regla "solo" puede ser usada bajo una distribución normal. Determina básicamente que: el 68 % de los datos se encuentran a una desviación estándar de la media ($\mu \pm \sigma$), el 95 % a dos ($\mu \pm 2\sigma$) y el 99,7 % a tres ($\mu \pm 3\sigma$).

Como se puede observar, no ha sido necesario indicar las fórmulas de cada distribución. Lo que nos interesa mayormente es conocer qué deberemos aplicar en cada caso y aprender a detectar rápidamente cada una de ellas cuando analicemos las variables o las características que estudiaremos en nuestros análisis en posteriores capítulos.

Antes de finalizar este punto, es necesario detallar qué es la estandarización de una distribución. Este término hace referencia a la posibilidad de convertir diferentes distribuciones a una misma escala, de tal forma que se puedan incluso comparar entre ellas. A grandes rasgos, es aplicable únicamente a distribuciones que se aproximan a una distribución normal. No todas pueden transformarse y mantener las mismas propiedades intactas.

En el caso de la distribución normal, cuando se realiza su estandarización pasa a denominarse como distribución normal estándar ($\mu = 0, \sigma = 1$). Esto sucede cuando se aplica la siguiente fórmula: $z = \dfrac{x - \mu}{\sigma}$

A este resultado se le conoce como "valor z" (*z-score*) o "valor tipificado". Para interpretar este valor desde un punto de vista probabilístico, tenemos a nuestra disposición una tabla tipo que muestra todos los resultados posibles, en la figura 5.3.

z	+0,00	+0,01	+0,02	+0,03	+0,04	+0,05	+0,06	+0,07	+0,08	+0,09
0,0	0,00000	0,00399	0,00798	0,01197	0,01595	0,01994	0,02392	0,02790	0,03188	0,03586
0,1	0,03983	0,04380	0,04776	0,05172	0,05567	0,05962	0,06356	0,06749	0,07142	0,07535
0,2	0,07926	0,08317	0,08706	0,09095	0,09483	0,09871	0,10257	0,10642	0,11026	0,11409
0,3	0,11791	0,12172	0,12552	0,12930	0,13307	0,13683	0,14058	0,14431	0,14803	0,15173
0,4	0,15542	0,15910	0,16276	0,16640	0,17003	0,17364	0,17724	0,18082	0,18439	0,18793
0,5	0,19146	0,19497	0,19847	0,20194	0,20540	0,20884	0,21226	0,21566	0,21904	0,22240
0,6	0,22575	0,22907	0,23237	0,23565	0,23891	0,24215	0,24537	0,24857	0,25175	0,25490
0,7	0,25804	0,26115	0,26424	0,26730	0,27035	0,27337	0,27637	0,27935	0,28230	0,28524
0,8	0,28814	0,29103	0,29389	0,29673	0,29955	0,30234	0,30511	0,30785	0,31057	0,31327
0,9	0,31594	0,31859	0,32121	0,32381	0,32639	0,32894	0,33147	0,33398	0,33646	0,33891
1,0	0,34134	0,34375	0,34614	0,34849	0,35083	0,35314	0,35543	0,35769	0,35993	0,36214
1,1	0,36433	0,36650	0,36864	0,37076	0,37286	0,37493	0,37698	0,37900	0,38100	0,38298
1,2	0,38493	0,38686	0,38877	0,39065	0,39251	0,39435	0,39617	0,39796	0,39973	0,40147
1,3	0,40320	0,40490	0,40658	0,40824	0,40988	0,41149	0,41308	0,41466	0,41621	0,41774
1,4	0,41924	0,42073	0,42220	0,42364	0,42507	0,42647	0,42785	0,42922	0,43056	0,43189
1,5	0,43319	0,43448	0,43574	0,43699	0,43822	0,43943	0,44062	0,44179	0,44295	0,44408
1,6	0,44520	0,44630	0,44738	0,44845	0,44950	0,45053	0,45154	0,45254	0,45352	0,45449
1,7	0,45543	0,45637	0,45728	0,45818	0,45907	0,45994	0,46080	0,46164	0,46246	0,46327
1,8	0,46407	0,46485	0,46562	0,46638	0,46712	0,46784	0,46856	0,46926	0,46995	0,47062
1,9	0,47128	0,47193	0,47257	0,47320	0,47381	0,47441	0,47500	0,47558	0,47615	0,47670
2,0	0,47725	0,47778	0,47831	0,47882	0,47932	0,47982	0,48030	0,48077	0,48124	0,48169

z	+0,00	+0,01	+0,02	+0,03	+0,04	+0,05	+0,06	+0,07	+0,08	+0,09
2,1	0,48214	0,48257	0,48300	0,48341	0,48382	0,48422	0,48461	0,48500	0,48537	0,48574
2,2	0,48610	0,48645	0,48679	0,48713	0,48745	0,48778	0,48809	0,48840	0,48870	0,48899
2,3	0,48928	0,48956	0,48983	0,49010	0,49036	0,49061	0,49086	0,49111	0,49134	0,49158
2,4	0,49180	0,49202	0,49224	0,49245	0,49266	0,49286	0,49305	0,49324	0,49343	0,49361
2,5	0,49379	0,49396	0,49413	0,49430	0,49446	0,49461	0,49477	0,49492	0,49506	0,49520
2,6	0,49534	0,49547	0,49560	0,49573	0,49585	0,49598	0,49609	0,49621	0,49632	0,49643
2,7	0,49653	0,49664	0,49674	0,49683	0,49693	0,49702	0,49711	0,49720	0,49728	0,49736
2,8	0,49744	0,49752	0,49760	0,49767	0,49774	0,49781	0,49788	0,49795	0,49801	0,49807
2,9	0,49813	0,49819	0,49825	0,49831	0,49836	0,49841	0,49846	0,49851	0,49856	0,49861
3,0	0,49865	0,49869	0,49874	0,49878	0,49882	0,49886	0,49889	0,49893	0,49896	0,49900
3,1	0,49903	0,49906	0,49910	0,49913	0,49916	0,49918	0,49921	0,49924	0,49926	0,49929
3,2	0,49931	0,49934	0,49936	0,49938	0,49940	0,49942	0,49944	0,49946	0,49948	0,49950
3,3	0,49952	0,49953	0,49955	0,49957	0,49958	0,49960	0,49961	0,49962	0,49964	0,49965
3,4	0,49966	0,49968	0,49969	0,49970	0,49971	0,49972	0,49973	0,49974	0,49975	0,49976
3,5	0,49977	0,49978	0,49978	0,49979	0,49980	0,49981	0,49981	0,49982	0,49983	0,49983
3,6	0,49984	0,49985	0,49985	0,49986	0,49986	0,49987	0,49987	0,49988	0,49988	0,49989
3,7	0,49989	0,49990	0,49990	0,49990	0,49991	0,49991	0,49992	0,49992	0,49992	0,49992
3,8	0,49993	0,49993	0,49993	0,49994	0,49994	0,49994	0,49994	0,49995	0,49995	0,49995
3,9	0,49995	0,49995	0,49996	0,49996	0,49996	0,49996	0,49996	0,49996	0,49997	0,49997
4,0	0,49997	0,49997	0,49997	0,49997	0,49997	0,49997	0,49998	0,49998	0,49998	0,49998

Figura 5.3. Tabla normal estándar.

Intervalos de confianza y contrastes de hipótesis

Como ya se ha comentado previamente, la estadística inferencial se basa esencialmente en el estudio de una muestra a partir de una población. Este muestreo es muchas veces necesario, ya que de otra forma sería imposible realizar el análisis de todos y cada uno de los individuos, entre otras razones por el tiempo o el coste de llevarlo a cabo en su totalidad. Como es lógico pensar, este muestreo deberá ser representativo de dicha población. Para ello, existen diferentes métodos de muestreo:

- **Aleatorio simple:** Cada elemento de la población tiene exactamente la misma probabilidad de ser elegido. Se hace una extracción al azar.
- **Estratificado:** La población se divide en subgrupos con características similares, llamados estratos, y a continuación se recoge una muestra aleatoria simple de cada uno de ellos. Esto permite que cada estrato esté representado al final del proceso.
- **Por conglomerados:** Similar al estratificado, pero en este caso se divide la población en grupos más amplios según la clase a la que pertenezcan, por ejemplo: país, región, etc.
- **Sistemático:** Se elige un punto de inicio aleatoriamente a partir del cual se seleccionan el resto de elementos a intervalos con una distancia k predeterminada.

Aunque de manera menos ortodoxa, también juega un papel fundamental el criterio del analista, decidiendo en base a su experiencia cómo debería ser la extracción de la muestra. Esto deberá llevarse a cabo en casos muy concretos y solo cuando la experiencia del negocio adquirida lo permita, ya que de cualquier otra forma se podría generar un sesgo evidente.

Además, hay que tener en cuenta que los resultados pueden verse alterados por el tipo de método de muestreo aplicado. Es decir, siempre existirá una discrepancia mayor o menor entre la muestra y la población. Esta diferencia se conoce como error de muestreo. Como regla generalizada, cuanto mayor sea el tamaño de la muestra, menor será este error. Para poder estimarlo con cierta precisión, lo primero que debemos entender es el significado del siguiente término.

La distribución muestral de la media, como su propio nombre indica, es la distribución de todas aquellas medias obtenidas del muestreo iterativo de una población. Se caracteriza porque su media es igual a la de la población, tiene una menor dispersión que la población y se aproxima a la distribución normal.

Precisamente, es esta última propiedad la que da lugar al "teorema central del límite" (TCL), el cual dictamina que la media muestral de un número relativamente grande de variables aleatorias independientes e idénticamente distribuidas (conocidas también como variables `iid`) se aproxima a una distribución normal. Es decir que, indistintamente de la forma de la distribución original y casi siempre con muestras relativamente amplias, la distribución muestral de la media se asemejará a la distribución normal. Este teorema es fundamental para llevar a cabo numerosas implementaciones, entre las que destacan: calcular intervalos de confianza, realizar estimaciones o incluso llevar a cabo contrastes de hipótesis.

Por lo tanto, volviendo sobre el cálculo del error entre la distribución muestral de la media y la distribución de la población, se puede definir esta discrepancia como la desviación estándar de todas las posibles muestras. Lo que se denomina como error estándar de la media y se describe tal que:

$z = \frac{\sigma}{\sqrt{n}}$, siendo n el tamaño de la muestra (número de observaciones).

Dicho esto, la realidad es algo más compleja. En muchas de las situaciones reales no se conocen algunos de los valores atribuidos a los parámetros de la población como puede ser la media, la desviación estándar o el tipo de distribución de los datos. En estos casos es preciso entender los siguientes dos conceptos:

- **Estimador puntual:** Valor usado para estimar el parámetro poblacional. Por ejemplo, la media muestral ($\bar{x}$), la desviación estándar muestral (s) o la proporción muestral (ρ). Este último estimador se refiere a la proporción que abarca una característica concreta dentro de una muestra. Por ejemplo, si en una muestra de 500 personas, 75 son compradores (clientes) de un *e-commerce*, se estima que su proporción es

  $$\rho = \frac{x}{n} = \frac{75}{500} = 0,15 = 15\%$$

  Este resultado servirá como estimador puntual de la media poblacional desconocida.
- **Intervalo de confianza:** Rango de valores dentro del cual se estima encontrar con cierta fiabilidad el parámetro desconocido. Tanto el valor mínimo como el máximo del rango se calculan en base a la fórmula:

  $$\bar{x} \pm MDE = \bar{x} \pm z\frac{\sigma}{\sqrt{n}}$$

  siendo *MDE* el margen de error. Este margen hace referencia al producto de un valor *z* y del error estándar previamente explicado, y el cual variará según se conozca o no la desviación estándar de la población.

- **Desviación estándar conocida:** El margen de error en este caso se estipula como: $z\frac{\sigma}{\sqrt{n}}$

  siendo z el valor tipificado (*z-score*). Por lo que, utilizando la tabla de valores estándar normales anteriormente mostrada, el intervalo de confianza sería fácilmente calculable.
- **Desviación estándar desconocida:** En este contexto, el margen de error se definirá a partir de la distribución t, en lugar de z, declarándose tal que:

  $$t\frac{s}{\sqrt{n}}$$

La distribución t de Student es una distribución de probabilidad continua que se usa esencialmente cuando el tamaño de la muestra es pequeño. De hecho, se establece habitualmente con muestras $n < 30$ y su forma suele ser más aplanada por el centro y más extendida por los lados que la distribución normal estándar. Precisamente por esta razón la dispersión será mayor que en z, y por lo tanto abarcará un rango algo más amplio. Este estadístico:

$$t = \frac{(\bar{x} - \mu)}{s/\sqrt{n}}$$

fue introducido por William Seally Gosset como parte de un test que permitiera monitorizar la calidad de la cerveza Guinness, empresa en la que trabajaba en Dublín.

Otro concepto vinculado de alguna forma a la distribución t y a otras de características similares son los grados de libertad (*df*). Estos hacen referencia al número de valores diferentes que pueden llegar a variar dentro de una muestra y cuyos principales objetivos son estimar los valores críticos o establecer cómo se aproximan dichas distribuciones a la normal. Dicho de otro modo, muestran la variabilidad es un contexto concreto.

Desde otro punto de vista, sumado a los diferentes métodos de muestreo y a los posibles errores asociados, la correcta elección del tamaño de una muestra es una parte indispensable más a tener en cuenta. Depende básicamente de los siguientes factores:

- **Margen de error tolerable:** Cantidad de error aceptado. Un margen más pequeño conllevará una mayor muestra.
- **Dispersión de la población:** A más variabilidad en los datos, se requerirá una muestra más grande.
- **Nivel de confianza:** Baremo que cuantifica cuál será la exactitud del resultado. Se suele usar el valor 95 %, aunque también es habitual ver un 90 % o incluso un 99 % dependiendo del ámbito en el que se esté aplicando. Lo que de ninguna forma tendrá sentido será establecerlo en un 100 %, ya

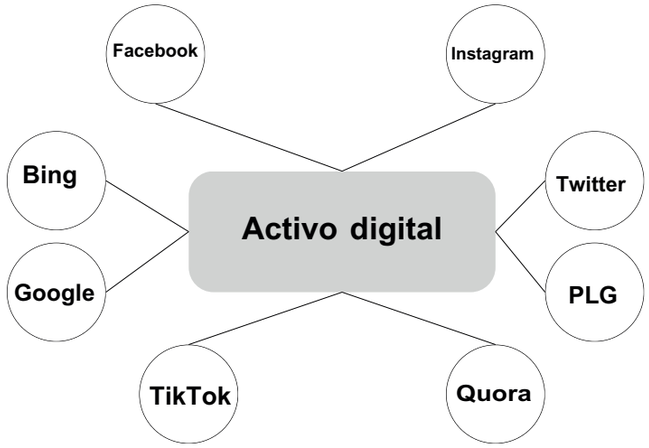

Figura 6.1. En esta gráfica se muestra cómo diferentes canales alimentan de tráfico web un único activo digital que bien puede ser un sitio web o una aplicación móvil. Hay muchas más fuentes de tráfico que las que se muestran en esta gráfica y no se realiza una distinción por medio para advertir si el tráfico es de pago u orgánico.

Todo esto con un único activo digital, por ejemplo, un sitio web, pero esto puede convertirse en algo mucho más complejo cuantos más activos digitales se incorporan al negocio digital. Un negocio que combina dos sitios web diferentes con una aplicación móvil y que, además, conectan entre sí, genera un entorno mucho más complejo, como se muestra en la figura 6.2.

Facebook, Instagram o TikTok (entre otros) pueden generar un tráfico determinado a tu activo digital en función de las acciones que generes en ese canal, lo cual conecta con el hecho de si es más rentable crear contenido o campañas en un canal u otro. Esa decisión solo se puede tomar con datos y para ello debe realizarse un plan de medición.

Elementos básicos de *cookies* y privacidad

Además de la complejidad operativa que se puede deducir procedente de los canales y de su correcta medición, también se debe tener en cuenta el aspecto legal, especialmente si se van a capturar datos de personas en Europa debido a la legislación vigente relacionada con las *cookies* y el seguimiento de los usuarios en los activos digitales.

En esencia, el consentimiento de *cookies* que da el usuario debe ser informado, explícito y haberse obtenido a través de una acción afirmativa inequívoca. Esto pone sobre la mesa algunas cuestiones como el hecho de que el botón de aceptación

de *cookies* no puede tener "continuar navegando", sino que debería poner algo similar a "Aceptar *cookies*", como se ve en la figura 6.3, por no mencionar todo lo relacionado con la distinción entre *cookies* de analítica (que permiten la medición a través de Google Analytics 4, por ejemplo) y *cookies* de marketing (que permitirían, por ejemplo, enviar información a Google para optimizar campañas desde sus plataformas de publicidad digital).

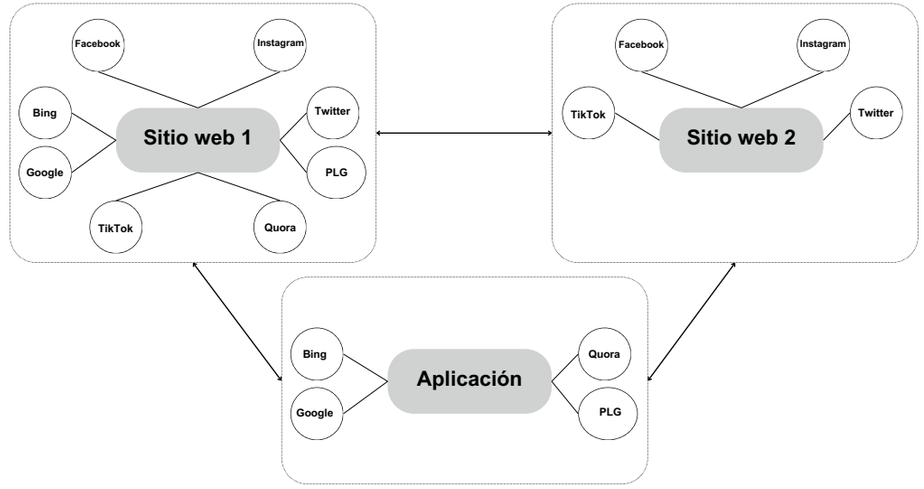

Figura 6.2. En esta gráfica se visualiza cómo tres activos digitales diferentes pueden recibir tráfico de distintas fuentes y pueden alimentar a otros activos digitales. Todo esto conlleva diferentes tipos de medición y consideraciones en materia de privacidad. El lector debe ver que cada activo digital es alimentado por diferentes canales porque cada activo tiene su objetivo específico.

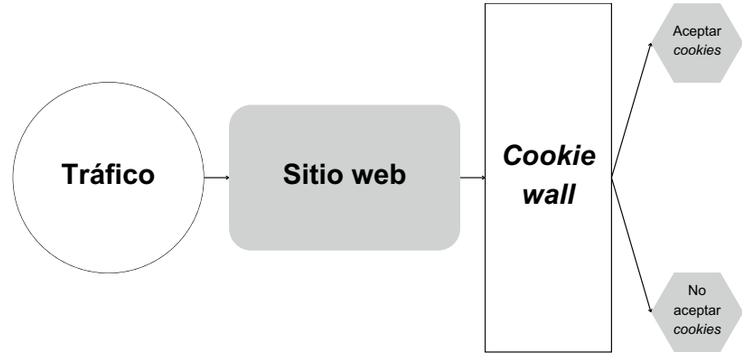

Figura 6.3. Ejemplo de consentimiento de *cookies*.

simular procesos como la interpretación del lenguaje, la toma de decisiones o la comprensión visual entre otros muchos. Esto hace que el concepto de IA se refiera a un campo de gran amplitud donde se conjugan disciplinas como las matemáticas, la estadística, el álgebra, la programación, la ingeniería o incluso el conocimiento del negocio. Esto de por sí implica ya muchas cuestiones, por ejemplo, el tipo de perfiles involucrados. Con el auge de aplicaciones como ChatGPT o Bard, muchas empresas han dado un paso de gigante con respecto al análisis de sus datos que, aunque *a priori* parezca sencillo y fácilmente gestionable, conlleva una evidente ausencia de experiencia y conocimientos previos.

De hecho, muchas personas vinculan la IA directamente con este tipo de herramientas. En verdad, la realidad es muy diferente. La inteligencia artificial es un concepto que engloba a varias disciplinas, tal y como se puede ver en la figura 8.1.

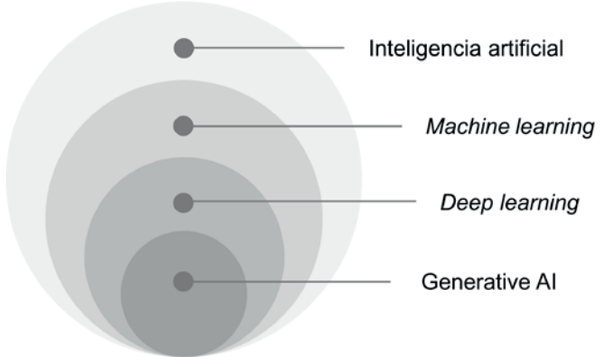

Figura 8.1. Ámbitos de la inteligencia artificial.

Seguramente a muchas de ellas esto les suponga una mejora en su trabajo diario; sin embargo, sin darse apenas cuenta, estarán desestimando muchas de las cuestiones más importantes para poder llevar a cabo análisis en detalle. Al fin y al cabo, estas novedosas herramientas son el culmen de lo que a continuación a lo largo de este capítulo vamos a explicar en detalle.

Machine learning, aprendizaje automático

También denominado como ML por sus siglas en inglés, el aprendizaje automático o *machine learning* es un subcampo de la inteligencia artificial que permite extraer conocimiento de nuestros datos utilizando para ello modelos estadísticos cuyo principal objetivo es encontrar determinado tipo de patrones. Es decir, se trata de promover el estudio autónomo de los datos a partir de un desarrollo humano previo.

Aunque no están claros sus orígenes exactos, este término comienza a popularizarse a finales de la década de los cincuenta, cuando Arthur Samuel desarrolla el primer programa de ordenador capaz de aprender. Este software, además de permitir jugar al *checkers* (juego similar a las damas), era capaz de mejorar a medida que se avanzaban niveles. Pero quizás la referencia más famosa sobre ML es la que trajo consigo Tom Mitchell declarando: "Se dice que un programa de computadora aprende a partir de la experiencia E con respecto a alguna clase de tareas T y alguna medida de rendimiento P, si su rendimiento en tareas de T, medido por P, mejora la experiencia E".

Esta definición es clave para entender el funcionamiento y el flujo básico de trabajo dentro del aprendizaje automático:

1. **Preprocesamiento de datos:** Se llevan a cabo tareas de limpieza y transformación como la modificación o la eliminación de valores ausentes, el tratamiento de anomalías (valores atípicos), la exclusión de duplicados, la normalización o estandarización de los datos, la conversión de variables categóricas si fuese necesario, los análisis univariante y multivariante, etc.
2. ***Feature engineering*:** Más conocido por su terminología en inglés, se refiere esencialmente a la creación y selección de atributos o características que puedan ser usadas a continuación en el entrenamiento y evaluación del modelo a generar.
3. **Entrenamiento del modelo:** Elección y configuración del algoritmo a utilizar en el modelo a implementar. Habitualmente se suele usar una proporción 70 %-30 % u 80 %-20 % del total del conjunto de datos (*dataset*), siendo la primera parte de dicha proporción la usada para entrenar (*train*) el modelo y la segunda parte la referente a la prueba (*test*), de tal modo que se pueda comprobar la predicción desarrollada sobre un conjunto de pruebas diferente.
4. **Evaluación del modelo:** Se intenta buscar la efectividad del modelo creado a través de diferentes métricas de valoración. Nuestro principal cometido será encontrar la menor diferencia entre lo esperable y lo pronosticado.

Estas fases intentan englobar un comportamiento que, como veremos más adelante, requiere ser diseccionado más al detalle. Con esto será suficiente por ahora para poder seguir avanzando.

Como "no todo es color de rosa", cualquier proyecto en estas lides debe enfrentarse a no pocos problemas y retos que superar. Entre todos ellos, destacan los siguientes:

- **Calidad de los datos:** Tanto por tener una baja cantidad de datos, como por tratarse de un conjunto de datos desbalanceado. O, en este último caso y dicho de otra manera, por tener una alta asimetría (*high skewed*) en la

```
# División de los datos en entrenamiento y test
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

# Creación de capas
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

# Elección del optimizador y de la función de pérdida
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test,  y_test, verbose=2)
```

Tras entrenar y evaluar este modelo de clasificación se ha obtenido una precisión del ~98 % sobre el total del conjunto de datos. Podemos decir con cierta vehemencia que cumple las expectativas.

Ya solo nos queda profundizar en los tipos de aprendizaje profundo más conocidos: redes neuronales convolucionales, recurrentes, de grafos, etc.

Redes neuronales convolucionales (CNN)

Están especializadas en procesar imágenes, texto y vídeos siendo capaces de aprovechar la información espacial. El término "convolución" hace referencia a la operación matemática que transforma dos funciones en una tercera o, dicho de otra manera, representa la similitud entre dos señales, lo que permite relacionarlas. Por ejemplo, en el caso de una imagen, como los píxeles próximos tienden a ser similares o, lo que es lo mismo, varían mínimamente, una CNN puede llegar a interpretarlo con la ayuda de cálculos como la covarianza.

Este tipo de redes, como se aprecia en la figura 9.6, se basa en un proceso que se puede resumir de manera cronológica en las tres siguientes capas:

1. **Convolución:** En esta capa es donde se detectan características como bordes, formas, etc., aplicando un barrido con filtros (*kernel*) que, a través de la operación de convolución, son capaces de detectar la intensidad de los píxeles en diferentes secciones de la imagen. Cada filtro está compuesto por una matriz (habitualmente 3 × 3) de pesos y un sesgo, y se centra en una característica en particular. El hecho de aplicar todo esto conlleva la pérdida parcial de información, ya que se reduce el tamaño de la imagen y se desaprovechan las esquinas. Para evitar esto último, se usa la técnica de *padding* y una de sus dos posibles técnicas asociadas, más conocidas por sus nomenclaturas en inglés: *valid* y *same*. La primera no añade ningún relleno, por lo que el filtro solo se mueve dentro del espacio existente de la imagen. La segunda, en cambio, añade píxeles adicionales en el borde. Unido a todo esto, hay que especificar también el desplazamiento de píxeles (*stride*). Es decir, si indicamos un valor de 2, el barrido moverá el filtro convolucional dos unidades tanto en vertical como en horizontal, reduciendo la imagen en la misma medida.

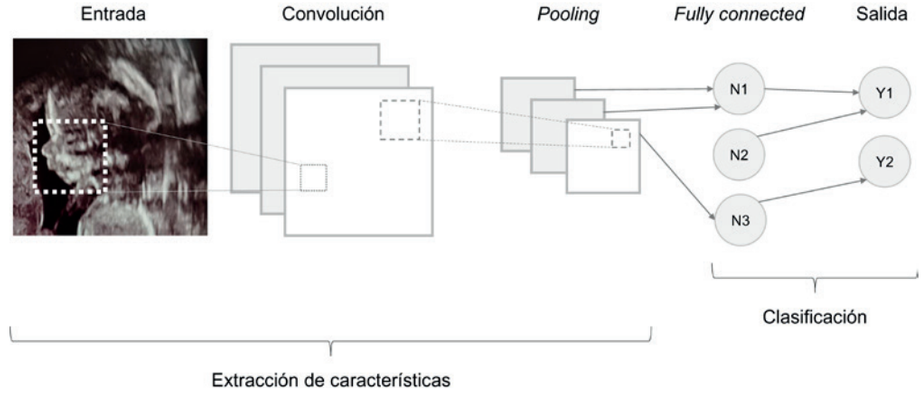

Figura 9.6. Proceso paso a paso de una red neuronal convolucional.

2. ***Pooling*:** Como su propio nombre indica, en esta capa se agrupan las características más representativas de cada sección o región escogida. Para ello, se pueden utilizar dos aproximaciones: *max-pooling* (valor máximo de la sección) o *average-pooling* (valor promedio). Su principal cometido es reducir el volumen de datos. Tanto la etapa de convolución como la de *pooling* irán siempre de la mano una tras la otra y podrán iterarse cuantas veces se estime, teniendo en cuenta factores como la complejidad o la profundidad de la arquitectura, el tamaño del *dataset* o los recursos computacionales.
3. ***Fully connected*:** Tras recibir un vector con valores de entrada, esta capa aplica una combinación lineal y una función de activación que genera a su vez un vector donde cada uno de sus elementos representa la probabilidad de que la imagen de entrada pertenezca a una clase.

Aunque este tipo de redes neuronales, casi con total seguridad, no van a ser parte de nuestro día a día en la empresa, cada vez más en la actualidad se les están encontrando aplicaciones ciertamente útiles: control de calidad de productos, búsqueda visual o recomendación de productos basada en imágenes propias, reconocimiento de firmas, extracción de textos escritos a mano, etc.

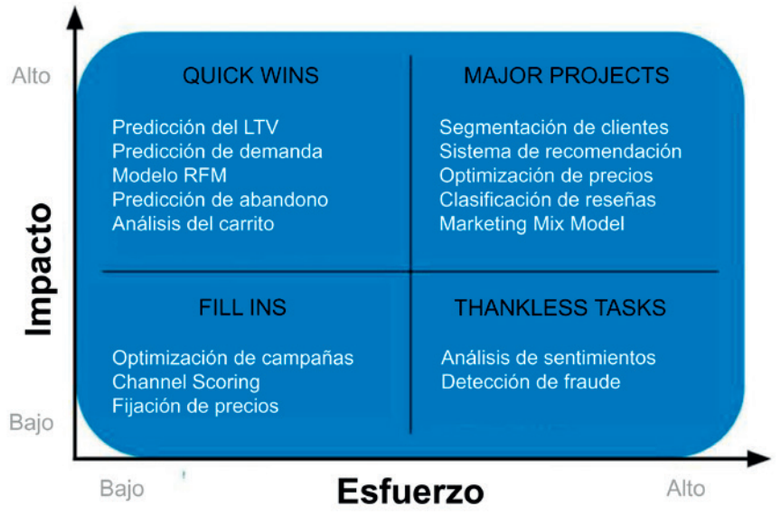

Figura 10.2. Matriz impacto-esfuerzo con desarrollos y análisis tipo.

Casos prácticos

A través de tres implementaciones desarrolladas en Python y basadas en casos reales se van a mostrar las casuísticas más habituales a las que nos hemos de enfrentar en nuestro día a día. Estamos hablando de intentar predecir y clasificar elementos de nuestro negocio como pueden ser los KPIs comentados previamente. En este sentido, para llevar a cabo este tipo de desarrollos, lo primero que hemos de tener instalado y configurado es un IDE (entorno de desarrollo integrado), el cual es un entorno de desarrollo integrado que permite programar en diferentes lenguajes, así como utilizar decenas de herramientas que nos hagan más fácil este cometido. Algunos de los IDE más conocidos quizás son PyCharm o Visual Studio Code. Además, también existe la posibilidad de utilizar servicios *online* como JupyterLab o Google Colab (navegador). Una vez que optemos por uno u otro, solo la experiencia nos dirá cuál se amolda mejor a nuestras características, deberemos instalar la versión de Python más actual, teniendo en cuenta que sea compatible con todo lo que vamos a programar más adelante. Es esencial prever esto, puesto que, si no, tendremos numerosas incompatibilidades. Python se puede instalar a través de su sitio web oficial `www.python.org/downloads/` o directamente ya incluido en entornos que incluyen absolutamente todo como puede ser Conda (`conda.io`), el cual incluye de manera gratuita aplicaciones, librerías, administración de paquetes y demás elementos para poder llevar a cabo desarrollos programáticos cercanos a la ciencia de datos.

Una vez que tenemos la herramienta para programar lista y el entorno de programación configurado (Python), solo queda empezar a desarrollar los *notebooks* sobre los que desarrollar nuestros tres proyectos. Pero ¿qué es un *notebook*? No es más que un tipo de archivo donde a través de dos tipos de celdas (código y *markdown*) se puede generar y explicar paso a paso nuestro proyecto. Crear un proyecto no solo es obtener unos resultados o crear un modelo, también se debe explicar cómo se ha llegado a ellos. Por lo tanto, el código que generemos se añadirá dentro de las celdas tipo código y las explicaciones oportunas o los títulos de cada una de las fases a llevar a cabo se añadirán en las celdas *markdown*. Comencemos pues. En este caso, se ha utilizado el IDE Visual Studio Code. Para abrir nuestro primer *notebook* solo tenemos que ir a Archivo>Nuevo archivo>Jupyter Notebook.

Predicciones

El primer caso real que vamos a detallar es una de las aplicaciones más utilizadas en cualquier tipo de empresa. Estamos hablando de generar predicciones de los KPIs más relevantes con la intención de poder prever comportamientos que ayuden a mejorar nuestro negocio. Pero, antes de profundizar en este desarrollo, es conveniente explicar la diferencia entre predecir y pronosticar. Quizás en nuestro entorno diario pueden considerarse como sinónimos; sin embargo, en este tipo de implementaciones suelen referirse a cuestiones distintas:

- **Predicción:** Es la estimación de resultados futuros basándose en información limitada o incompleta.
- **Pronóstico:** Es la generación de predicciones teniendo en cuenta tendencias y estacionalidades pasadas. Se le conoce en inglés como *forecasting*.

En este caso concreto, nos vamos a centrar en el primer punto, intentando predecir las ventas que generarán las diferentes categorías de producto existentes en el conjunto de datos `ga4_obfuscated_sample_ecommerce` en BigQuery. Este *dataset* almacena datos de la tienda Google Merchandise Store, `shop.googlemerchandisestore.com`, la cual se usa por parte de Google como ejemplo de implementación de comercio electrónico en GA4. Es una tienda *online* que genera datos sobre productos de Google y que es accesible "gratuitamente", siempre y cuando no superemos los límites mensuales establecidos de 10 GB de almacenamiento o 1 TB de consultas. Además, debemos tener en cuenta que este conjunto de datos contiene datos ofuscados, por lo que su uso solo estará aconsejado para realizar pruebas, ejemplos prácticos o desarrollos básicos. De ahí su denominación *obfuscated*. Para agregarlo a nuestro proyecto de BigQuery, simplemente deberemos acudir al panel Explorador, hacer clic en Agregar más y buscar por el término

“QUEREMOS QUE LAS PERSONAS QUE SE DEDICAN AL MARKETING DIGITAL APRENDAN LO QUE SE ESTÁN PERDIENDO POR NO SABER PYTHON”.

Esta fue la premisa que los autores, Ubaldo Hervás y Joseba Ruiz, plantearon como principal en el momento de creación de este libro. Ya seas especialista en marketing o analítica digital, CRO, producto, SEO, *performance, email marketing o social media*, te interesa conocer lo que se cuece detrás de la tecnología que soporta el negocio digital.

Existen varios lenguajes y herramientas que generan el escenario sobre el que se levanta el negocio digital y la toma de decisiones basada en datos: **Python** y sus librerías de tratamiento de datos (**Pandas**), álgebra (**NumPy, SciPy**) o ***machine learning*** (scikit-learn, statsmodels), lenguajes de consulta para base de datos (**SQL**), así como herramientas como **Google BigQuery, Microsoft PowerBI** y **estadística descriptiva e inferencial**.

Conocer en qué consisten estas herramientas y aprender un lenguaje de programación como Python te ofrecerá un abanico de soluciones totalmente superior. Si estás buscando un libro que aumente tus habilidades técnicas y convertirte en un profesional de marketing mucho más técnico y atractivo para el mercado laboral, este es tu libro.

ANAYA MULTIMEDIA

www.anayamultimedia.es

ISBN 978-84-415-4994-4

2315260

9 788441 549944