TRANSFORMACIÓN DE DATOS EN INTELIGENCIA

AUTORA: ANA LUISA ORTEGA

CONTENIDO

INTRODUCCIÓN

En el mundo actual, los datos están en todos lados. Algunas veces nosotros consumimos y cedemos nuestros datos conscientemente. Algunas otras veces, no está claro este consentimiento. Basta pensar en todas las aplicaciones que utilizamos en el día a día. Se puede resumir en la frase del libro “Age of Surveillance Capitalism” de Shoshana Zuboff, donde dice “si no puedes ver el producto, tú eres el producto” en el contexto de las redes sociales. Esta frase refleja la idea de que en muchos servicios gratuitos, como las redes sociales, los datos que creamos al utilizar y navegar son utilizados (y algunas veces vendidos) por empresas a terceros con el objetivo de ser rentables. Es decir, los datos, el día de hoy, son un activo comercial.

A nivel individual, es casi imposible escapar de la recolección masiva de datos. Es importante dicha consciencia para ser muy responsables al explotar los datos a los que tenemos acceso. Un mal uso de datos puede terminar en casos como el famoso COMPAS.

01

EL CASO DE COMPAS

¿Qué es COMPAS? COMPAS (cuya sigla proviene del inglés y corresponde a "Correctional Offender Management Profiling for Alternative Sanctions"), un *software* creado para el sistema judicial de Estados Unidos que predice la probabilidad de reincidencia de los delincuentes.

¿Cómo se creó COMPAS? Se utilizaron los datos históricos recolectados por el mismo sistema judicial, por mencionar algunos:

- La información de los delitos cometidos en el país
- Las características raciales de los acusados (su fotografía)
- Si el acusado era encontrado culpable o inocente por el delito, resultado de la deliberación de un jurado compuesto de ciudadanos
- La condena de aquellos declarados culpables dictada por el juez
- El comportamiento de los reos arrestados

¿Qué implicaciones tuvo? El resultado fue un modelo peligrosamente sesgado: asignaba mayor probabilidad de reincidencia a la población afroamericana y menor probabilidad de reincidencia a la población blanca. Es decir, el modelo aprendió a replicar el sistema que históricamente, y por acciones de humanos, ha sido un sistema discriminatorio.

Ejemplo: imagina que tienes dos acusados por el mismo delito (asumamos que robo a casa habitación), cuyas características raciales son las siguientes:

Figura 1: Acusados en COMPAS
(Pacific Standard)

En sus comienzos, COMPAS tendía a asignar un nivel de acuerdo a su probabilidad de reincidencia. Para este ejemplo, les asignó “bajo riesgo” y “mediano riesgo” a los acusados.

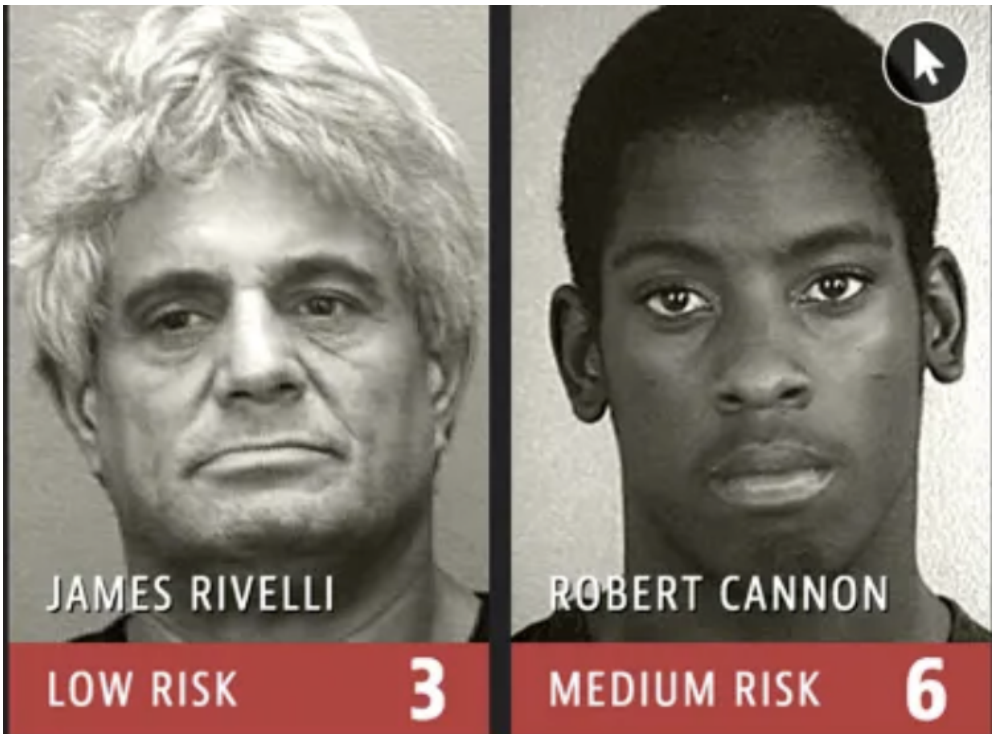


Figura 2: Nivel de riesgo de reincidencia
(Pacific Standard)

La controversia ocurría al comparar los delitos por los que se les acusaba a cada uno con el nivel de riesgo de reincidencia asociado, resumidos en la siguiente tabla:

Acusado	Delitos pasados	Riesgo de reincidencia
James Rivelli	-1 cargo por violencia doméstica agravada - 1 cargo por tráfico de drogas - 1 cargo por robo a casa habitación	Nivel bajo
Robert Cannon	-1 cargo por abuso de confianza	Nivel mediano

Figura 3: Delitos y riesgo de reincidencia
(elaboración propia)

¿Estarías de acuerdo? ¿En qué crees que el *software* está basando sus decisiones para asignar niveles de riesgo de reincidencia? Este es uno de los muchos ejemplos en donde se resalta la importancia de ser muy cuidadosos y responsables en la recolección y explotación de los datos, pues las expectativas terminan en no cumplirse al tener conclusiones no deseadas.

02

EL SESGO EN LOS DATOS

Según la Real Academia Española (RAE), el sesgo en el contexto de la estadística se define como: “error sistemático en el que se puede incurrir cuando al hacer muestreos o ensayos se seleccionan favorecen unas respuestas frente a otras.” Decimos que nuestro datos “están sesgados” cuando no están representando apropiadamente a la población que estudiaremos. Es un problema del que la mayoría no está consciente, pero es un tema de suma importancia en la actualidad, pues cada vez se aprueban más leyes de protección, transparencia y no discriminación de datos.

Por un lado, Daniel Kahnneman explora dos sistemas de pensamiento: uno que es rápido, reactivo y automático y el segundo que es lento, lógico y cuidadoso. Después explica cómo estos dos sistemas de pensamiento coexisten y afectan nuestras decisiones y juicios de formas inconscientes. Además, se exploran distintos tipos de sesgos cognitivos existentes:

EXTRA

Algunas lecturas para profundizar en este tema son:
“Pensar rápido, pensar despacio” de Daniel Kahneman
“Mujeres invisibles” de Caroline Criado Perez

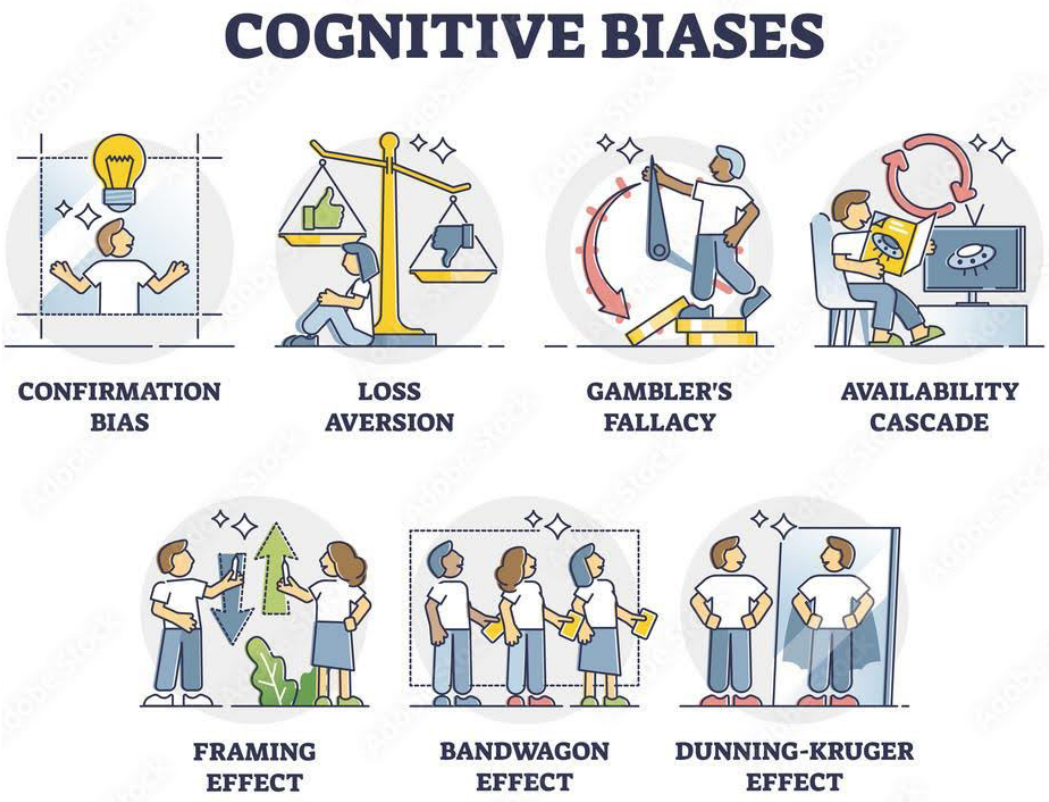


Figura 4: Sesgos cognitivos
(Adobe)

2.1. TIPOS DE SESGOS COGNITIVOS

Algunos de estos sesgos, si están implícitos en los datos, nos harán producir modelos sesgados y tomar decisiones distintas a las que en un principio planeamos. Podemos mencionar los más comunes:

- **Sesgo de confirmación:** ocurre cuando tomamos decisiones o juicios dada la información que se alinee con las creencias preexistentes. Un caso sería que al querer tomar decisiones basadas en datos, filtramos los datos usando juicios personales.

- **Sesgo de disponibilidad:** ocurre cuando tomamos decisiones o juicios con la información que estuvo disponible en ese momento. Un caso sería que al querer tomar decisiones basadas en datos, usemos datos históricos que no tomaron en cuenta alguna población (como en el mencionado caso COMPAS).

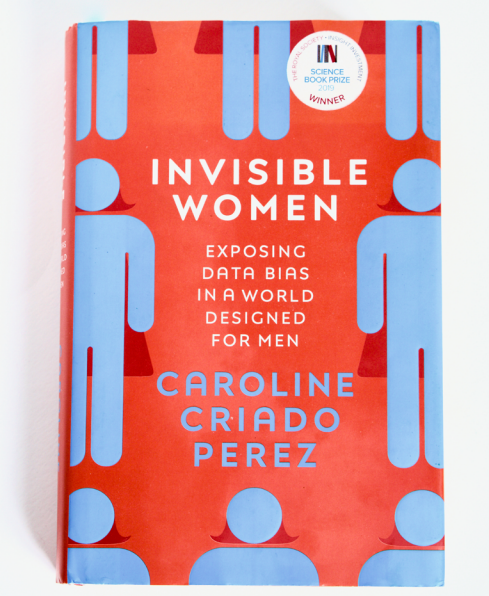
Por otro lado, Caroline Criado Pérez hace una crítica al hecho de que las decisiones históricas se han tomado considerando solamente a la población masculina. Todo esto, como consecuencia, ha creado un mundo donde las

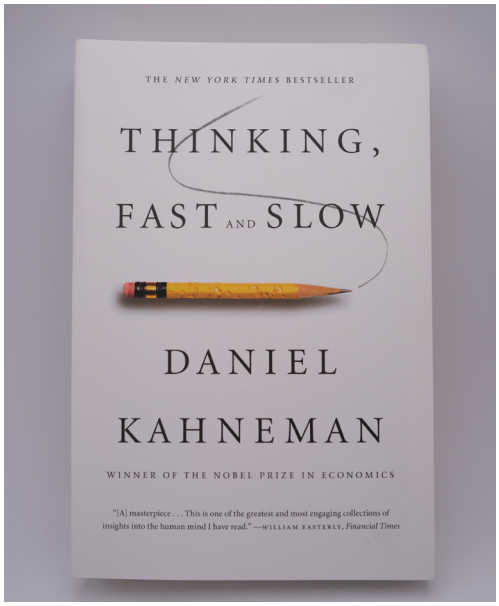
mujeres son invisibles. Veamos algunos ejemplos descritos por Caroline:

- **Salud:** los estudios clínicos se realizan, en general, contemplando las características antropomórficas de los hombres, sin contemplar las características físicas, biológicas y hormonales de las mujeres. Esto termina en tratamientos pensados para hombres aplicados a toda la población.

- **Seguridad al viajar:** los sistemas de cinturones de seguridad de los transportes personales suelen ser diseñados y probados en hombres, ignorando la diferencia promedio de altura y peso entre distintos sexos.

Sin duda, estos dos libros son una lectura recomendada para aquellos que quieran profundizar en estos temas, pues son el primer paso para construir decisiones sobre datos sin sesgos, al ser conscientes de los sesgos existentes.





03

PROCESO PARA ACCIONAR CON LA INTELIGENCIA ARTIFICIAL

Es importante entender el proceso que se sigue para crear modelos predictivos con datos y herramientas a las que tenemos acceso, es decir, el proceso que se sigue para poder accionar con la inteligencia artificial.

Repasemos el proceso general que se sigue para crear un modelo, el cual consta de pasos secuenciales que se describirán a lo largo de la materia. Es importante mencionar que se deberán ajustar los pasos según las necesidades del objetivo a resolver, pues no todos serán necesarios en todas las problemáticas a resolver.

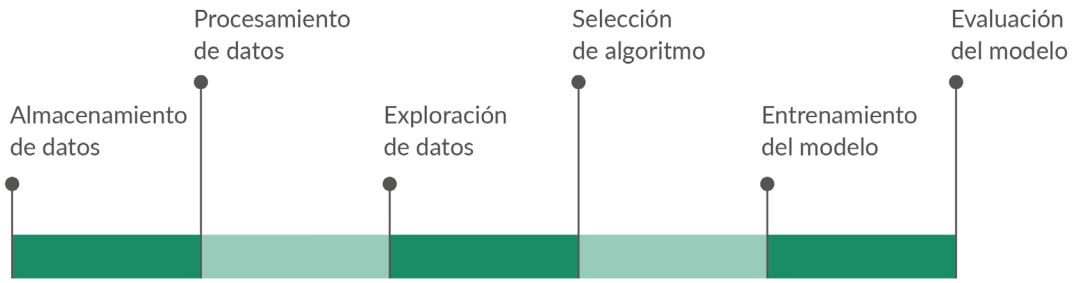


Figura 5: Pasos para accionar con la inteligencia artificial
(elaboración propia)

Recordemos que en los recursos audiovisuales de la semana en curso se habló de los primeros dos pasos específicamente: **almacenamiento de datos y procesamiento de datos.** Exploremos cada una de estas etapas para lograr el objetivo final: entender cómo se entretejen todos estos pasos.

3.1. ALMACENAMIENTO DE DATOS

En general las empresas e instituciones son aquellos entes que recolectan datos de las personas con mucha más facilidad, sobre todo datos que ayuden con sus objetivos: mejorar las ventas, retener clientes, llegar a mayor penetración de mercado. Es por eso que, como se ha mencionado a través de este curso se debe seguir los siguientes puntos:

- Ser cuidadoso en la recolección de datos, siendo conscientes de la responsabilidad que implica y el poder que pueden tener los datos.
- Se debe especificar el fin u objetivo de la recolección de los datos.
- También ser claro y transparente, es buena idea tener redactado los detalles en una sección de términos y condiciones.

Los datos pueden clasificarse en alguno de los dos siguientes:

1. Datos tabulares

Son aquellos datos que pueden ser representados en filas y columnas. Algunos formatos en los que se almacenan estos datos son: XLS (eXtensible Stylesheet Language) y CSV (Comma-Separated Values). Por ejemplo, los datos de las ventas de una empresa son datos tabulares.

2. Datos no tabulares

Son aquellos datos que no pueden ser representados en filas y columnas; no tienen una estructura fija. Algunos formatos en los que se almacenan estos datos son: JSON (JavaScript Object Notation) y MP3(MPEG-1 Audio Layer III). Por ejemplo, las imágenes del catálogo de productos de una empresa son datos no tabulares.

	Loremis t	Amis terim	Gato lepis	Tortores
Lorem dolor siamet	8 288	123 %	YES	$89
Consecter odio	123	87 %	NO	$129
Gatoque accums	1 005	12 %	NO	$99
Sed hac enim rem	56	69 %	N/A	$199
Rempus tortor just	5 554	18 %	NO	$999
Klimas nsecter	455	56%	NO	$245

Figura 6: Catálogos de productos
(Shutterstock)

3.1.1. Opciones de almacenamiento de datos

Una vez que se definan aquellos datos de interés para la empresa y se identifique el tipo de los datos, existen diversas opciones para almacenar los datos recolectados.

- **Hojas de cálculo:** son las más utilizadas, comúnmente en programas como Excel o Google Sheets. Permiten almacenar datos tabulares (en filas y columnas). Una ventaja de las hojas de cálculo es que los programas ofrecen funciones y herramientas para hacer análisis numéricos, gráficos, entre otros de una forma rápida. Una desventaja es que las hojas de cálculo dejan de ser eficientes con un volumen muy grande de datos.

- **Archivos CSV:** formato de texto plano que almacena datos tabulares separados por comas. Es fácil de leer y escribir, y ampliamente compatible con diversas aplicaciones y lenguajes de programación. Los archivos CSV almacenan datos tabulares de una forma más eficiente en temas de memoria.

- **Almacenamiento en la nube:** servicios que usualmente tienen costo, que almacenan distintos tipos de datos en un clúster independiente. El servicio más conocido es AWS (Amazon Web Services), y basta ver la siguiente imágen para entender que en AWS se puede almacenar cualquier tipo de datos: tabulares y no tabulares.

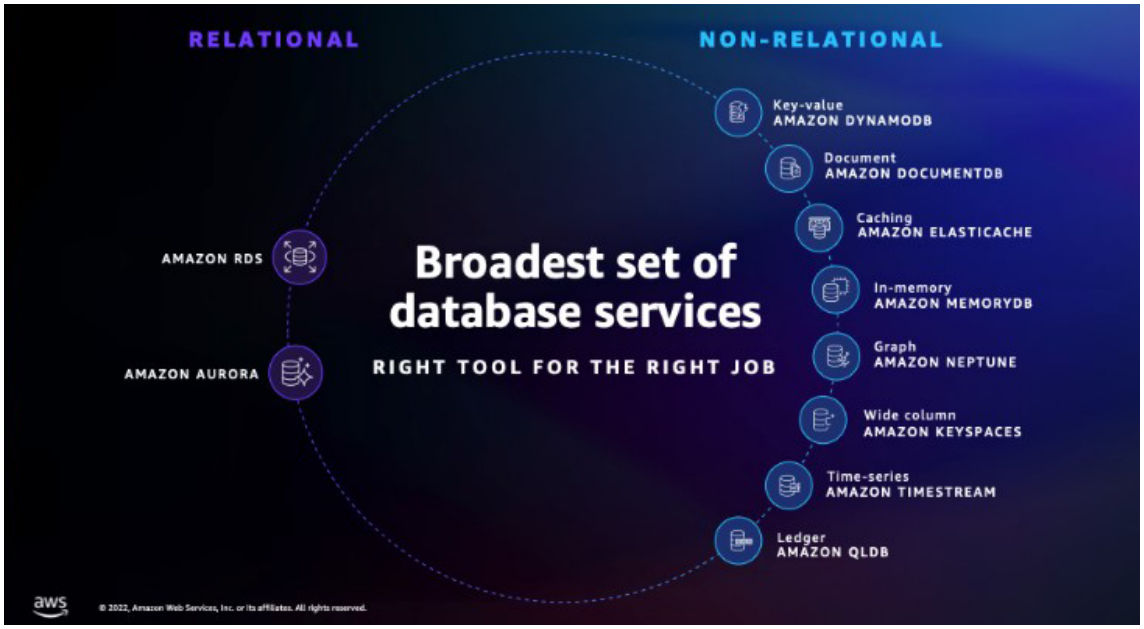


Figura 7: Amazon Web Services
(AWS)

- **Kaggle:** En esta sección vale la pena mencionar una alternativa muy popular donde se pueden encontrar datos (tabulares y no tabulares) para crear análisis y modelos predictivos. Kaggle se autodefine como "la comunidad de ciencia de datos más grande del mundo, con herramientas y recursos poderosos para ayudarte a alcanzar tus objetivos en ciencia de datos."Basta acceder a la página web https://www.kaggle.com/ para encontrar competencias, datos, modelos y código con el que se pueden explorar datos y soluciones propuestas por la comunidad para distintas industrias.

3.1.2. Caso de éxito

La competencia a nivel principiante más famosa en Kaggle es "Titanic: Machine Learning from Disaster" en donde más de 14.000 equipos alrededor del mundo han contribuido con alternativas para crear un modelo que predice la probabilidad de los pasajeros de sobrevivir ante el accidente del Titanic.

El archivo usado para esta competencia es un CSV. El que haya tantos equipos y personas trabajando en esta solución ha ayudado a que las personas principiantes puedan mejorar sus habilidades muy rápidamente, incluso algunos que proponen soluciones innovadoras pueden llegar a conseguir empleo.

Figura 8: Kaggle y el Titanic
(Shutterstock)

3.2. PROCESAMIENTO DE DATOS: SQL

Una vez que se tienen datos almacenados en alguna de las alternativas mencionadas en la sección anterior, la pregunta natural inmediata es: ¿cómo puedo consultar esos datos almacenados?. Existe una variedad de herramientas para consultar y procesar datos, donde se destacan SQL y lenguajes de programación (como Python y R). Exploremos la herramienta SQL, pues Python se explorará en las siguientes semanas del curso.

- **SQL**

SQL (Structured Query Language) es un lenguaje de consulta estructurada utilizado para gestionar y manipular bases de datos. Al ser un lenguaje de consulta estructurado, es importante conocer las instrucciones que podemos hacer y aún más importante, cómo se escriben dichas instrucciones.

Por simplicidad, asumamos que tenemos una base de datos con una tabla llamada ventas en donde almacenamos las ventas históricas de una tienda de perfumes. Para tres renglones, se vería de la siguiente forma:

Id	Sucursal	Vendedor	Fecha	tot_productos	pago_total
1	Matriz	5	17/04/24	3	1,300
2	Matriz	8	17/04/24	1	200
3	Suc1	2	17/04/24	1	750

Figura 9: Base de datos
(Elaboración propia)

Consultas en SQL

- **Consulta general:** la instrucción en SQL para ver todos los renglones y todas las columnas de una base de datos (en este caso, llamada ventas) es la siguiente:

```
SELECT *
FROM ventas;
```

- **Consulta general limitando renglones:** algunas veces queremos ver todas las columnas pero sólo un par de renglones (en este caso, 100 renglones). La instrucción en SQL es la siguiente:

```
SELECT *
FROM ventas
LIMIT 100;
```

- **Consulta de columnas limitando columnas:** otras veces queremos ver todos los renglones pero sólo estamos interesados en unas columnas en particular (en este *caso sucursal, tot_productos y pago_total)*. La instrucción en SQL es la siguiente:

```
SELECT sucursal, tot_productos, pago_total
FROM ventas;
```

- **Consulta donde se cumple una condición.** Cuando queremos ver todos los renglones que cumplen una condición, lo podemos hacer usando la cláusula WHERE y algún operador lógico (<, <=, = ,>, >=). La instrucción en SQL para encontrar las ventas del vendedor 5 es la siguiente:

```
SELECT *
FROM ventas
WHERE vendedor = 5;
```

- **Consulta ordenando los valores de alguna columna:** si nos interesa ver todos los renglones y columnas, pero que se muestran ordenando el valor de alguna columna (en este caso, *pago_total)*, tendríamos la siguiente instrucción:

```
SELECT *
FROM ventas
ORDER BY pago_total;
```

- **Consulta combinando las instrucciones anteriores:** las instrucciones mostradas hasta ahora no son excluyentes, es decir, se pueden utilizar todas a la vez (o las que más nos convenga, según nuestras necesidades). En el caso en donde quisiéramos consultar una tabla llamada *ventas* con todas las siguientes condiciones:

  - » seleccionando columnas sucursal, *tot_productos y pago_total*
  - » limitar a sólo mostrar los primeros 100 renglones
  - » filtrar a sólo mostrar información del vendedor 5
  - » ordenado por el *pago_total*

tendríamos la siguiente instrucción:

```
SELECT sucursal, tot_productos, pago_total
FROM ventas
WHERE vendedor = 5
ORDER BY pago_total
LIMIT 100;
```

Funciones en SQL

Una de las últimas utilidades importantes a mencionar de SQL son las funciones. Hay casos donde nos interesa hacer alguna operación sobre alguna columna de la base de datos más allá de ver columnas y renglones especificados.

Es importante recalcar que el orden sí importa: no podemos poner una instrucción antes que otra (por ejemplo, LIMIT siempre va después de ORDER BY). Además, podemos omitir cualquiera de las instrucciones menos SELECT y FROM, las cuales son lo mínimo necesario para ejecutar una consulta.

Volviendo al ejemplo de ventas, un caso específico que nos interesaría sería saber la venta total de las tiendas. Eso podría hacerse haciendo una suma de la columna *pago_total* y usando la función SUM.

```
SELECT SUM(pago_total)
FROM ventas;
```

A continuación podemos ver una tabla con las funciones más utilizadas en SQL:

Función	Descripción	Ejemplo
COUNT	Contar el número de renglones de una base de datos.	SELECT COUNT(*) FROM ventas;
SUM	Sumar los valores de una columna de una base de datos.	SELECT SUM(pago_total) FROM ventas;
AVG	Calcular el promedio o media de una columna de una base de datos.	SELECT AVG(pago_total) FROM ventas;
MIN	Calcular el valor mínimo de una columna de una base de datos.	SELECT MIN(pago_total) FROM ventas;
MAX	Calcular el valor máximo de una columna de una base de datos	SELECT MAX(pago_total) FROM ventas;

Figura 10: Funciones más usadas de SQL
(elaboración propia)

Agrupaciones en SQL

Una tarea más compleja se puede realizar con ayuda de las funciones en SQL. Esta tarea consiste en los siguientes pasos:

1. Agrupar la información por los distintos valores de una columna.
2. Usar alguna función para sacar una estadística de todo el grupo.
3. Mostrar la información resultante.

Por ejemplo, en el ejemplo de ventas, podemos estar interesados en sumar el

monto total de *ventas* por cada sucursal. Asumiendo que la tabla sólo consta de los 3 renglones mostrados, primero quisiéramos agrupar por la columna sucursal para después sumar, por *sucursal*, el total de la columna *pago_total*. Las instrucciones descritas se escriben con el siguiente formato en SQL:

```
SELECT sucursal, SUM(pago_total)
FROM ventas
GROUP BY sucursal;
```

Para el ejemplo de ventas, el resultado de ejecutar la instrucción (también decimos “ejecutar el query”) es el siguiente:

Sucursal	pago_total
Matriz	1,500
Suc1	750

Figura 11: Ejecutar el query
(elaboración propia)

Notemos que podemos agrupar por distintas columnas y podemos utilizar cualquiera de las funciones de SQL descritas.

04

EJEMPLO DE NEGOCIO

Para finalizar, pensemos en un ejemplo de negocio donde sigamos los pasos descritos en esta semana del curso. Utilizaremos el ejemplo de las *fintech* en Latinoamérica que son alternativas a los bancos tradicionales y el problema que tienen al otorgar tarjetas de crédito a las personas.

Para poder tomar esta decisión, primero necesitamos tener nuestros datos almacenados. Si la empresa es nueva, el camino más fácil es comprar datos de los burós de crédito de cada país. El camino alternativo y costoso es crear un experimento en donde generas datos que te ayuden a tomar la decisión.

Supongamos que nos inclinamos por comprar datos históricos. Entonces, los debemos almacenar en alguna de las alternativas vistas: puede ser en local en un archivo CSV o en AWS.

Después, debemos determinar cuáles variables de todas las que compramos nos será de utilidad para resolver nuestro problema, así como la variable que vamos a intentar predecir (en este caso, debería estar relacionada con información de pago o impago de otras tarjetas de crédito).

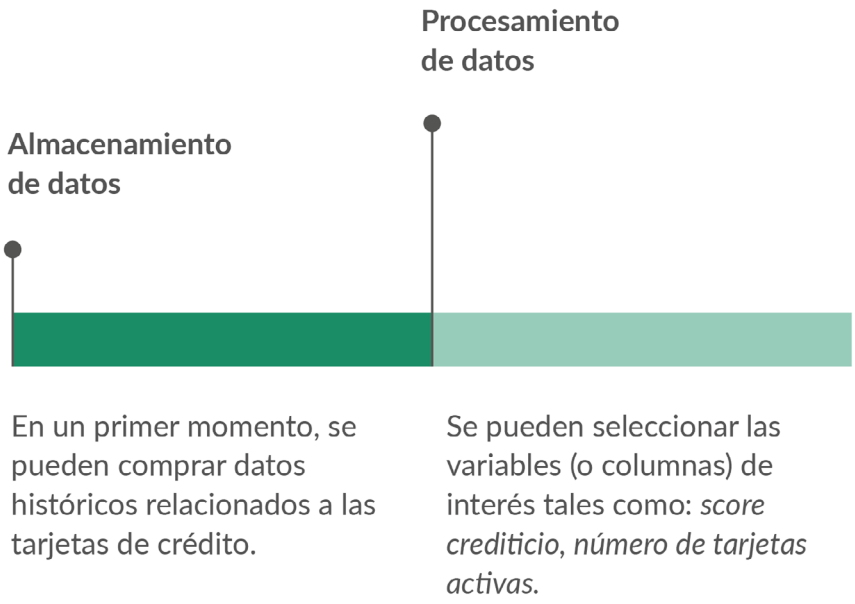


Figura 12: Pasos
(elaboración propia)

Como podrás notar cada una de las etapas se puede definir y adaptar dependiendo de cada problema, haciendo este proceso algo único que cumpla con las necesidades del negocio.

05

IDEAS CLAVE Y CONCLUSIONES

En esta primera semana repasamos las implicaciones éticas de trabajar y explotar datos, para poder reflexionar en el nivel de responsabilidad y consciencia que debemos tener al manipular datos.

Después, repasamos el proceso general que seguimos para accionar con la inteligencia artificial. En el primer paso (el almacenamiento de datos) vimos que existen distintas alternativas (como Spreadsheet, CSV y AWS) y la elección de almacenamiento debe tener en cuenta las necesidades del negocio y cada caso particular.

Por último, vimos que el segundo paso (el procesamiento de datos) es aquel donde extraemos la información necesaria. Para esto, existen herramientas como SQL, un lenguaje de consulta estructurado que nos facilita el procesamiento.

BIBLIOGRAFÍA

AMAZON (S.F.). “Databases”. https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html

CRIADO PEREZ, C. (2019). “Mujeres invisibles”. Nueva York, Abrams.

DEB, E. (2023). “COMPAS — an AI tool sending or keeping people in Jail“. https://fiatlexica.medium.com/compas-an-ai-tool-sending-or-keeping-people-in-jail-d9228df3a2c6

KAGGLE https://www.kaggle.com/

TANIMURA, C. (2021). “SQL for Data Analysis”. California, O’Reilly Media.

ZUBOFF, S. (2019).. “The Age of Surveillance Capitalism”. Profile Books.

Las imágenes de portada fueron tomadas de Shutterstock.