

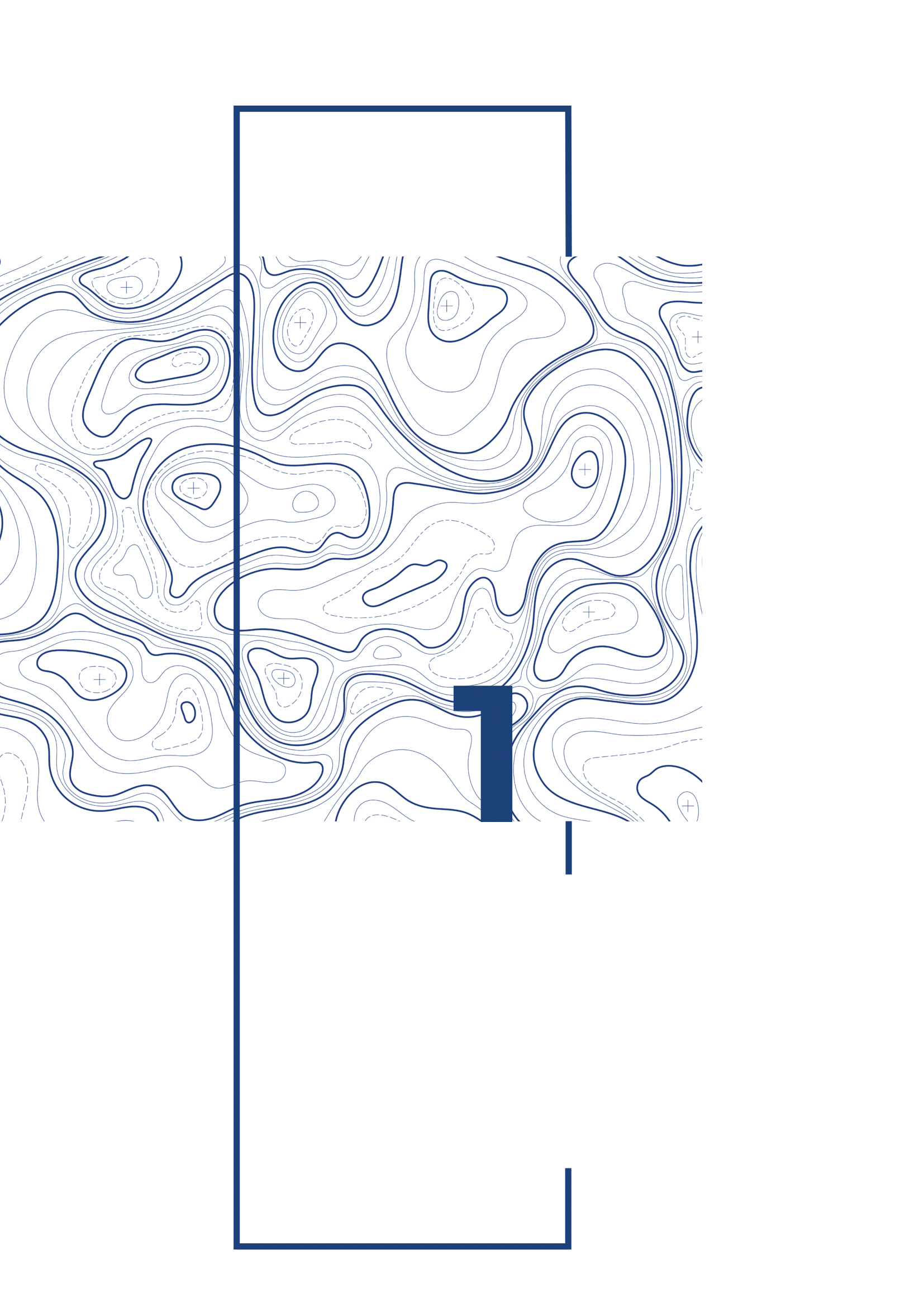
# **Proyecto de Datos I**

## **Preparación de datos**

**Óscar Marín Esteban**

**Carlos Mantilla Mateos**

**Álvaro Enol Alonso Ortega**



# **EXTRACCIÓN DE LOS DATOS**

**Vamos a usar diferentes fuentes de datos, entre ellas las siguientes:**

**Fuente 1: Página de Whoscored (**[**Link**](https://www.whoscored.com/Regions/206/Tournaments/4/Seasons/9682/Stages/22176/PlayerStatistics/Spain-LaLiga-2023-2024)**), de esta página hemos sacado todas las características generales de los jugadores.**

**Librerías o herramientas:**

**Las librerías utilizadas son selenium para hacer todo el tema relacionado con la extracción de datos, pandas para el manejo del dataframe, HTML para el manejo en si de diferentes elementos de la web y time para esperar el tiempo suficiente antes de acceder a un elemento.**

**Pseudocódigo, Aspectos y Problemas:**

**En cuanto a la captura, lo que hacemos es, primero que nada, inicializar el navegador, una vez inicializado, tuvimos un problema con las cookies, porque al iniciarlo nos salía un aviso de cookies, por lo que lo que hicimos fue una función que pasa las cookies nada más iniciar la web, que pique en el botón correspondiente. Una vez pasadas las cookies pasamos a leer la tabla leemos la primera página de el primer año seleccionado, tenemos dos bucles, uno por los años y otro por el número de páginas que hay, en este segundo, para a partir de la primera página, teníamos un problema al cambiar la página, ya que salía un icono privacy que no se podía quitar, por lo que optamos por la solución de bajar la página un poco para que desapareciera con (driver.execute\_script("window.scrollBy(0, 750)")), a partir de aquí ya habríamos hecho todo lo necesario para tener todos los datos en nuestras manos.**

**Fuente 2: Página de Transfermarket, de aquí hemos sacado principalmente, aunque se pondrá detalladamente más adelante, la posición exacta, la edad y el precio.**

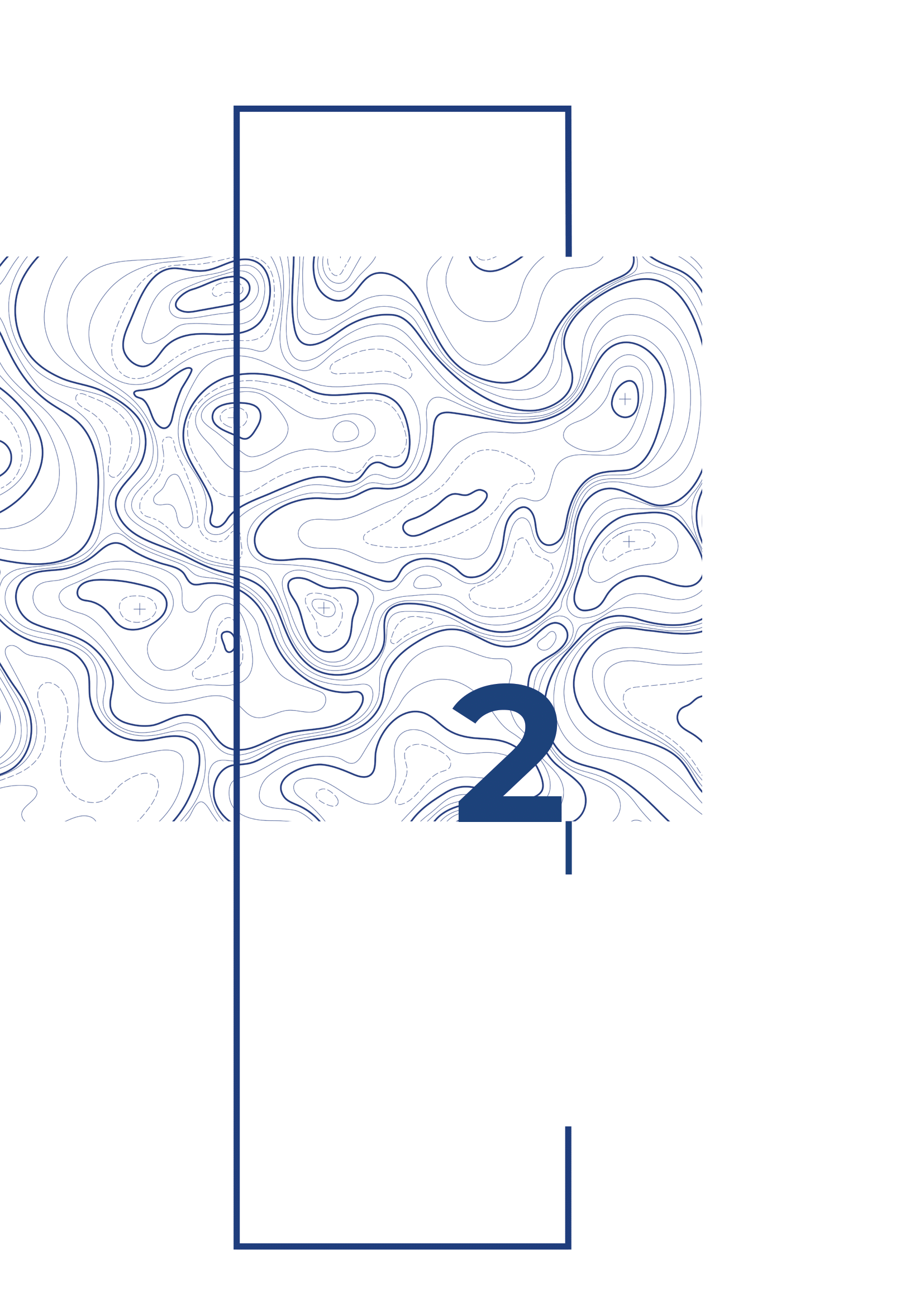
**Librerías o herramientas:**

**En este caso, hemos utilizado una API, ya que al sacarlo normal no nos dejó, por lo que tras investigar un poco descubrimos esta API.**

**Pseudocódigo, Aspectos y Problemas:**

**A partir de aquí, todo fue bastante sencillo, simplemente usar la API, e ir añadiendo lo leído a un dataframe, que luego descargaremos, lo realmente difícil fue encontrarla y aplicarla al principio.**

**.**

* + - 1. ****

# **TRANSFORMACIÓN DE LOS DATOS**

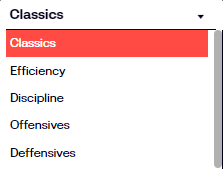
1. **Definición de la tarea de aprendizaje automático**
   * + 1. Vamos a utilizar regresión lineal para predecir el valor de mercado de los jugadores. Esto se debe a que vamos a intentar predecir una variable cuantitativa basándonos en el resto de variables. La regresión nos permite realizar una evaluación precisa del valor, crucial para las decisiones financieras y estratégicas de la compañía.
2. **Definir la variable objetivo**

Nuestra variable objetivo va a ser el valor de mercado del jugador de fútbol al final de la temporada. Solo puede tomar valores reales positivos en la moneda específica, en este caso, euros. Puede haber valores extremos para jugadores muy destacados o situaciones excepcionales. Además, es probable que haya concentraciones de valores en ciertos rangos con algunos jugadores teniendo valores muy superiores a la media.

Valores de esta variable los vamos a encontrar en diferentes páginas web que los recopilan, como pueden ser [Transfermarkt](https://www.transfermarkt.es/) o [La Liga Beyond Stats](https://www.laliga.com/en-GB/advanced-stats). Los obtendremos mediante *web scrapping,* con el código que se indicará en el apartado 2.4

1. **Identificar los datos de entrada**
   * + 1. Los datos de entrada que usaremos para predecir nuestra variable objetivo serán tanto cualitativos, como el equipo en el que se encuentra el jugador, la posición en la que juega (lo cual nos permitirá clasificar los jugadores en diferentes modelos), y si ha participado en un campeonato recientemente, como puede ser la Champions; y cuantitativos, como pueden ser los partidos jugados, los goles metidos o que le han metido (según su posición), el valor de mercado de la temporada anterior, tarjetas recibidas y faltas realizadas, minutos jugados, posiciones en otros campeonatos, etcétera.
       2. Estos valores pertenecerán tanto a la temporada anterior, en relación con el valor de mercado de esa temporada, como los de la temporada actual.
2. **Fuente de datos principal**

Sacaremos la mayoría de los datos estadísticos de la página de la Liga Beyond Stats, que nos ofrecerá una vista de las diferentes habilidades de los jugadores, a continuación se pasa el código utilizado para sacarlo:

Nota: Para sacar las diferentes secciones hay que cambiar deffensives por la que queramos, la razón por la cual no hicimos un bucle con todas las categorías, es porque se nos capa el número de requests, y ya hay problemas incluso para sacar los de todas las posiciones de una categoría, por lo que para poder obtener los datos lo hemos dejado así. Posteriormente se hará un concat junto con un merge para obtener el dataframe general.

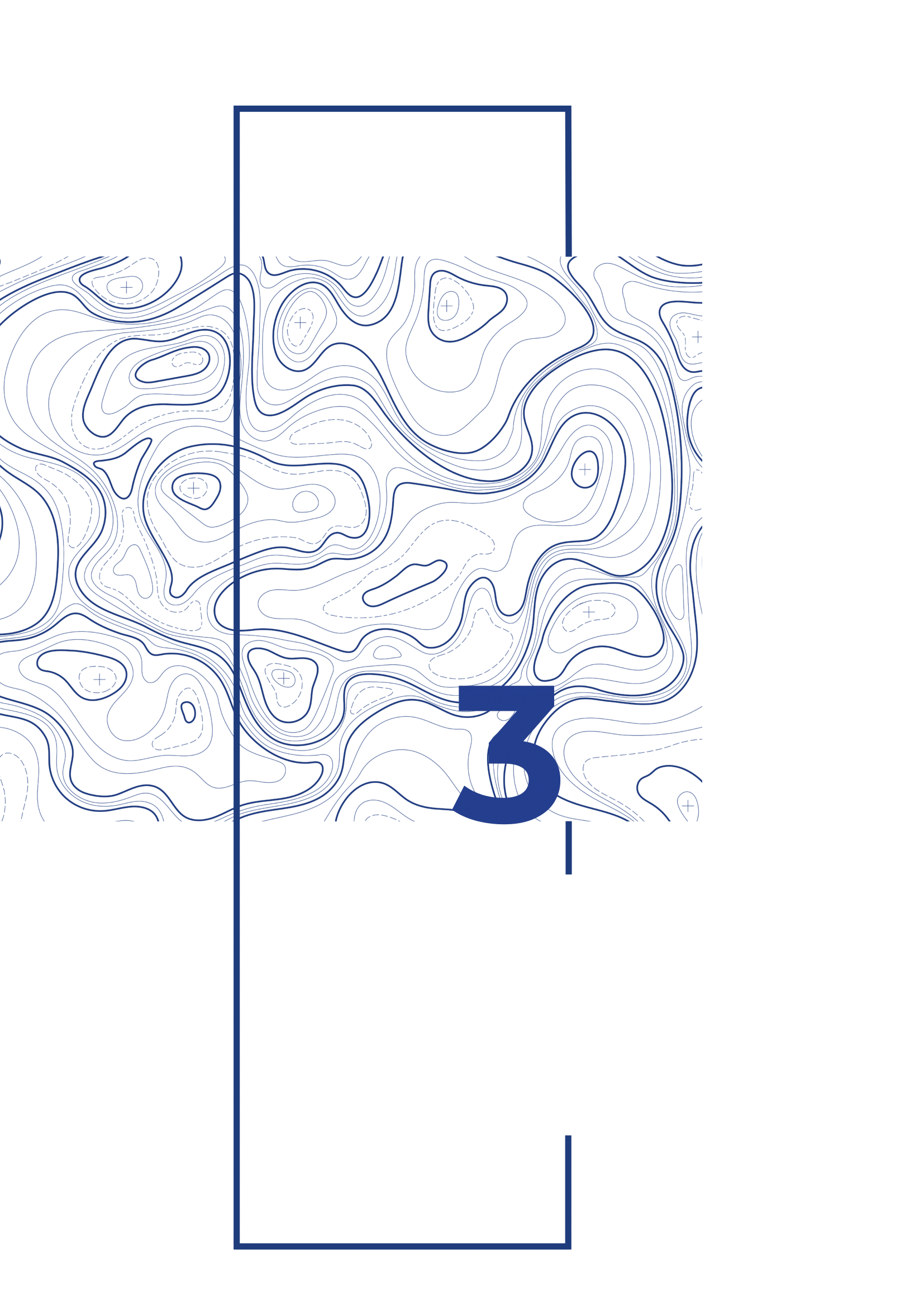
1. **Fuentes de datos secundarias**

Las fuentes de datos secundarias nos servirán de base para sacar el precio de mercado al principio de la temporada, su edad, su contrato, de qué está lesionado, cuál es su pie bueno y su precio actual, la fecha de medición como actual la especificaremos en la próxima entrega, cuando hayamos analizado detenidamente qué sería lo mejor, este es el precio con el que compararemos nuestras predicciones, lo sacaremos de transfermarket, concretamente de una API de este, encontrada en (<https://github.com/felipeall/transfermarkt-api?tab=readme-ov-file>), para la próxima entrega se juntarán estos dos conjuntos de datos, ya que por falta de tiempo no lo hemos podido hacer, y suponen un problema al juntarlos por nombre.

El código presentado es para sacar los dos conjuntos de datos adjuntados.

1. **Factores que afecten al rendimiento**

Factores que pueden afectar a nuestro sistema pueden ser personales como daños sufridos en un partido que pueden afectar al jugador a la larga, ya sea debido a sus secuelas o al tiempo que se tarde en recuperar, como roturas de huesos; hasta aquellos que no pueden ser controlados por los jugadores como puede ser la cancelación de un campeonato.

****

# **CARGA DE LOS DATOS**

## **Evaluación de éxito del sistema**

La medida en este caso que se puede utilizar para medir el éxito del sistema podría ser un error relacionado con la oferta aceptada por el jugador si es fichado por otro equipo, las ofertas de otros equipos y su relación con la nuestra si el jugador ha aceptado nuestra oferta y el valor de mercado del jugador al final de la temporada.

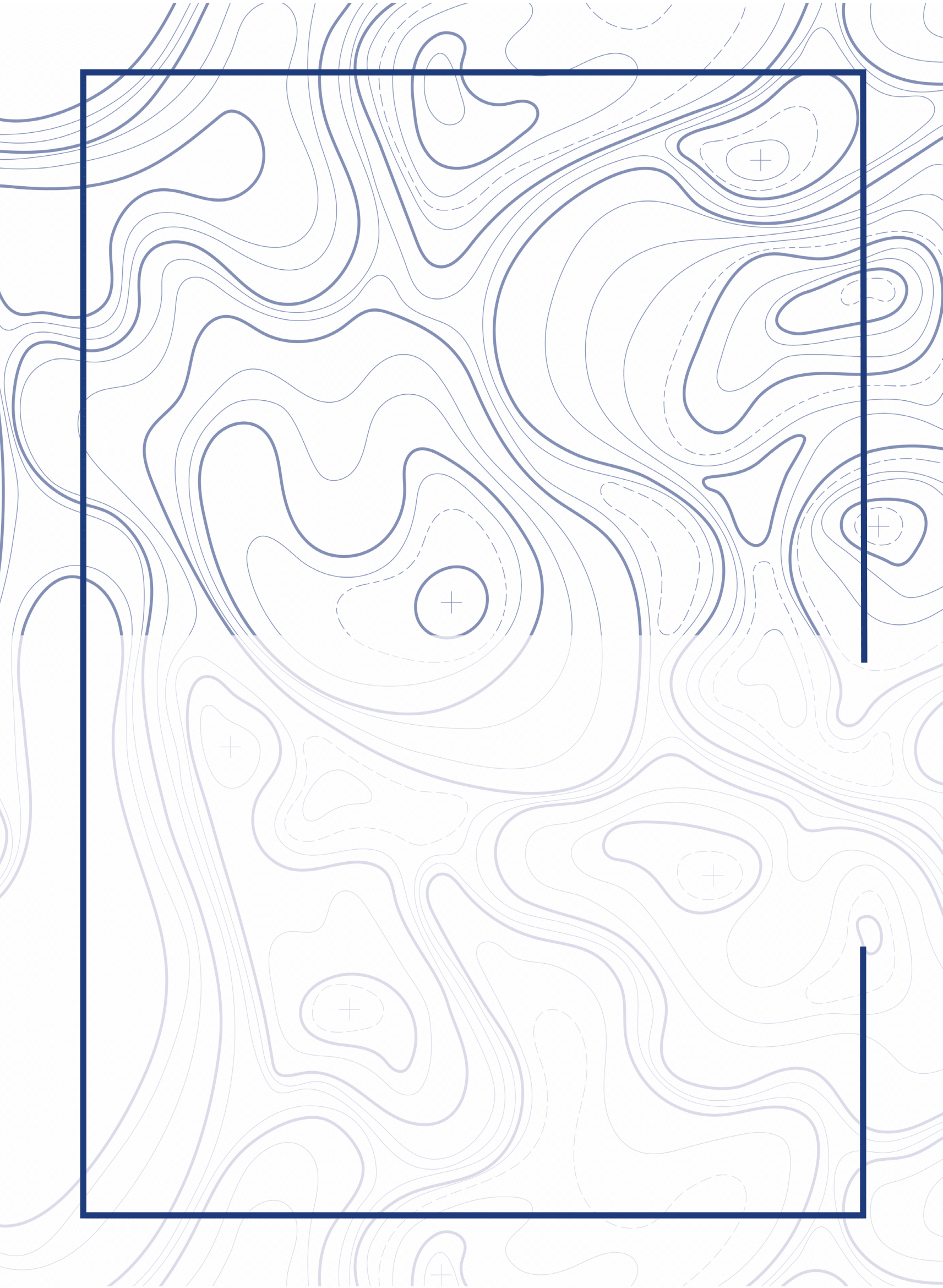
Además, el valor de mercado del jugador al final de la temporada se puede comparar con el que nuestro modelo predecía a lo largo de esta, así como su evolución según los partidos jugados o los campeonatos ganados. Estas medidas nos pueden ayudar a mejorar las predicciones de nuestro modelo y determinar sí dichas predicciones se ajustan a la realidad.

Este modelo no es el único que se puede crear, por tanto, la existencia de otros modelos puede ayudar a mejorar el nuestro.

1. **Evaluación de éxito de la tarea de aprendizaje automático**

Las medidas que nos pueden ayudar a evaluar la tarea del aprendizaje automático pueden ser la coherencia con valores del pasado, los errores cuadráticos medios, el cumplimiento de determinadas hipótesis como independencia, homocedasticidad y normalidad de los residuos.

Estas medidas nos ayudan a saber si nuestro sistema es válido para realizar las predicciones, así como la posibilidad de error de nuestro modelo, la cual debemos minimizar. En cambio, la evaluación de éxito del sistema nos permite saber si funciona correctamente con valores reales, no con los del test, así como su precisión en un ámbito real. La evaluación de la tarea de aprendizaje automático es imprescindible: si nuestro modelo falla, la empresa no tendrá ningún uso para ello y no contratará nuestros servicios. En cambio, la evaluación de éxito del sistema sirve para ver su precisión en ámbitos reales, por lo que siempre podemos ajustar nuestro sistema para hacerlo más efectivo según nuevos valores. La empresa querrá un sistema que evolucione en el tiempo, ya que es posible que el modelo cambie a lo largo de los años y sea necesario retocar el sistema.

****