UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRADO DE MATEMÁTICAS

Trabajo de fin de grado

ESTIMACIÓN DE RESULTADOS DEPORTIVOS MEDIANTE MODELOS LINEALES GENERALIZADOS

Autor: Javier Canela Ribas

Director: Josep Fortiana
Realizado en: Departamento de Matemáticas e Informática
Barcelona, 24 de enero de 2023

Resumen

En este trabajo se explica el contexto histórico sobre diversos estudios que se han hecho tratando de encontrar una forma óptima para predecir el resultado de un partido de futbol. Explicaremos, definiremos y detallaremos también como calcular los modelos lineales generalizados.

Finalmente trataremos de aplicar lo aprendido de cara a definir un modelo de regresión lineal de Poisson para predecir los resultados de una temporada de fútbol entera.

Agradecimientos

Quisiera hacer una especial mención a Josep Fortiana, mi tutor para este trabajo, su implicación en él desde el primer momento que contactamos. En todo momento se ha mostrado proactivo convocando una reunión semanal desde principios del semestre. Me ha proporcionado un enorme número de documentos y fuentes de información para poder profundizar en este trabajo.

Índice

1 Introducción

Queremos crear un modelo estadístico para predecir los resultados de los partidos de fútbol de la liga Profesional española. Para ello necesitaremos el historial de los partidos disputados a lo largo de los últimos años, obteniendo las estadísticas y resultados.

Para poder hacer un estudio sobre la fiabilidad de nuestro modelo, seleccionaremos los resultados previos a la temporada 2021-2022 con ellos entrenaremos a nuestro modelo y finalmente lo probaremos tratando de predecir los resultados de la temporada 2021-2022. Como conocemos todos los resultados de esta última temporada, podemos hacernos una idea de la precisión de nuestro modelo.

Un evento de fútbol tiene tres resultados posibles: victoria del equipo local (1), empate (x), y victoria del equipo visitante (2). En el supuesto caso de que todos los equipos fueran idénticos, las probabilidades de cada uno de estos sucesos serian también iguales (1/3, 1/3, 1/3) por lo que, si apostáramos que todos los partidos de una temporada los gana el equipo local, nuestro modelo tendría una fiabilidad del 33,33%.

En la vida real, no todos los equipos son iguales por la cantidad de factores distintos que diferencian a cada uno de ellos, tales como la calidad de los jugadores, de los entrenadores, calidad de las instalaciones o el dinero que tiene cada club para poder mejorar su equipo entre otras. De esta manera, nuestro modelo predictivo debería tener en cuenta muchos de estos factores para finalmente conseguir una fiabilidad superior al 33,33% que hemos visto anteriormente.

Así pues, el objetivo de este trabajo es conseguir una fiabilidad superior al 50%. Veamos si es posible.

Tras realizar una búsqueda inicial, encontramos una amplia variedad de enfoques para modelar el fútbol. De donde nacen tres cuestiones relevantes desde un punto de vista estadístico:

1. La técnica utilizada para modelar los resultados: incluye diferentes modelos de regresión, modelos mixtos, enfoques bayesianos, …
2. La variable dependiente elegida para el modelado y/o la unidad de muestra considerada: puede ser el número de goles, el resultado (victoria, derrota o

empate), la diferencia de goles o los puntos obtenidos en cada partido, o bien el total de goles, puntos o clasificación de los equipos en una competición, etc.

3. Variables explicativas para tener en cuenta: pueden incluir la capacidad goleadora y defensiva de cada equipo, el hecho de jugar en casa, los resultados obtenidos en partidos previos, la diferencia competitiva entre los equipos enfrentados e incluso eventos ocurridos durante el propio partido entre otras.

En el mundo del deporte es común utilizar la distribución de Poisson predecir la cantidad de goles que anotarán dos equipos en un partido. Esta distribución se basa en la suposición de que cada equipo tiene una tasa de gol constante y que los goles anotados son independientes entre sí. Aunque esto puede ser cierto en algunos casos, varios investigadores han demostrado que existe una correlación entre el número de goles que anotan los dos equipos en un partido. Esto se debe a que los equipos interactúan durante el juego y pueden afectar el resultado del otro.

Sin embargo, en la mayoría de los enfoques de modelado que utilizan la distribución de Poisson, se ignora este hecho y se supone que los goles son independientes. Esto se debe en parte a la complejidad de utilizar técnicas más avanzadas para modelar la correlación entre los goles de los dos equipos. Sin embargo, algunos autores como Maher (1982) y Dixon y Coles (1997) han tratado de abordar este problema al introducir un tipo de dependencia indirectamente en el modelo de Poisson independiente.

En el caso de deportes de equipo como el fútbol, es particularmente relevante tener en cuenta la correlación entre el resultado de los dos equipos. Esto se debe a que estos deportes involucran una interacción constante entre los equipos durante el juego y el resultado de un equipo puede afectar directamente al del otro. Además, en algunos deportes como el baloncesto, los equipos anotan puntos en secuencia y la velocidad del juego de un equipo puede crear más oportunidades para que ambos equipos anoten.

Otra opción para modelar el número de goles en un partido de deporte es utilizar la distribución de Poisson bivariante. En este caso, las distribuciones marginales son distribuciones de Poisson, mientras que las variables aleatorias están ahora correlacionadas. Maher (1982) mencionó la distribución de Poisson bivariante.

Karlis y Ntzoufras (2003) utilizan una distribución de Poisson bivariante para modelar el número de goles anotados por cada equipo en partidos de fútbol. La distribución de Poisson bivariante es más general que una Poisson doble (dos Poisson independientes) y permite un mejor ajuste a los datos observados. Los parámetros se estiman mediante el algoritmo EM (expectation maximization). Además, proponen modelos inflados en la diagonal que mejoran la precisión en el ajuste de los empates. Los autores también incluyen como variables independientes si el equipo juega en casa o fuera y su rendimiento defensivo y ofensivo. Karlis y Ntzoufras mencionan los trabajos de Maher (1982) y Dixon y Coles (1997) como antecedentes de su modelo. A continuación, se explican los aspectos más relevantes

1.1 Maher (1982)

Maher (1982) presenta un artículo titulado "Modelling association football scores", en el que propone el uso de dos variables de Poisson independientes para modelar el número de goles de cada equipo en partidos de fútbol. Explora varias variables explicativas relacionadas con la fuerza defensiva y ofensiva de los equipos en función de si juegan en casa o fuera. Compara las frecuencias observadas y esperadas de su modelo mediante pruebas de bondad de ajuste y demuestra que, aunque existen algunas pequeñas diferencias sistemáticas, un modelo de Poisson independiente proporciona una descripción precisa de las puntuaciones en el fútbol. Finalmente, evita la condición de independencia sugiriendo el uso de una Poisson bivariante.

Un aspecto interesante de este artículo es que defiende el uso de la distribución de Poisson en lugar del modelo de la Binomial Negativa que había sido propuesto hasta entonces, haciendo que la media de la Poisson varíe en función de variables explicativas. Para ello, presenta las siguientes razones:

- La posesión es un factor importante en el fútbol, ya que cuando un equipo tiene el balón tiene la oportunidad de atacar y marcar goles.
- La probabilidad de que un ataque termine en gol es pequeña, pero el número de veces que un equipo tiene posesión del balón durante un partido es muy grande. Si esta probabilidad es constante y los ataques son independientes, el número de goles sigue una distribución Binomial y, en estas condiciones, la aproximación que mejor se ajusta es la Poisson.
- La media de esta Poisson variará en función de la calidad del equipo y, si se considera la distribución de todos los goles marcados por todos los equipos, se debería considerar la distribución Poisson con media variable

1.2 Dixon y Coles (1997)

Dixon y Coles parten del modelo propuesto por Maher (1982) pero plantean varias modificaciones con el objetivo de mejorar el ajuste del modelo en partidos con un bajo número de goles y también para permitir que los parámetros de habilidad en ataque y defensa de un equipo sean dinámicos y basados en su rendimiento reciente. Para la estimación de los parámetros, proponen una función de "pseudoverosimilitud" que maximizan mediante métodos numéricos.

1.3 Otros

Recientemente, Karlis y Ntzoufras (2011) han propuesto una estimación robusta de un modelo similar al suyo de 2003, asumiendo dos variables de Poisson independientes y sustituyendo la función de verosimilitud por una verosimilitud ponderada, cuyos pesos producen estimaciones más robustas. Asimismo, Koopman y Lit (2012) han publicado un

Discussion Paper en el que proponen un modelo de regresión de Poisson bivariante dinámico. Este modelo generaliza el modelo de Karlis y Ntzoufras (2003) para poder incluir variables que cambian con el tiempo.

En los últimos tiempos, algunos estudios han considerado el efecto del tiempo en la modelización del número de goles. Por ejemplo, Malcata, Hopkins y Richardson (2012) han utilizado un modelo de regresión lineal generalizado (Poisson) para analizar los goles anotados por un pequeño grupo de equipos durante varias temporadas, teniendo en cuenta su rendimiento a lo largo del tiempo, su calidad, la edad de los jugadores y la ventaja de jugar en casa. Por otro lado, Volf (2009) ha realizado una modelización de la secuencia de goles anotados en un partido utilizando procesos puntuales.

Por otro lado, Dyte y Clarke (2000) han utilizado variables de Poisson independientes con media variable para modelar el número de goles de cada equipo en un partido, y han considerado como variables explicativas la puntuación de la FIFA para cada equipo y el lugar del partido en el contexto de partidos de fútbol internacionales.

Greenhough (2002) evalúa la distribución de goles anotados por los equipos en casa, fuera de casa y el total de goles en cada partido sin tener en cuenta las variables dependientes. En este contexto, demuestran que las distribuciones de valores extremos pueden mejorar el ajuste de la distribución de Poisson o la distribución binomial negativa cuando son inadecuadas. En cuanto a la elección más conveniente de tipo de distribución (Poisson, Binomial negativa, valor extremo) para modelar el número de goles, Bittner, Nussbaumer, Janke y Weigel (2007, 2009) sugieren modificar el proceso de Bernouilli (marcar gol en cada instante del partido) para incluir un componente llamado autoafirmación (marcar un gol motiva al equipo que lo anota y desmotiva al contrario), lo que permite comprender el motivo de la modelización con un tipo de distribución u otro y obtener un buen ajuste a los datos.

Hay una serie de artículos que han utilizado un enfoque bayesiano para modelar el número de goles en partidos de fútbol. Un ejemplo es el trabajo de Rue y Salvensen (1998), que utilizaron un modelo lineal generalizado bayesiano dinámico para predecir el número de goles en cada partido. Este modelo tenía en cuenta la capacidad de ataque y defensa de cada equipo, así como el efecto de que el equipo local a veces subestima la capacidad del equipo visitante si es inferior. Otro ejemplo es el trabajo de Karlis y Ntzoufras (2009), que utilizaron una aproximación bayesiana para modelar la diferencia en el número de goles, es decir, el margen de victoria. Los autores argumentan que este enfoque tiene la ventaja de eliminar la correlación entre los dos equipos oponentes y de no tener que imponer que los goles marcados por cada equipo sigan una distribución de Poisson.

Otros estudios han enfocado su atención en la modelización de la diferencia de goles entre los equipos que compiten en un campeonato. Heuer y Rubner (2009) y Heuer, Müller y Rubner (2010) argumentan que la diferencia de goles es una medida más precisa que el número de puntos para determinar el equipo más fuerte en un campeonato. En sus trabajos, analizan la evolución de la diferencia de goles a lo largo del tiempo utilizando la

teoría de paseos aleatorios, una técnica comúnmente utilizada en campos como la física (de donde proceden estos autores).

Algunos estudios han abordado la modelización del resultado de un partido de fútbol en términos de ganar, perder o empatar. Dobson y Goddard (2000) utilizaron un modelo probit ordenado y métodos de Monte Carlo para predecir el resultado de un partido. Goddard y Asimakopoulos (2004) también aplicaron este modelo, considerando si el partido se jugaba en casa o en un campo visitante. Por otro lado, Brillinger (2006) evaluó las probabilidades de ganar, perder o empatar, modelizando directamente una variable latente que representaba la diferencia de calidad entre los equipos.

Harville (1997) utilizó modelos lineales mixtos para predecir la diferencia de puntos en un partido de fútbol. Además, señaló la importancia de tener en cuenta la ventaja de jugar en casa. Naim, Redner y Vazquez (2007) utilizaron procesos estocásticos para examinar los resultados de diferentes equipos y analizar la posición de cada equipo en diferentes ligas o torneos. Ben-Naim y Hengartner (2007) también se centraron en la posición de cada equipo y determinaron que el rango de cada equipo dependía del número de equipos y partidos jugados. Llegaron a la conclusión de que el formato de liga no es un método efectivo para determinar cuál es el mejor equipo.

Varios estudios han examinado el beneficio de jugar en casa. Jamieson (2010) publicó un metaanálisis sobre este tema. Saavedra et al. (2012) evaluaron el beneficio de jugar en casa en la liga española de fútbol desde 1928 hasta 2011 y encontraron que hay una ventaja significativa de jugar en casa. Lago-Peñas y Lago-Ballesteros (2011) descubrieron que, en su estudio, los equipos locales marcaban más goles y también disparaban más veces al arco que los equipos visitantes. Waters y Lovell (2002) examinaron el beneficio de jugar en casa desde un enfoque psicológico. Panaretos (2002) investigó otras variables relacionadas con el juego (disparos al arco, posesión del balón, etc.) que podrían influir en los goles y en los puntos obtenidos en la Liga de Campeones. Para concluir este análisis de la modelización estadística en datos de fútbol, cabe mencionar dos referencias que hacen una revisión de la estadística en el análisis de los resultados del fútbol: Emonet (2000) y Brillinger (2009).

En el contexto de estudiar el fútbol y su relación con la estadística, Emonet (2000) describe cómo la probabilidad de ganar un partido puede seguir una distribución de Poisson o una distribución binomial negativa. También evalúa el impacto de la ventaja de jugar en casa, el efecto de jugar en un campo artificial, el resultado de recibir tarjetas rojas y la influencia de las estrategias de juego. Brillinger (2009) presenta un informe técnico que menciona los trabajos más importantes en el análisis de datos del fútbol. Se menciona el uso de diferentes modelos estocásticos que incluyen distintas distribuciones específicas como la de Poisson bivariada, la exponencial, el valor extremo, la logística, la binomial negativa u ordinal. Algunos de estos modelos toman en cuenta los goles mientras que otros consideran los puntos de cada equipo.

1.4 Recapitulando

Karlis y Ntzoufras (2003) realizaron un estudio relevante en el uso de modelos de regresión avanzados en el fútbol. Se basaron en las investigaciones de Maher (1982) y Dixon y Coles (1997) al proponer una distribución de Poisson bivariada para analizar el número de goles anotados por cada equipo. También sugirieron modelos de inflado en la diagonal para mejorar la predicción de empates. Además, el modelo incluyó variables explicativas como el hecho de jugar en casa o en el extranjero y la habilidad ofensiva y defensiva de cada equipo.

Muchos otros estudios han examinado la modelización de los resultados del fútbol, tomando como variable dependiente el número de goles (por ejemplo, Rue y Salvensen, 1998; Skinner y Freeman, 2009; Volf, 2009 o Baio y Blangiardo, 2010). Otros han considerado modelos con la diferencia de goles como variable respuesta (Karlis y Ntzoufras, 2009, Heuer y Rubner, 2009), el resultado del partido en términos de ganar, perder o empatar (por ejemplo, Dobson y Goddard, 2000; Goddard y Asimkopoulos, 2004 y Brillinger, 2006) o la puntuación de los equipos (Harville, 1997, Naim et al., 2007 y Ben-Naim y Hengartner, 2007). Muchos de estos trabajos utilizan modelos donde el número de goles sigue una distribución de Poisson o una distribución binomial negativa. También se aplican modelos mixtos y diferentes tipos de procesos estocásticos que permiten considerar efectos temporales. Además, un número significativo de artículos utiliza un enfoque bayesiano.

2 Modelo Lineal Generalizado (GLM)

El modelo lineal generalizado es una técnica de análisis estadístico que se utiliza para examinar la relación entre el predictor lineal (definido más adelante) y la variable dependiente. A diferencia de la regresión lineal ordinaria, el modelo lineal generalizado permite que la variable dependiente siga una distribución diferente a la normal. Esto se logra mediante el uso de una función de enlace que establece la relación entre la variable predictora y la variable dependiente, y al permitir que la varianza de cada medición sea una función de su esperanza.

Los modelos lineales generalizados fueron desarrollados por John Nelder y Robert Wedderburn y lo publicaron en el libro *Generalized Linear Models,* publicado en 1983. Estos se presentan como una forma de unificar diferentes tipos de modelos estadísticos, tales como la regresión lineal, la regresión logística y la regresión de Poisson. Uno de los métodos más comunes para estimar los parámetros del modelo es mediante el uso de mínimos cuadrados iterativamente ponderados. Sin embargo, también se han desarrollado otros enfoques, incluyendo la estimación de máxima verosimilitud y métodos bayesianos, así como ajustes de mínimos cuadrados a respuestas de varianza estabilizadas.

En resumen, el modelo lineal generalizado es una herramienta útil para investigar la relación entre variables predictoras y dependientes, incluso cuando la distribución de la variable dependiente es diferente a la normal.

La regresión lineal ordinaria es una técnica de análisis estadístico que se utiliza para predecir el valor esperado de una variable desconocida (la variable de respuesta) a partir de un conjunto de valores observados (las variables predictoras). Este método supone que la relación entre la variable de respuesta y las variables predictoras es lineal, lo que significa que un cambio constante en una de las predictoras produce un cambio constante en la variable de respuesta. Esta técnica es apropiada cuando la variable de respuesta sigue una distribución normal, es decir, cuando puede variar de manera indefinida en cualquier dirección sin un valor fijo o cero, o más generalmente, cuando solo varía en una cantidad relativamente pequeña en comparación con la variabilidad de las variables predictoras, como por ejemplo la altura humana.

Sin embargo, estos supuestos no son aplicables a ciertos tipos de variables de respuesta. Por ejemplo, cuando se espera que la variable de respuesta siempre sea positiva y varíe en un amplio rango, los cambios constantes en las entradas pueden dar lugar a cambios en los resultados o salidas que varían de manera geométrica (es decir, exponencial) en lugar de manera constante. Como ejemplo, imagina que tienes un modelo de predicción que aprende a partir de ciertos datos (que podrían haber sido recogidos de grandes playas) que una disminución de 10 grados en la temperatura lleva a una disminución de 1000 personas que visitan la playa. Es poco probable que este modelo se pueda generalizar bien a playas de diferentes tamaños. Más específicamente, el problema es que si se utiliza este modelo para predecir la asistencia que habrá con una disminución de la temperatura de 10 grados en una playa que normalmente recibe 50 personas, se pronosticaría una asistencia con el imposible valor de -950 personas. Un modelo más realista debería pronosticar una tasa constante de incremento de la asistencia a la playa (es decir, un incremento de 10 grados provocaría una duplicación de la cantidad de visitantes, y una

disminución de 10 grados llevaría a una reducción a la mitad de la asistencia). Este tipo de modelo se conoce como modelo de respuesta exponencial (o modelo log-lineal, ya que se predice que el logaritmo de la respuesta variará linealmente). De esta manera, cuando se espera que la variable de respuesta siempre sea positiva y varíe en un amplio rango, un modelo de respuesta exponencial puede ser más apropiado que un modelo de respuesta lineal.

De manera similar, un modelo que predice la probabilidad de elegir entre dos opciones (una variable de Bernoulli) es aún menos adecuado como modelo de respuesta lineal, ya que las probabilidades están limitadas en ambos extremos (deben estar entre 0 y 1). Por ejemplo, imaginemos un modelo que predice la probabilidad de que una persona determinada vaya a la playa en función de la temperatura. Un modelo razonable podría predecir, por ejemplo, que un cambio de 10 grados hace que una persona tenga el doble de probabilidades de ir o no ir a la playa. Pero ¿qué significa "el doble" en términos de probabilidad? No puede significar literalmente duplicar el valor de la probabilidad (por ejemplo, 50% se convierte en 100%, 75% se convierte en 150%, etc.). Más bien, es la razón de oportunidades la que se duplica: de una razón de oportunidades 2:1 a una razón de oportunidades 4:1, a una razón de oportunidades 8:1, etc. Esta escala se conoce como log-odds. Así pues, cuando se trata de predecir la probabilidad de elegir entre dos opciones, un modelo logístico puede ser más adecuado que un modelo de respuesta lineal.

Los modelos lineales generalizados abarcan todas estas situaciones al permitir variables de respuesta que tienen distribuciones arbitrarias (en lugar de simplemente distribuciones normales) y al permitir que una función arbitraria de la variable de respuesta (la función de enlace) varíe linealmente con los valores predichos. Por ejemplo, en el caso anterior, el número pronosticado de asistentes a la playa se suele modelar con una distribución de Poisson y una función de enlace logarítmica, mientras que la probabilidad pronosticada de asistencia a la playa se suele modelar con una distribución de Bernoulli (o distribución binomial, dependiendo exactamente de cómo se exprese el problema) y una función de enlace log-odds (o logit).

Empezamos pues a definir conceptos:

Un modelo de regresión es un modelo estadístico que busca determinar la relación entre una variable dependiente con respecto a otras variables independientes. Este define el atributo en estudio como la suma entre una componente sistemática y una componente errática de la siguiente forma:

$$y_i = f(x_i) + \varepsilon_i \qquad i = 1, \ldots, n \quad indep.$$

Este modelo estadístico tiene dos particularidades:

- $Esperanza[y_i] = f(x)$
- $Varianza[y_i] = Varianza[\varepsilon_i]$

De esta manera, definimos en la regresión lineal simple:

$$f(x_i) = \beta_0 + \beta_1 x_i$$

y en el modelo de regresión lineal múltiple:

$$f(x_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}$$

El modelo Lineal General está formado por datos independientes $y_1, y_2, \cdots, y_n$ normalmente distribuidos

$$y_i \sim \mathrm{N}(\beta_0 + \beta_1 {x_i}^1 + \cdots + \beta_p {x_i}^p, \sigma^2)$$

Con predictor lineal $\beta' x_i$ y varianza constante $E[y] = X\beta,\ V[y] = \sigma^2 I$.

Así pues, el Modelo Lineal Generalizado (GLM) está formado por datos independientes de una distribución de la familia exponencial de McCullagh-Nelder. La familia de distribuciones exponenciales de McCullagh-Nelder incluye distribuciones como la distribución normal, la distribución de Poisson, la distribución binomial, la distribución de Bernoulli y otras. Estas distribuciones se caracterizan por tener una función de densidad de probabilidad que se puede expresar en términos de una función exponencial y modelizan $E[y]$ como una función no lineal de $X\beta$

Definimos GLM como conjunto de variables aleatorias independientes $y_1, y_2, \cdots, y_n$ con función de densidad, o función de probabilidad, que puede escribirse como:

$$p(y_i \mid \theta_i, \phi) = exp\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\}$$

donde:

- θ_i es el parámetro natural o canónico
- ϕ es un parámetro adicional de escala o dispersión
- $a_i(\cdot),\ b(\cdot),\ y\ c(\cdot)$ son funciones especificas

Y si ϕ es conocido, este es un modelo de la familia exponencial lineal mientras que si es desconocido es un modelo de dispersión exponencial.

Queremos modelizar $\mu_i = E[y_i]$ en términos del predictor lineal $\beta' x_i$ formado con un conjunto de p covariables

$$\beta' x_i = \beta_0 + \beta_1 {x_i}^1 + \cdots + \beta_p {x_i}^p$$

Tenemos n observaciones $y_1, y_2, \cdots, y_n$ aleatorias, independientes, de una variable respuesta. Donde suponemos, de momento, que estas variables tienen hasta segundo momento finito, es decir son variables que pueden tomar un número finito de valores. Por ejemplo, una variable de segundo orden finito podría ser "género" con los valores "masculino" y "femenino", o "estado civil" con los valores "soltero", "casado", "viudo" y "divorciado". Las variables de segundo orden finito se utilizan a menudo en el análisis de datos para representar ciertas características categóricas de una población o un conjunto de observaciones.

Como se supone que las observaciones tienen hasta segundo momento finito, lo que significa que la varianza de las observaciones es finita. Esto se refiere a que la variabilidad de las observaciones no es ilimitada, sino que está limitada a ciertos límites. En resumen,

las observaciones aleatorias casi siempre pertenecen a una familia de distribuciones exponenciales y que tienen una varianza finita.

Cada observación y_i corresponde a un vector $x_i = (x_{i0}, \cdots, x_{ip})$ conocido, es decir no aleatorio de variables predictoras. La coordenada x_{i0} es igual a 1 para unificar y simplificar notaciones.

2.1 Componente Aleatoria

La componente aleatoria de un GLM se refiere a una variable aleatoria Y con observaciones independientes $(y_1, y_2, \cdots, y_n)$.

A menudo, las observaciones de Y son binarias y se clasifican como éxito o fracaso. Sin embargo, también es posible que cada Y_i indique el número de éxitos entre un número fijo de pruebas y sea modelado como una distribución binomial. En otras ocasiones, cada observación es un recuento y se le puede asignar a Y una distribución de Poisson o una distribución binomial negativa. Por último, si las observaciones son continuas, se puede suponer que Y sigue una distribución normal.

Todos estos modelos se pueden incluir dentro de la llamada familia exponencial. La familia exponencial de distribuciones sobredispersas es una extensión de la familia exponencial y el modelo de dispersión exponencial de distribuciones, e incluye esas distribuciones de probabilidad parametrizadas por θ y ϕ cuyas funciones de densidad f (o funciones de masa de probabilidad en el caso de una distribución discreta) se pueden expresar de la siguiente manera:

$$f_Y(y \mid \theta, \phi) = h(y, \phi) \cdot exp\{\frac{A(\theta)^T T(y) - b(\theta)}{a(\phi)}\}$$

El parámetro de dispersión, T, generalmente es conocido y está relacionado con la varianza de la distribución. Las funciones $h(y, \phi)$, $A(\theta)$, $T(y)$, $b(\theta)$ y $a(\phi)$ son conocidas. Muchas distribuciones comunes se encuentran en esta familia, incluyendo la binomial, multinomial y binomial normal, exponencial, gamma, Poisson, Bernoulli y (para un número fijo de ensayos).

Para el escalar Y y θ, esto se reduce a:

$$f_Y(y \mid \theta, \phi) = h(y, \phi) \cdot exp\{\frac{A(\theta)T(y) - b(\theta)}{a(\phi)}\}$$

θ está relacionado con la media de la distribución. Si $A(\theta)$ es la función de identidad, se dice que la distribución está en forma canónica (o forma natural). Tenemos en cuenta que cualquier distribución se puede convertir a forma canónica reescribiendo θ como θ' y luego aplicando la transformación $\theta = A(\theta')$. Siempre es posible convertir $b(\theta)$ en términos de la nueva parametrización, incluso si $A(\theta')$ no es una función uno a uno. Si, además, $T(y)$ es la identidad y ϕ es conocido, entonces θ se llama el parámetro canónico (o parámetro natural) y está relacionado con la media a través de

$$\mu = E(Y) = \nabla b(\theta)$$

Para el escalar Y y θ, esto se reduce a

$$\mu = E(Y) = b'(\theta)$$

Bajo este escenario, la varianza de la distribución puede mostrarse como

$$Var(Y) = \nabla^2 b(\theta) a(\phi)$$

Para el escalar Y y θ, esto se reduce a

$$Var(Y) = A''(\theta) a(\phi)$$

2.2 Componente Sistemática

La componente sistemática de un GLM especifica las variables explicativas, que entran en forma de efectos fijos en un modelo lineal, es decir, las variables x_j se relacionan mediante

$$\alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Esta combinación lineal de variables explicativas se denomina predictor lineal.

El predictor lineal es una medida que incluye información sobre las variables independientes en un modelo. El símbolo η representa al predictor lineal y está vinculado con el valor esperado de los datos a través de una función de vinculación. η se expresa como combinaciones lineales (por lo tanto, "lineales") de parámetros desconocidos llamados β. Los coeficientes de la combinación lineal se representan como una matriz de variables independientes llamada X . Por lo tanto, η puede ser expresado como $\eta = X\beta$.

Alternativamente, se puede expresar como un vector $(\eta_1, \eta_2, \cdots, \eta_N)$ tal que

$$\eta_i = \sum_j \beta_j \cdot x_{ij}$$

donde x_{ij} es el valor j-ésimo predictor en el i-ésimo individuo, e $i = 1, \ldots, N$. El término independiente α se obtendría con esta notación haciendo que todos los x_{ij} sean igual a 1 para todos los i.

En cualquier caso, se pueden considerar variables que estén basadas en otras variables como $x_3 = x_1 x_2$ o $x_3 = {x_2}^2$, para modelizar interacciones entre variables o efectos curvilíneos de x_2.

2.3 Función de enlace

Las esperanzas μ_i de las y_i se relacionan con los predictores lineales por la ecuación:

$$\mu_i \equiv E(y_i) = F(\eta_i) = F(x_i \cdot \beta)$$

F forma parte del modelo. Su dominio es toda la recta real $\mathbb{R}$, conjunto de los valores posibles de los η_i.

Su recorrido es el intervalo I de valores posibles de los μ_i. Se exige también que F sea biyectiva. Su función inversa $g = F^{-1}$, es decir, que $F \circ g = Id_I$, $g \circ F = Id_{\mathbb{R}}$, es la función de enlace (link) del modelo.

El valor esperado de Y se denota como $\mu = E(Y)$, entonces la función "link" o enlace especifica una función $g(\cdot)$ que relaciona μ con el predictor lineal como

$$g(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Así, la función link $g(\cdot)$ relaciona las componentes aleatoria y sistemáticas.

De este modo, para $i = 1, \ldots, N$

$$\mu_i = E(Y_i)$$

$$\eta_i = g(\mu_i) = \sum_j \beta_j \cdot x_{ij}$$

La función $g(\cdot)$ más simple es $g(\mu) = \mu$, esto es, la identidad que da lugar al modelo de regresión lineal clásico

$$\mu = E(Y) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

La función de enlace se utiliza para establecer la relación entre un predictor lineal y la media de la distribución. Existen muchas opciones comunes para elegir, y la elección se basa en diversos factores. Siempre hay una función de enlace establecida que se deriva del exponencial de la función de densidad de la respuesta. Sin embargo, en algunos casos es conveniente hacer que el rango de la media de la función de distribución coincida con el dominio de la función de enlace, o utilizar una función de enlace no establecida con propósitos algorítmicos, como la regresión probit Bayesiana.

Los modelos de regresión lineal típicos para respuestas continuas son un caso particular de los modelos de regresión generalizada (GLM). Estos modelos amplían la regresión ordinaria de dos maneras: permitiendo que, Y tenga distribuciones diferentes a la normal y, por otro lado, incluyendo distintas funciones de enlace de la media. Esto resulta muy útil para datos categóricos.

Los modelos GLM permiten la unificación de una amplia variedad de métodos estadísticos, como la regresión, los modelos ANOVA y los modelos de datos categóricos. En realidad, se utiliza el mismo algoritmo para obtener los estimadores de máxima

verosimilitud en todos los casos. Este algoritmo es la base del procedimiento GENMOD de SAS y de la función glm de Python.

2.4 Estimación de las $\boldsymbol{\beta}$

Las respuestas y_i de los GLM pueden ser variables discretas o absolutamente continuas. Imponemos la condición que cada y de un mismo modelo pertenece a una misma clase de funciones dentro de la familia exponencial de McCallugah-Nelder:

$$\mathcal{L}(y\,;\,\theta,\phi)\;=\;exp\{\frac{A(\theta)T(y)-\;b(\theta)}{a(\phi)}\;+\;c(y,\phi)\,\}$$

Para un MLG, en el que suponemos que las n observaciones y_i son independientes, la función de verosimilitud del modelo es el producto

$$\mathcal{L}(y\,;\,\beta)\;=\;\prod_{i\,=\,1}^{n}\mathcal{L}(y_i\,,\beta)$$

de n factores de la forma.

Para los logaritmos,

$$\ell(y\,;\,\beta)\;=\;\sum_{i=1}^{n}\ell\,(y_i\,,\beta)$$

En la notación mostramos solamente el parámetro β, que se relaciona con los μ_i mediante la función de enlace. En cambio, no explicamos el parámetro ϕ, que suponemos igual para las n observaciones.

Esta hipótesis es la extensión natural de la *homoscedasticidad* en el modelo lineal ordinario. También omitimos en la notación todas las letras no estrictamente necesarias, para no recargar.

Derivando ℓ con respecto a cada uno de los $p+1$ parámetros, teniendo en cuenta la regla de la cadena, obtenemos las $p+1$ ecuaciones de verosimilitud,

$$\frac{\partial\ell}{\partial\beta_j}=\sum_{i=1}^{n}\frac{\partial\ell}{\partial\mu_i}\frac{\partial\mu_i}{\partial\beta_j}\;=\;0,\qquad j=0,\ldots,p.$$

Por las propiedades de la familia exponencial, si F no es la identidad,

$$\frac{\partial\ell}{\partial\beta_j}=\sum_{i=1}^{n}\frac{y_i\;-\;\mu_i}{V_i}F'(x_i\cdot\beta)x_{ij}\;=\;0,\qquad j=0,\ldots,p.$$

Si la función de enlace no es la identidad, derivando la identidad $\mu_i = (F \circ g)(\mu_i)$,

$$1 = F'(g(\mu_i))g'(\mu_i) = F'(\eta_i)g'(\mu_i) = F'(x_i \cdot \beta)g'(\mu_i).$$

Sustituyendo,

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{V_i g'(\mu_i)} x_{ij} = 0, \qquad j = 0, \ldots, p.$$

Puesto que

$$V_i = \frac{a(\phi)}{h'(\mu_i)},$$

la ecuación de verosimilitud se simplifica si la función de enlace g (que forma parte del modelo, o sea que es potestativa) coincide con la función h (que depende solo de la distribución de probabilidad de las respuestas y_i).

La función de enlace $g = h$ se llama enlace natural o canónico de la distribución de probabilidad de las respuestas.

La elección de la función de enlace natural equivale a plantear el modelo como "hacemos el predictor lineal igual al parámetro natural θ_i de la distribución",

$$x_i \cdot \beta = \eta_i = g(\mu_i) = h(\mu_i) = \theta_i .$$

2.4.1 Caso particular

Si la función de enlace es la identidad, cosa que ocurre, por ejemplo, en el modelo lineal,

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - x_i \cdot \beta}{V_i} x_{ij} = 0, \qquad j = 0, \ldots, p.$$

En notación matricial, resulta un sistema de ecuaciones normales ponderadas,

$$X' \cdot V \cdot (y - X \cdot \beta) = 0 ,$$

Siendo V la matriz diagonal $V = diag(\frac{1}{V_1}, \ldots, \frac{1}{V_n})$.

En el modelo lineal normal homoscedástico, todas las V_i serían iguales a $V = \sigma^2$. En un GLM general esto no ocurre, aun así, suponemos igual ϕ para las n observaciones.

Si h es la identidad, como en el modelo lineal normal (heteroscedástico), V no depende de β y podemos resolver las ecuaciones normales ponderadas

$$X' \cdot V \cdot y = X' \cdot V \cdot X \cdot \beta.$$

La solución de las ecuaciones normales ponderadas da la estimación, por mínimos cuadrados ponderados (WLS),

$$\hat{\beta} = (X' \cdot V \cdot X)^{-1} \cdot X' \cdot V \cdot y,$$

si $(X' \cdot V \cdot X)$ es invertible.

Más en particular, en el modelo lineal normal homoscedástico, $V = \sigma^2 \cdot I$, recuperamos la estimación OLS: $\hat{\beta} = (X' \cdot X)^{-1} \cdot X' \cdot y$.

2.4.2 Caso general

En el caso general, definimos nuevas "observaciones" $z_i, i = 1, \ldots, n$, por:

$$z_i := g(\mu_i) + g'(\mu_i)(y_i - \mu_i) = \eta_i + g'(\mu_i)(y_i - \mu_i).$$

La motivación es un intento de acercar el caso general al caso particular para aplicar el mismo método de solución, mediante una linealización aproximada

$$g(y) \approx g(\mu) + g'(\mu)(y - \mu).$$

Esto es, sin embargo, una definición exacta (no aproximada) de las n nuevas variables z_i.

Sustituyendo $y_i - \mu_i = \frac{z_i - \eta_i}{g'(\mu_i)}$ en las ecuaciones de verosimilitud, obtenemos:

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} w_i (z_i - \eta_i)\, x_{ij} = \sum_{i=1}^{n} w_i (z_i - x_i \cdot \beta)\, x_{ij}, \qquad j = 0, \ldots, p$$

siendo: $w_i = \frac{1}{V_i (g'(\mu_i))^2}$, $i = 1, \ldots, n$.

Estas ecuaciones tienen idéntica forma funcional que las ecuaciones normales del caso lineal ponderado, con el w_i en lugar de $1/V_i$. En forma matricial,

$$X' \cdot W \cdot (z - X \cdot \beta) = 0,$$

siendo

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}, \qquad W = diag(w_1, \ldots, w_n).$$

La solución WLS, $\hat{\beta} = (X' \cdot W \cdot X)^{-1} \cdot X' \cdot W \cdot z$ no es practicable en principio, puesto que z no se puede calcular a partir de y al depender su relación de los parámetros β desconocidos y que W es función también de estos parámetros.

Es posible un cálculo iterativo, de aproximaciones sucesivas, partiendo de un valor inicial, $\beta[0]$, de los parámetros β.

Las funciones $F(\cdot)$, $g(\cdot)$, se aplican elemento a elemento a cada componente de los vectores.

A partir de un valor inicial $\beta[0]$,

1. $\eta[0] = X \cdot \beta[0]$,
2. $\mu[0] = F(\eta[0])$,
3. $z[0] = \eta[0] + g'(\mu[0])(y - \mu[0])$,
4. $W[0] = 1/V[0]\big(g'(\mu[0])\big)^2$,
5. La solución WLS, $\beta[1] = (X' \cdot W[0] \cdot X)^{-1} \cdot X' \cdot W[0] \cdot z[0]$.

Se itera hasta convergencia (o decidir que no se va a alcanzar).

2.5 Planteamiento de nuestro modelo

Para aplicar un modelo GLM para predecir resultados deportivos, lo primero que debemos hacer es definir su variable respuesta. En el caso de los resultados deportivos, esto podría ser el resultado del partido (por ejemplo, gana el equipo local, empate o gana el equipo visitante), pero en nuestro caso será el número de goles que anota cada equipo.

A continuación, debemos seleccionar las variables predictivas que deseamos utilizar en nuestro modelo. Esto podría incluir cualquier número de características que puedan afectar el resultado del partido, como el rendimiento del equipo en partidos anteriores, el nivel de forma de los jugadores, la calidad del equipo visitante, etc. En nuestro caso definiremos para cada equipo unas fuerzas de ataque y de defensa tanto en campo rival como en campo visitante calculadas con sus estadísticas en un periodo acotado de tiempo previo al partido. Detallaremos todo esto en los próximos capítulos

Una vez que hayamos seleccionado las variables predictivas, utilizaremos Python para el análisis estadístico para construir y ajustar su modelo GLM. En general, el proceso de construcción de un modelo GLM incluye las siguientes etapas:

1. Limpiar y preparar los datos: asegurándonos de que datos sean completos y consistentes, y eliminando cualquier valor atípico o outlier que pueda afectar el rendimiento del modelo.

2. Definir y seleccionar tanto las variables dependientes como las variables independientes.

3. Especificar la función de enlace: elegir la función de enlace que mejor se ajusta a los datos y a la distribución de respuesta seleccionada. La función de enlace determina cómo se relacionan las variables predictivas con la variable respuesta.

4. Ajustar el modelo: usar un algoritmo de optimización (como el método de máxima verosimilitud) para ajustar los parámetros del modelo y minimizar el error de predicción.

Una vez que hayamos ajustado el modelo GLM, podemos utilizarlo para hacer predicciones sobre resultados deportivos futuros. Para hacer esto, simplemente proporcionaremos los valores de las variables predictivas para el partido que deseamos predecir y el modelo nos devolverá una predicción.

3 Los Datos

Para hacer un estudio de las estadísticas deportivas necesitamos un gran número de datos históricos los cuales serán estudiados y analizados posteriormente buscando patrones o tendencias.

Para ello, extraeremos los datos de una web [1] en la cual la cual podemos encontrar los resultados deportivos de las principales ligas de futbol profesional de cada temporada desde el año 1993 en adelante.

Nosotros nos centraremos únicamente en los resultados de la liga española (actualmente conocida como "La Liga Santander")

La herramienta utilizada para este trabajo será Python, utilizando principalmente las librerías de pandas, numpy, seaborn y matplotlib y scikit-learn entre otras.

3.1 Extracción y análisis de resultados

Para iniciar nuestro estudio con Python, importamos varias librerías que iremos necesitando a lo largo del trabajo entre las cuales destaco la librería 'pandas' que es librería de código abierto dentro de los desarrolladores de Python, y sobre todo dentro del ámbito de Data Science y Machine Learning, ya que ofrece unas estructuras muy poderosas y flexibles que facilitan la manipulación y tratamiento de datos.

3.1.1 Extracción y limpieza

Los datos que necesitamos para nuestro análisis los extraemos de la web que hemos mencionado anteriormente. En esta web, los enlaces de la web son archivos .csv.

Esta web tiene todos los partidos disputados desde el año 1993, incluyendo mucha información de cada partido. Al importar cualquiera de estas temporadas, obtenemos una tabla con tantas filas como partidos se disputaron y para cada uno de estos partidos la siguiente información organizada en 105 columnas: ['Div', 'Date', 'Time', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR', 'HTHG', 'HTAG', 'HTR', 'HS', 'AS', 'HST', 'AST', 'HF', 'AF', 'HC', 'AC', 'HY', 'AY', 'HR', 'AR' , … , 'MaxCAHH', 'MaxCAHA', 'AvgCAHH', 'AvgCAHA'].

El significado de cada una de estas columnas lo podemos encontrar en otra web [2], entre las cuales hay información sobre el partido en si e incluso información sobre diversas cuotas de varias casas de apuestas previas a cada uno de esos partidos, pero nos limitaremos a seleccionar las columnas que creo convenientes para nuestro trabajo.

Así pues, importamos los datos de todas las temporadas creando un único dataframe con la información de 11.352 partidos seleccionando las columnas:

- Temporada: Temporada en la que se disputa el encuentro.
- Date: Fecha del encuentro.
- HomeTeam: Equipo que juega de local.
- AwayTeam: Equipo que juega de visitante.
- FTHG: Goles anotados por el equipo local. (Full Time Home Goals).
- FTAG: Goles anotados por el visitante. (Full Time Away Goals).
- FTR: Quien gana el Partido (H: equipo local, D: empate, A: equipo visitante). (Full Time Result).

Un dataframe es una estructura de datos en forma de tabla, utilizada en el lenguaje de programación Python para el análisis de datos. De ahora en adelante hablaremos de dataframes, pero podemos entenderlos como las tablas que utilizamos en las hojas de cálculo.

Adjunto una muestra de cómo queda el dataframe:

	Temporada	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR
0	22/23	2023-01-16	Cadiz	Elche	1	1	D
1	22/23	2023-01-15	Almeria	Ath Madrid	1	1	D
2	22/23	2023-01-15	Getafe	Espanol	1	2	A
3	22/23	2023-01-14	Osasuna	Mallorca	1	0	H
4	22/23	2023-01-14	Girona	Sevilla	2	1	H
...	...	...	...	...	...	...	...
11347	93/94	1993-09-05	Osasuna	Real Madrid	1	4	A
11348	93/94	1993-09-05	La Coruna	Celta	0	0	D
11349	93/94	1993-09-05	Barcelona	Sociedad	3	0	H
11350	93/94	1993-09-05	Ath Madrid	Logrones	1	0	H
11351	93/94	1993-09-05	Ath Bilbao	Albacete	4	1	H

Al examinar la tabla para ver que no contenga datos nulos, nos damos cuenta de que hay ciertos nombres de equipos que han ido cambiando mínimamente a lo largo de los años. Es por ello por lo que, mediante métodos dentro de la librería de pandas, podemos unificarlos. (Villareal → Villarreal, Vallecano → Rayo Vallecano)

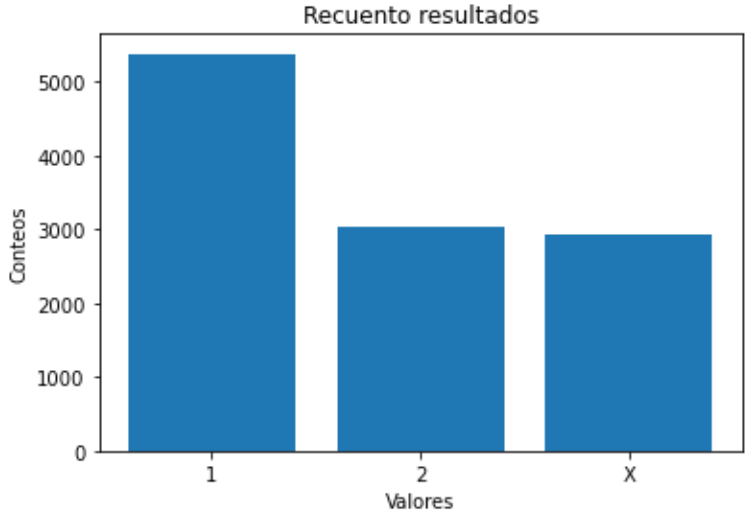
Como esta será la tabla con la que trabajaremos de ahora en adelante, optamos también por renombrar algunas columnas para tener la tabla con nombres de variables con las que me sienta cómodo para trabajar con ellas. Así pues, tras algunos cambios estéticos, obtenemos la siguiente tabla:

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR
0	22/23	2023-01-16	Cadiz	Elche	1	1	D
1	22/23	2023-01-15	Almeria	Ath Madrid	1	1	D
2	22/23	2023-01-15	Getafe	Espanol	1	2	A
3	22/23	2023-01-14	Osasuna	Mallorca	1	0	H
4	22/23	2023-01-14	Girona	Sevilla	2	1	H
...	...	...	...	...	...	...	...
11347	93/94	1993-09-05	Osasuna	Real Madrid	1	4	A
11348	93/94	1993-09-05	La Coruna	Celta	0	0	D
11349	93/94	1993-09-05	Barcelona	Sociedad	3	0	H
11350	93/94	1993-09-05	Ath Madrid	Logrones	1	0	H
11351	93/94	1993-09-05	Ath Bilbao	Albacete	4	1	H

3.1.2 Visualización de resultados

Para la parte de análisis, utilizaremos la librería Matplotlib la cual proporciona una interfaz para trazar una variedad de tipos de gráficos en varios formatos. Es muy útil para la exploración de datos y la visualización de resultados en el campo de la ciencia de datos y la ingeniería. Es una de las bibliotecas más utilizadas para hacer gráficos en Python.

Empezaremos visualizando como están distribuidos los distintos tipos de resultados.



En este gráfico vemos como el 26% de los partidos terminan en empate, el 27% termina en victoria del equipo visitante mientras que el 47% de los partidos disputados los gana el equipo en campo local. Este gráfico es una muestra de que el hecho de jugar como local es un factor bastante determinante.

Hay varias razones por las que el equipo local suele tener una ventaja en los partidos de fútbol. Una de las principales razones es el apoyo del público. El apoyo de los aficionados locales puede dar un impulso adicional a los jugadores del equipo local, lo que les permite jugar con más confianza y energía. Además, el equipo local está acostumbrado a jugar en su campo y conoce mejor las condiciones del terreno de juego.

Es importante tener en cuenta que estas ventajas no son determinantes para el resultado final de un partido de fútbol y que un equipo visitante también puede ganar.

Analizando ahora los goles,

	FTHG	FTAG
mean	1,555457	1,110297
std	1,310228	1,114533

podemos ver como el promedio de goles anotados en por el equipo local es de 1'55, siendo este promedio bastante superior que el promedio de goles anotados por el equipo visitante. Este estudio reafirma el hecho que hemos visto previamente de que el equipo local parte con ventaja.

Vemos también la desviación estándar del número de goles anotados por el equipo local y visitante. La desviación estándar nos dice cuán dispersos están los datos respecto a la media o el valor esperado. Una desviación estándar pequeña indica que los datos están cerca del valor esperado, mientras que una desviación estándar grande indica que los datos están más dispersos.

Si la desviación estándar es muy similar a la media, significa que los datos están muy concentrados alrededor de la media. Esto sugiere que los datos tienen una distribución relativamente estrecha y con poca variabilidad.

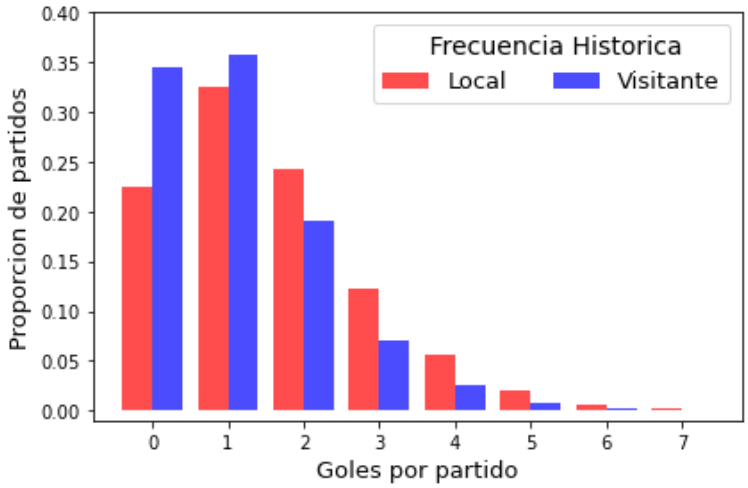
En una distribución normal, la desviación estándar es aproximadamente igual a la mitad de la media, por lo que, si la desviación estándar es muy similar a la media, esto sugiere que la distribución de los datos es similar a una distribución normal.

Sin embargo, también es posible que los datos tengan una distribución no normal, pero con una varianza muy pequeña en comparación con la media. En este caso, la desviación estándar sería similar a la media, pero los datos no se distribuirían de manera normal.

Es importante analizar los datos en conjunto con otras medidas estadísticas y gráficos para tener una comprensión completa de la distribución de los datos.

Examinemos el número de goles que se anotan en cada partido, tanto el equipo local como el visitante:

	0	1	2	3	4	5	6	7
Local	0.224630	0.325846	0.243481	0.122533	0.055233	0.020085	0.005726	0.001762
Visitante	0.344521	0.357646	0.191244	0.071001	0.025018	0.007840	0.002114	0.000088



La distribución de Poisson es una distribución de probabilidad discreta que expresa, a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto período de tiempo. Concretamente, se especializa en la probabilidad de ocurrencia de sucesos con probabilidades muy pequeñas, o sucesos raros.

$$X \sim Poisson(\lambda)$$

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad para\ k = 0, 1, 2, \ldots$$

En una distribución de Poisson la media es igual a la varianza, es decir, la desviación estándar es igual a la raíz cuadrada de la media.

La distribución de Poisson se utiliza para modelar un número de eventos en un período de tiempo o en un espacio dado, siempre y cuando se cumplan ciertas condiciones:

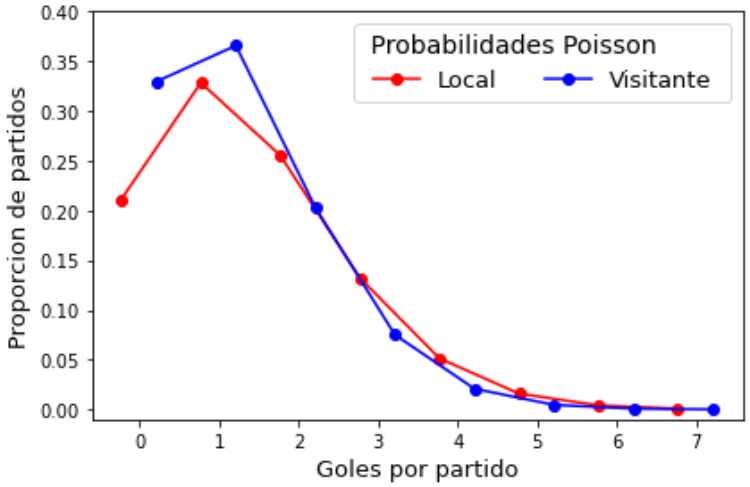
- Los eventos son independientes
- La tasa de eventos es constante
- El número de eventos es grande

Esta es útil para modelar eventos que ocurren raramente, pero con una tasa constante. Veamos si nos encaja.

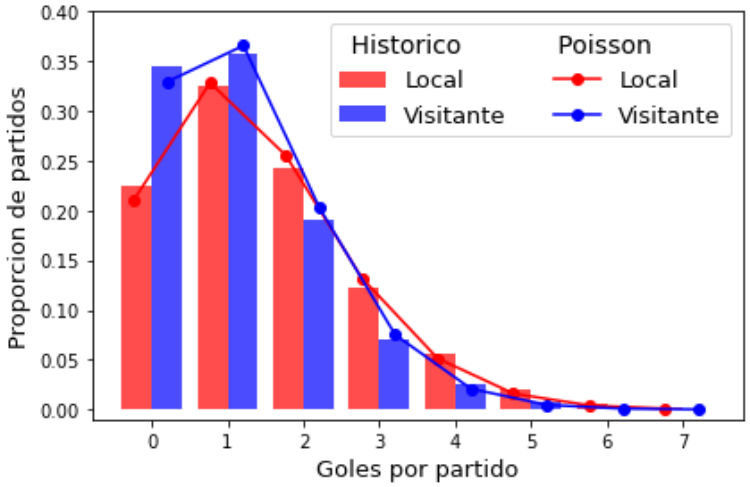
En nuestro caso el equipo local tiene una varianza de $1{,}310^2 = 1{,}716$, un valor algo similar a la media $1{,}55$. De la misma manera, el equipo visitante tiene una varianza de $1{,}114^2 = 1{,}242$ similar también a su media de $1{,}11$.

Esta distribución describe la probabilidad de que suceda un suceso en un intervalo de tiempo determinado (90 minutos) a través de una cota promedia de ocurrencia. Esta cota será el promedio de goles anotados tanto en campo local como en campo visitante. Finalmente, como el número de eventos es independiente del tiempo calculamos las probabilidades de que tanto el equipo local como el equipo visitante anoten exactamente k goles en un partido.

	0	1	2	3	4	5	6	7
Local	0.211326	0.328475	0.255283	0.132267	0.051397	0.015978	0.004139	0.000919
Visitante	0.329580	0.365813	0.203014	0.075111	0.020842	0.004627	0.000856	0.000136



Nos damos cuenta de que, en los dos gráficos anteriores, el primero de los cuales muestra el promedio de partidos en los que se han anotado exactamente k goles y el segundo grafico que muestra la probabilidad de anotar k goles según la distribución teórica de Poisson tienen una forma muy similar. De hecho, veámoslos uno encima del otro:



Efectivamente siguen una forma muy similar tanto para el equipo local como para el equipo visitante.

Tenemos pues un primer punto de partida con el que guiaremos nuestro modelo.

3.2 Definición de las variables explicativas

Trataremos ahora de definir las variables explicativas calculándolas a partir de las estadísticas de cada equipo previas al día del partido. Estas variables tendrán en cuenta el

rendimiento de un equipo tanto en campo rival como en campo visitante, definiendo así para cada equipo cuatro variables:

1. FAL: Fuerza de ataque en campo local.
2. FDL: Fuerza defensiva en campo local.
3. FAV: Fuerza de ataque en campo visitante.
4. FDV: Fuerza defensiva en campo visitante.

Utilizaremos la base de datos que hemos preparado previamente con todos los partidos desde 1993 y la información que hemos considerado relevante de estos.

3.2.1 Funciones auxiliares

Para definir unas cuotas de ataque y defensa lo más realistas posibles para un equipo de cara a un partido concreto un día determinado, escogeremos las estadísticas de este equipo para estudiar su rendimiento únicamente de los partidos que ha disputado en los 365 días previos hasta la fecha del partido.

Para ello, definimos una función

```
>def resultados_n_años_antes(resultados, fecha, n=1):
>       fecha_hace_n_años = (fecha - timedelta(days=365*n))
>       return resultados[(resultados.Date >= fecha_hace_n_años) &
        (resultados.Date <= fecha)]
```

la cual recibe una base de datos en forma de dataframe y una fecha concreta y nos devuelve otro dataframe con todos los partidos que se han disputado en los 365 días previos a esta fecha. En caso de que quisiéramos los datos de n años previos a la fecha, también podemos darle a la función un valor n concreto. En caso de no especificar cuantos años queremos, por defecto la función nos devolverá un año.

Definimos también la función:

```
>def historico_1_equipo(team, resultados):
>       historico = resultados[(resultados.TeamA == team) | (resultados.TeamB
        == team)]
>       team_local = historico[historico.TeamA==team]
>       team_visit = historico[historico.TeamB==team]
>       df = pd.DataFrame()
>       df['Equipo'] = [team]
>       return team_local, team_visit, df
```

Esta, recibe el nombre de un equipo y una base de datos en forma de dataframe y nos devuelve dos dataframes. El primero de ellos está formado por todos los partidos en los que el equipo que hemos introducido juega como equipo local y el segundo está formado por todos los partidos en los que juega de visitante.

3.2.1.1 Fuerza de ataque (FA)

Este valor será calculado para un equipo determinado analizando su rendimiento ofensivo jugando en campo local (FAL) en relación con el resto de los equipos y de la misma manera calcularemos también el rendimiento del equipo jugando en campo visitante (FAV).

Para el cálculo de estos valores utilizaremos com bases de datos las que obtendremos gracias a las funciones que hemos definido previamente consiguiendo así poder calcular, a medida de los posible, el rendimiento de un equipo en una fecha determinada. De esta manera, un equipo no tendrá los mismos valores de FAL y FAV cada jornada pues las bases de datos irán cambiando constantemente. Así pues, estas se calculan de la siguiente manera:

$$\text{FAL} = \frac{\text{promedio de goles anotados por el equipo jugando de local}}{\text{promedio de goles anotados por todos los equipos que juegan de locales}}$$

$$\text{FAV} = \frac{\text{promedio de goles anotados por el equipo jugando de visitante}}{\text{promedio de goles anotados por todos los equipos que juegan de visitantes}}$$

```
>def fuerza_ataque(equipo, resultados):
>       team_local = historico_1_equipo(equipo, resultados)[0]
>       team_visit = historico_1_equipo(equipo, resultados)[1]
>       promedio_goles_de_local = team_local.FTHG.mean()
>       promedio_goles_total_locales = resultados.FTHG.mean()
>       promedio_goles_de_visitante = team_visit.FTAG.mean()
>       promedio_goles_total_visitantes = resultados.FTAG.mean()
>       FAL = promedio_goles_de_local / promedio_goles_total_locales
>       FAV = promedio_goles_de_visitante / promedio_goles_total_visitantes
>       return FAL, FAV
```

3.2.1.2 Fuerza defensiva (FD)

Estos valores serán calculados de una manera muy similar a la fuerza de ataque, pero en lugar de contabilizar los goles marcados, contabilizaremos el número de goles que le han anotado al equipo (goles encajados). Dicho lo cual, definimos:

$$\text{FDL} = \frac{\text{promedio de goles encajado por el equipo jugando de local}}{\text{promedio de goles encjados por todos los equipos que juegan de locales}}$$

$$\text{FDV} = \frac{\text{promedio de goles encajado por el equipo jugando de visitante}}{\text{promedio de goles encjados por todos los equipos que juegan de visitantes}}$$

```
>def fuerza_defensa(equipo, resultados):
>       team_local = historico_1_equipo(equipo, resultados)[0]
>       team_visit = historico_1_equipo(equipo, resultados)[1]
>       promedio_encajados_de_local = team_local.FTAG.mean()
>       promedio_encajados_total_locales = resultados.FTAG.mean()
>       promedio_encajados_de_visitante = team_visit.FTHG.mean()
>       promedio_encajados_total_visitantes = resultados.FTHG.mean()
>       FDL = promedio_encajados_de_local / promedio_encajados_total_locales
>       FDV = promedio_encajados_de_visitante /
        promedio_encajados_total_visitantes
>       return FDL, FDV
```

3.2.1.3 Unificación de funciones

Finalmente creamos una última función auxiliar la cual nos permite unificar todas las anteriores. Esta función nos pide un dataframe con los partidos de los cuales queremos calcular las fuerzas atacantes y defensivas tanto del equipo local como del equipo visitante y una base de datos.

Nos pide también la base de datos y también un valor opcional que, en caso de no darle nada, nos lo detecta como False. Mediante este último parámetro podemos cambiar que la función en vez de escoger la fecha de cada uno de los partidos para hacer los respectivos cálculos escoja únicamente una fecha en concreto (la primera fecha que encuentra en el dataframe). Mas adelante veremos para qué es útil esta acción.

Finalmente, esta función nos devuelve el mismo dataframe que le hemos proporcionado con partidos incluyendo nuevas columnas con las cuotas para cada equipo.

```
>def definir_varibales_explicativas(df, resultados, pasado=False):
>       FAL_A, FDL_A, FAV_B, FDV_B = [],[],[],[]
>       for i in range(len(df)):
>               fecha = df.Date.iloc[i]
>               if pasado:
>                       fecha = df.Date.iloc[0]
>               FAL, FAV = fuerza_ataque(df.TeamA.iloc[i],
                resultados_n_años_antes(resultados, fecha))
>               FDL, FDV = fuerza_defensa(df.TeamA.iloc[i],
                resultados_n_años_antes(resultados, fecha))
>               FAL_A.append(FAL)
>               FDL_A.append(FDL)
>               FAL, FAV = fuerza_ataque(df.TeamB.iloc[i],
                resultados_n_años_antes(resultados,fecha))
>               FDL, FDV = fuerza_defensa(df.TeamB.iloc[i],
                resultados_n_años_antes(resultados, fecha))
>               FAV_B.append(FAV)
>               FDV_B.append(FDV)
>       df['FAL_A'], df['FDL_A'] = FAL_A, FDL_A
>       df['FAV_B'], df['FDV_B'] = FAV_B, FDV_B
>       return df
```

3.2.2 Veamos un ejemplo

Una vez definidas estas funciones auxiliares, podemos enfrentarnos ahora a poner un ejemplo concreto.

Estudiemos el caso que queremos analizar los últimos 10 partidos que se han disputado entre el FC Barcelona y el Valencia CF con el Barcelona como equipo local.

```
>equipoA = 'Barcelona'
>equipoB = 'Valencia'
>df = historico_1_equipo(equipoA, resultados)[0]
>df[df.TeamB == equipoB].head(10)
```

Creamos un dataframe con todos los partidos en los que el Barcelona ha jugado como equipo local y finalmente filtramos este para que únicamente nos muestre los 10 últimos en los que el Valencia juega como visitante. Este trozo de código al ejecutarlo nos devuelve:

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR
467	21/22	2021-10-17	Barcelona	Valencia	3	1	1
797	20/21	2020-12-19	Barcelona	Valencia	2	2	X
1275	19/20	2019-09-14	Barcelona	Valencia	5	2	1
1476	18/19	2019-02-02	Barcelona	Valencia	2	2	X
1755	17/18	2018-04-14	Barcelona	Valencia	2	1	1
2171	16/17	2017-03-19	Barcelona	Valencia	4	2	1
2501	15/16	2016-04-17	Barcelona	Valencia	1	2	2
2895	14/15	2015-04-18	Barcelona	Valencia	2	0	1
3376	13/14	2014-02-01	Barcelona	Valencia	2	3	2

Vamos a calcular ahora las cuotas de ataque y defensa para cada uno de los equipos en la fecha determinada.

```
>definir_varibales_explicativas(df.head(10), resultados)
```

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR	FAL_A	FDL_A	FAV_B	FDV_B
467	21/22	2021-10-17	Barcelona	Valencia	3	1	1	1,718637	0,983453	0,860522	1,189826
797	20/21	2020-12-19	Barcelona	Valencia	2	2	X	1,635118	0,802117	0,687927	1,315016
1275	19/20	2019-09-14	Barcelona	Valencia	5	2	1	1,823679	0,879271	1,24949	0,91184
1476	18/19	2019-02-02	Barcelona	Valencia	2	2	X	1,921354	0,906103	0,860798	0,63885
1755	17/18	2018-04-14	Barcelona	Valencia	2	1	1	1,982195	0,598013	1,303939	0,794865
2171	16/17	2017-03-19	Barcelona	Valencia	4	2	1	1,901809	0,676968	1,111653	1,158244
2501	15/16	2016-04-17	Barcelona	Valencia	1	2	2	2,021909	0,668132	0,96044	0,802817
2895	14/15	2015-04-18	Barcelona	Valencia	2	0	1	2,031221	0,632245	1,215856	0,757404
3376	13/14	2014-02-01	Barcelona	Valencia	2	3	2	1,948988	0,655319	1,146809	1,049455

Podemos ver como efectivamente las cuotas cambian en función de la fecha. Puesto que los equipos tienen mejores y peores momentos.

3.3 Generación de tablas finales

Vamos ahora a separar todos los datos de las temporadas 21/22 y posteriores de manera que nuestro estudio consistirá en que, teniendo toda la información previa a la temporada 21/22, trataremos de predecir los resultados de la temporada 21/22.

Así pues, tenemos un dataframe llamado *resultados_previos* que contiene todos los partidos desde la temporada 93/94 hasta la temporada 20/21 y otro dataframe con llamado *resultados_a_predecir* con los partidos de la temporada 21/22.

Vamos a aplicar la función existente para definir variables explicativas con la cual obtenemos el dataframe de los partidos con las nuevas columnas con las variables explicativas. Veamos que obtenemos:

```
>funciones.definir_varibales_explicativas(resultados_previos, resultados)
```

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR	FAL_A	FDL_A	FAV_B	FDV_B
548	20/21	2021-05-23	Sevilla	Alaves	1	0	1	1,011229	0.426860	0.640290	1368134
549	20/21	2021-05-23	Granada	Getafe	0	0	X	1,040971	1173866	0.533575	1040971
550	20/21	2021-05-22	Celta	Betis	2	3	2	1,245957	1346207	0.921089	1186626
551	20/21	2021-05-22	Huesca	Valencia	0	0	X	0,663574	0.978890	0.637677	1186626
552	20/21	2021-05-22	Eibar	Barcelona	0	1	2	0,556231	0.959468	2019310	0.622979
...	...	...	...	...	...	...	...	...	...	...	...

Nos ha generado bien la tabla que estábamos buscando para los resultados previos. Sin embargo, para los resultados a predecir, no queremos que las cuotas se calculen con los resultados de un año anterior respecto de la fecha de partido ya que se supone que este estudio lo estamos realizando antes de empezar la temporada 21/22.

Para ello, mediante el parámetro *pasado* definido en la función *definir_variables_explicativas()*, podemos calcular las cuotas de la temporada utilizando siempre la misma base de datos de partidos previos con los que hacer el estudio.

```
>funciones.definir_varibales_explicativas(resultados_a_predecir,
resultados, pasado = True)
```

Veamos que efectivamente las cuotas para un equipo son iguales para toda la temporada 21/22. Veámoslo filtrando los partidos en los que juega por ejemplo el Celta como equipo local.

```
>resultados_a_predecir[resultados_a_predecir.TeamA == 'Celta'].head()
```

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR	FAL_A	FDL_A	FAV_B	FDV_B
178	21/22	2022-05-15	Celta	Elche	1	0	1	1,154662	1,389328	0,740975	1,23164
205	21/22	2022-05-07	Celta	Alaves	4	0	1	1,154662	1,389328	0,694664	1,23164
224	21/22	2022-04-20	Celta	Getafe	0	2	2	1,154662	1,389328	0,57194	1,133493
257	21/22	2022-04-02	Celta	Real Madrid	1	2	2	1,154662	1,389328	1,574572	0,577331
261	21/22	2022-03-20	Celta	Betis	0	0	X	1,154662	1,389328	1,065151	1,116173

De cara a crear las tablas finales con las que aplicaremos nuestro modelo, definimos una última función.

```
>def tabla_final(df):
>       tabla = pd.DataFrame(columns=['Date', 'TeamA_local', 'TeamA', 'TeamB',
                    'GolesA', 'GolesB', 'FTR', 'FA_A', 'FD_A', 'FA_B', 'FD_B'])
>       for i in range(len(df)):
>              tabla = tabla.append({'Date': df.iloc[i][1], 'TeamA_local': 1, 'TeamA':
              df.iloc[i][2], 'TeamB': df.iloc[i][3], 'GolesA': df.iloc[i][4], 'GolesB':
              df.iloc[i][5], 'FTR': df.iloc[i][6] , 'FA_A':df.iloc[i][7],
              'FD_A':df.iloc[i][8], 'FA_B':df.iloc[i][9], 'FD_B':df.iloc[i][10]},
              ignore_index=True)
>              a = df.iloc[i][6]
>              result = ['1' if a == '2' else '2' if a == '1' else a for a in [a]][0]
>              tabla = tabla.append({'Date': df.iloc[i][1], 'TeamA_local':0, 'TeamA':
              df.iloc[i][3], 'TeamB':df.iloc[i][2], 'GolesA':df.iloc[i][5], 'GolesB':
              df.iloc[i][4], 'FTR': result , 'FA_A':df.iloc[i][9], 'FD_A':df.iloc[i][10],
              'FA_B':df.iloc[i][7], 'FD_B':df.iloc[i][8]}, ignore_index=True)
>       return tabla
```

Para ver lo que hace esta función lo veremos con un ejemplo concreto. Partimos de una tabla

	Temporada	Date	TeamA	TeamB	FTHG	FTAG	FTR	FAL_A	FDL_A	FAV_B	FDV_B
548	20/21	2021-05-23	Sevilla	Alaves	1	0	1	1,011229	0.426860	0.640290	1368134

Al llamar a la función, dándole esta tabla, nos devuelve otra de la siguiente forma:

	Date	TeamA_local	TeamA	TeamB	GolesA	GolesB	FTR	FA_A	FD_A	FA_B	FD_B
0	2021-05-23	1	Sevilla	Alaves	1	0	1	1,011229	0,42686	0,64029	1,368134
1	2021-05-23	0	Alaves	Sevilla	0	1	2	0,64029	1,368134	1,011229	0,42686

Podemos ver como el partido sale duplicado. De esta manera podemos analizar independientemente el rendimiento de cada equipo.

En el primer registro de este partido, analizaremos el rendimiento del Sevilla, escogiendo como FA_A y FD_A (Fuerza atacante y defensiva del equipo A), su fuerza atacante y defensiva jugando de local (FAL_A y FDL_A en la tabla anterior) ya que el Sevilla juega de local. De la misma manera, para los valores de FA_B y FD_B escogeremos los valores de las cuotas del Alavés en campo visitante (FAV_B y FDV_B en la tabla anterior).

En el segundo registro de este partido, analizamos el rendimiento del Alavés que en este caso será el Team_A. De la misma manera definimos FA_A, FD_A como las cuotas del Alavés esta vez en campo visitante (FAV_B y FDV_B en la tabla anterior) y FA_B, FD_B como las cuotas del Sevilla jugando como local (FAL_A y FDL_A en la tabla anterior)

Así pues, generamos finalmente las 2 tablas finales

```
>tabla _previos = tabla_final(resultados_previos)
>tabla _a_predecir = tabla_final(resultados_a_predecir)
```

3.4 El modelo

Recordamos el objetivo del estudio que es a tratar de ajustar un modelo para con él, intentar predecir los resultados de la temporada 21/22.

Para ello utilizaremos la librería interna de Python scikit-learn (sklearn) la cual es una librería de aprendizaje automático de código abierto en Python. Esta incluye una variedad de algoritmos, incluyendo clasificación, regresión y agrupamiento.

Utilizaremos en concreto la sublibrería linear_model la cual contiene un conjunto de modelos lineales y funciones de ayuda para trabajar con ellos. Entre los modelos lineales incluidos en esta sublibrería se encuentran la regresión lineal, la regresión logística y la regresión de ridge, pero nosotros

Para nuestro estudio utilizaremos la clase PoissonRegressor situada en la sublibrería sklearn.linear_model para ajustar datos que siguen, como hemos visto una distribución de Poisson.

Tenemos las dos tablas finales *tabla _previos* y *tabla_a_predecir* que recordamos que en la primera tenemos todos los partidos disputados previos a la temporada 21/22 definiendo para cada partido unas cuotas en función de los resultados del equipo en los 365 días previos a cada partido y por otro lado la segunda tabla contiene los partidos de la temporada 21/22 con unas cuotas definidas antes de empezar la temporada con las estadísticas de los equipos a lo largo su último año antes de empezar la temporada.

Recordemos como es la *tabla_previos:*

	Date	TeamA_local	TeamA	TeamB	GolesA	GolesB	FTR	FA_A	FD_A	FA_B	FD_B
0	2021-05-23	1	Sevilla	Alaves	1	0	1	1,011229	0,42686	0,64029	1,368134
1	2021-05-23	0	Alaves	Sevilla	0	1	2	0,64029	1,368134	1,011229	0,42686
2	2021-05-23	1	Granada	Getafe	0	0	X	1,040971	1,173866	0,533575	1,040971
3	2021-05-23	0	Getafe	Granada	0	0	X	0,533575	1,040971	1,040971	1,173866
4	2021-05-22	1	Celta	Betis	2	3	2	1,245957	1,346207	0,921089	1,186626
...	...	...	...	...	...	...	...	...	...	...	...
14435	2001-08-26	0	Alaves	Tenerife	2	0	1	1,14966	0,99635	0	1,768707
14436	2001-08-26	1	ayo Vallecan	Villarreal	1	2	2	0,882482	1,193878	1,193878	0,79708
14437	2001-08-26	0	Villarreal	ayo Vallecan	2	1	1	1,193878	0,79708	0,882482	1,193878
14438	2001-08-25	1	Valencia	Real Madrid	1	0	1	0,877489	0,450355	1,260993	0,735958
14439	2001-08-25	0	Real Madrid	Valencia	0	1	2	1,260993	0,735958	0,877489	0,450355

3.4.1 Definición variable respuesta

Nuestra variable respuesta será el número de goles que anota el equipo A (GolesA) utilizando como variables predictoras las columnas ['TeamA_local', 'FA_A', 'FD_A', 'FA_B', 'FD_B']. Así pues, para la tabla de *tabla_previos* definimos:

X_previos

	TeamA_local	FA_A	FD_A	FA_B	FD_B
0	1	1,01123	0,42686	0,64029	1,36813
1	0	0,64029	1,36813	1,01123	0,42686
2	1	1,04097	1,17387	0,53358	1,04097
3	0	0,53358	1,04097	1,04097	1,17387
4	1	1,24596	1,34621	0,92109	1,18663
...	...	...	...	...	...
14435	0	1,14966	0,99635	0	1,76871
14436	1	0,88248	1,19388	1,19388	0,79708
14437	0	1,19388	0,79708	0,88248	1,19388
14438	1	0,87749	0,45036	1,26099	0,73596
14439	0	1,26099	0,73596	0,87749	0,45036

y_previos

GolesA
1
0
0
0
2
...
0
1
0
4
1

y de la misma manera, partiremos la tabla *tabla_a_predecir*:

X_a_predecir

	TeamA_local	FA_A	FD_A	FA_B	FD_B
0	1	1,30862	0,97253	1,20408	0,53884
1	0	1,20408	0,53884	1,30862	0,97253
2	1	1,0392	0,50942	0,78729	0,88524
3	0	0,78729	0,88524	1,0392	0,50942
4	1	1,6935	0,92622	1,43564	0,80826
...	...	...	...	...	...
539	0	1,57457	0,57733	0,80826	1,15777
540	1	0,65431	1,38933	0,87991	1,0392
541	0	0,87991	1,0392	0.654309	1,38933
542	1	1,27975	1,01189	0,57194	1,13349
543	0	0,57194	1,13349	1,27975	1,01189

y_a_predecir

GolesA
1
2
1
0
0
...
4
1
1
1
0

3.4.2 Cálculo de las β y predicción

Llega el momento de entrenar nuestro modelo. Para ello utilizaremos las tablas *X_previos* y *y_previos* para estimar las β que hemos definido en la explicación del GLM.

```
>from sklearn.linear_model import PoissonRegressor
```

```
>modelo=PoissonRegressor()

>modelo.fit(X_previos, y_previos)
```

Una vez definidas estas β, podemos conocer cuáles son mediante

```
>modelo.coef_
```

Que nos devuelve la lista [0.07834002, 0.15082545, -0.03569365, -0.03534429, 0.09964336]. Con estas β podemos finalmente calcular el vector *y_predicho* y este vector lo compararemos con el vector *y_a_predecir* para ver cuántos valores hemos acertado

```
>y_predicho = modelo.predict(X_a_predecir)
>acierto = sum(y_predicho==y_a_predecir)/y_predicho.shape[0] * 100
```

Tenemos pues un accuracy del 0%. Este resultado puede asustar, pero se debe a que los valores que *y_predicho* son de la forma [1.38225013, 1.32923785, …, 1.49115284, 1.18884197] que es muy poco probable que coincidan con los resultados *y_a_predecir* ya que estos son valores enteros sin decimales.

3.4.3 Análisis de los resultados

De la tabla que teníamos *tabla_a_predecir,*

	Date	TeamA_local	TeamA	TeamB	GolesA	GolesB	FTR	FA_A	FD_A	FA_B	FD_B
0	2022-05-22	1	Sociedad	Ath Madrid	1	2	2	1,308617	0,972529	1,204084	0,538842
1	2022-05-22	0	Ath Madrid	Sociedad	2	1	1	1,204084	0,538842	1,308617	0,972529
2	2022-05-22	1	Sevilla	Ath Bilbao	1	0	1	1,039196	0,50942	0,787286	0,885241
3	2022-05-22	0	Ath Bilbao	Sevilla	0	1	2	0,787286	0,885241	1,039196	0,50942
4	2022-05-22	1	Barcelona	Villarreal	0	2	2	1,693504	0,926219	1,435639	0,808263
…	…	…	…	…	…	…	…	…	…	…	…
539	2021-08-14	0	Real Madrid	Alaves	4	1	1	1,574572	0,577331	0,808263	1,157773
540	2021-08-14	1	Cadiz	Levante	1	1	X	0,654309	1,389328	0,879908	1,039196
541	2021-08-14	0	Levante	Cadiz	1	1	X	0,879908	1,039196	0.654309	1,389328
542	2021-08-13	1	Valencia	Getafe	1	0	1	1,27975	1,011894	0,57194	1,133493
543	2021-08-13	0	Getafe	Valencia	0	1	2	0,57194	1,133493	1,27975	1,011894

Nos quedaremos con las columnas [‘TeamA’, ‘TeamB’, ‘GolesA’, ‘FTR’] y añadimos una nueva columna con los valores de *y_predicho*:

	TeamA	TeamB	GolesA	FTR	y_predicho
0	Sociedad	Ath Madrid	1	2	1,38225
1	Ath Madrid	Sociedad	2	1	1,329238
2	Sevilla	Ath Bilbao	1	1	1,417444
3	Ath Bilbao	Sevilla	0	2	1,188574
4	Barcelona	Villarreal	0	2	1,494929
…	…	…	…	…	…
539	Real Madrid	Alaves	4	1	1,455361
540	Cadiz	Levante	1	X	1,311884
541	Levante	Cadiz	1	X	1,326454
542	Valencia	Getafe	1	1	1,491153
543	Getafe	Valencia	0	2	1,188842

Podemos ver que nuestro valor de *y_predicho* está muy alejado del *GolesA*. Aun parecer que no tengan sentido estos valores, podemos calcular según los valores de *y_predicho* una nueva columna de resultados predichos la cual llamaremos *FTR_predicho*.

Para ello definimos una función:

```
>def lista_resultados(lista):
>       resultados = []
>       for i in range(0, len(lista)-1,2):
>               if lista[i]>lista[i+1]:
>                       resultados.append('1')
>                       resultados.append('2')
>       elif lista[i]<lista[i+1]:
>               resultados.append('2')
>               resultados.append('1')
>       else:
>               resultados.append('X')
>               resultados.append('X')
>       return resultados
```

Con esta fórmula podemos calcular en función de los valores cual es el resultado del partido. Así pues, nos queda:

	TeamA	TeamB	GolesA	FTR	y_predicho	FTR_predicho
0	Sociedad	Ath Madrid	1	2	1,38225	1
1	Ath Madrid	Sociedad	2	1	1,329238	2
2	Sevilla	Ath Bilbao	1	1	1,417444	1
3	Ath Bilbao	Sevilla	0	2	1,188574	2
4	Barcelona	Villarreal	0	2	1,494929	1
...	...	...	...	...	...	...
539	Real Madrid	Alaves	4	1	1,455361	1
540	Cadiz	Levante	1	X	1,311884	2
541	Levante	Cadiz	1	X	1,326454	1
542	Valencia	Getafe	1	1	1,491153	1
543	Getafe	Valencia	0	2	1,188842	2

Si comparamos ahora los resultados de FTR con FTR_predicho, tenemos que finalmente mediante nuestro modelo hemos sido capaces de, antes de empezar la temporada 21/22, predecir el resultado de un 50,37% de los partidos.

4 Conclusiones

El objetivo del trabajo era tratar de predecir y acertar el resultado del 50% de los partidos disputados en la temporada 21/22. Al aplicar el modelo entrenándolo con los datos de las temporadas previas, hemos conseguido un acierto del 50,37%.

Pese a el resultado obtenido ser muy ajustado al objetivo inicial del trabajo, considero que independientemente del objetivo numérico, he alcanzado con creces los objetivos personales que tenía frente a este trabajo. Elaborar un proyecto matemático para finalizar mi etapa como estudiante con el cual poder disfrutar, seguir aprendiendo y sobre todo, encontrar un entorno práctico en el cual poder aplicarlo, en nuestro caso predecir resultados de partidos de fútbol.

Referencias

- Varios autores. "Distribución de Poisson." Wikipedia. https://es.wikipedia.org/wiki/Distribuci%C3%B3n_de_Poisson

- Josep Fortiana. Noviembre 2013. Modelos lineales generalizados. Apuntes de la asignatura de Estadística. Universidad de Barcelona

- Eva María García Quintero. Julio 2014. Aplicación de Modelos de Regresión de Poisson Bivariados. Trabajo Fin de Máster. Universidad de Vigo

- M. J. Maher. 1982. Modelling association football scores.

- Karlis, D. & Tsiamyrtzis, P. 2008. Exact Bayesian modeling for bivariate Poisson data and extensions. Statistics and Computing, 18 (1), pp. 27-40.

- Karlis, D. & Ntzoufras, L. 2003. Analysis of sports data by using bivariate Poisson models. Journal of the Royal Statistical Society Series D: The Statistician, 52 (3), pp. 381-393.

- Malcata, R.M., Hopkins, W.G. & Richardson, S. 2012. Modelling the progression of competitive performance of an academy's soccer teams. Journal of Sports Science and Medicine, 11 (3), pp. 533- 536.

- Karlis, D. & Ntzoufras, I. 2011. Robust fitting of football prediction models. IMA Journal Management Mathematics, 22 (2), pp. 171-182.

- Baio, G. & Blangiardo, M. 2010. Bayesian hierarchical model for the prediction of football results. Journal of Applied Statistics, 37 (2), pp. 253-264.

- Skinner, G.K & Freeman, G.H. 2009. Soccer matches as experi- ments: How often does the 'best' team win? Journal of Applied Statistics, 36 (10), pp. 1087-1095.

- Karlis, D. & Ntzoufras, I. 2009. Bayesian modelling of football outcomes: Using the Skellam's distribution for the goal difference. IMA Journal Management Mathematics, 20 (2), pp. 133-145.

- Greenhough, J., Birch, P.C., Chapman, S.C. & Rowlands, G. 2002. Football goal distributions and extremal statistics. Physica A: Sta- tistical Mechanics and its Applications, 316 (1-4), pp. 615-624.

- Dyte, D. & Clarke, S.R. 2000. A ratings-based Poisson model for World Cup soccer simulation. Journal of the Operational Research Society, 51 (8), pp. 993-998.

[1] https://www.football-data.co.uk/spainm.php

[2] https://www.football-data.co.uk/notes.txt