

Universidad de San Carlos de Guatemala  
Escuela de Ciencias Físicas y Matemáticas  
Prácticas Finales  
Asesor: MSc. Damián Ochoa

# ESTUDIO ESTADÍSTICO EN COMPETICIONES DEPORTIVAS

Diego Sarceño  
201900109

Guatemala, 31 de enero del 2024

# Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>                          | <b>3</b>  |
| <b>2. Objetivos</b>                             | <b>3</b>  |
| 2.1. General . . . . .                          | 3         |
| 2.2. Específicos . . . . .                      | 3         |
| <b>3. Marco Teórico</b>                         | <b>4</b>  |
| 3.1. Estadística . . . . .                      | 4         |
| 3.1.1. Distribución de Poisson . . . . .        | 4         |
| 3.1.2. Distribución de Skellam . . . . .        | 4         |
| 3.1.3. Análisis Bayesiano . . . . .             | 4         |
| 3.2. Librerías . . . . .                        | 5         |
| 3.2.1. Numpy . . . . .                          | 5         |
| 3.2.2. Matplotlib . . . . .                     | 5         |
| 3.2.3. Pandas . . . . .                         | 5         |
| 3.2.4. Scipy . . . . .                          | 5         |
| 3.2.5. Scikit-Learn . . . . .                   | 6         |
| 3.2.6. Seaborn . . . . .                        | 6         |
| 3.2.7. Extras . . . . .                         | 6         |
| <b>4. Metodología</b>                           | <b>6</b>  |
| 4.1. Conjunto de Datos . . . . .                | 6         |
| 4.1.1. Opciones de Conjuntos de Datos . . . . . | 6         |
| 4.1.2. Elección del Conjunto de Datos . . . . . | 6         |
| 4.1.3. Extracción y Limpieza . . . . .          | 7         |
| 4.1.4. Elección del Modelo . . . . .            | 8         |
| 4.1.5. Creación del Modelo . . . . .            | 10        |
| <b>5. Resultados</b>                            | <b>11</b> |
| <b>6. Discusión de Resultados</b>               | <b>16</b> |
| <b>7. Conclusiones</b>                          | <b>16</b> |
| <b>8. Apéndice</b>                              | <b>17</b> |

## Índice de figuras

|    |  |    |
|----|--|----|
| 1. | Diagramas de correlación para La Liga y la Premier League desde la temporada 2016-17 | 8  |
| 2. | Diagrama de Correlación para la UEFA Champions League, desde la temporada 2016-17    | 8  |
| 3. | Histograma de resultados en La Liga y la Premier League desde la temporada 2016-17.  | 9  |
| 4. | Frecuencia de goles por partido (LaLiga).  | 9  |
| 5. | Frecuencia de diferencia de goles por partido (LaLiga).                              | 10 |

## Índice de cuadros

|     |   |    |
|-----|---|----|
| 1.  | Datos de <i>football-data</i> .                                   | 7  |
| 2.  | Matriz de Probabilidad entre Athletic de Bilbao v FC Barcelona.   | 11 |
| 3.  | Matriz de Probabilidad entre Manchester City v Manchester United. | 11 |
| 4.  | Predicción Jornada 27 La Liga según Modelo de Poisson             | 12 |
| 5.  | Predicción Jornada 27 Premier League según Modelo de Poisson      | 13 |
| 6.  | Resultados Jornada 27 La Liga                                     | 13 |
| 7.  | Resultados Jornada 27 Premier League                              | 14 |
| 8.  | Predicción 8vos de Final Vuelta UCL                               | 14 |
| 9.  | Resultado más probable en los 8vos de Final Vuelta UCL            | 15 |
| 10. | Predicción 4tos de Final UCL                                      | 15 |
| 11. | Resultado más probable en los 4vos de Final UCL                   | 15 |
| 12. | Predicción Semi-Finales UCL                                       | 15 |
| 13. | Resultado más probable en las Semi-Finales UCL                    | 16 |
| 14. | Predicción FINAL UCL 2023-24                                      | 16 |
| 15. | Predicción FINAL UCL 2023-24                                      | 16 |

# 1. Introducción

El mundo del deporte, en particular el fútbol, ha sido objeto de análisis y predicción a lo largo de los años. Con el avance de la tecnología y el acceso a grandes cantidades de datos, se ha vuelto cada vez más común utilizar métodos estadísticos y modelos matemáticos para predecir resultados deportivos con mayor precisión. En este contexto, el presente informe se centra en el desarrollo y la evaluación de un sistema de predicción de resultados deportivos, específicamente orientado hacia la prestigiosa UEFA Champions League (UCL).

El objetivo principal de este proyecto ha sido utilizar un enfoque basado en el modelo de Poisson para pronosticar los resultados de los partidos de la UEFA Champions League. El modelo de Poisson, ampliamente utilizado en la teoría de probabilidad y estadística, proporciona un marco matemático sólido para estimar la probabilidad de eventos, en este caso, los resultados (goles) de los encuentros deportivos.

A lo largo de este informe, se detallará el proceso de desarrollo del sistema de predicción, que incluye la recopilación y el procesamiento de datos relevantes, la implementación del modelo de Poisson y la evaluación de su rendimiento. Además, se discutirán los resultados obtenidos, así como las implicaciones y limitaciones del modelo propuesto.

Es importante destacar que, si bien los modelos de predicción deportiva pueden ofrecer insights valiosos, siempre existe un grado de incertidumbre inherente a la naturaleza impredecible del deporte. Por lo tanto, aunque se haya logrado una efectividad aproximada del 55 % en las predicciones, es fundamental interpretar los resultados con cautela y reconocer las posibles variables no consideradas que podrían influir en los resultados finales.

## 2. Objetivos

### 2.1. General

- El objetivo general de este estudio es profundizar en el análisis estadístico y comprender los procesos subyacentes a los programas de predicción probabilística de resultados deportivos. Se busca adquirir conocimiento sobre cómo se utilizan los métodos estadísticos y probabilísticos en este contexto específico, con el fin de mejorar la precisión y la fiabilidad de las predicciones deportivas.

### 2.2. Específicos

1. Realizar una revisión de la literatura académica y científica relacionada con la teoría estadística aplicada a la predicción de resultados deportivos. Se buscará identificar y recopilar artículos, estudios y publicaciones relevantes que aborden métodos, técnicas y modelos utilizados en este campo, con el objetivo de comprender los fundamentos teóricos que sustentan los programas de predicción.
2. Recopilar los datos necesarios para llevar a cabo simulaciones y análisis estadísticos. Se realizará un proceso de recolección de datos que incluirá la identificación de fuentes confiables de información deportiva, la selección de variables relevantes y la obtención de datos históricos sobre eventos deportivos pasados, resultados de partidos, características de los equipos y cualquier otro factor relevante para el análisis.
3. Utilizar los datos recopilados y la teoría revisada para realizar simulaciones y experimentos. Se aplicarán los métodos estadísticos y probabilísticos aprendidos para analizar los datos y generar predicciones sobre los resultados deportivos. Se llevarán a cabo simulaciones computacionales utilizando modelos y algoritmos adecuados, con el fin de evaluar la efectividad y precisión de los programas de predicción desarrollados.

### 3. Marco Teórico

#### 3.1. Estadística

##### 3.1.1. Distribución de Poisson

La distribución de Poisson es una distribución de probabilidad discreta que describe el número de eventos que ocurren en un intervalo de tiempo o espacio específico, dado un promedio conocido de ocurrencias y bajo ciertas condiciones. En nuestro caso, lo estudiaremos con la cantidad de goles por el equipo local y visitante por partido. Esta distribución tiene la siguiente función de probabilidad

$$f(x) = P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}. \quad (3.1.1.1)$$

**Proposición 3.1.1.1** Sean  $X$  y  $Y$  dos variables aleatorias independientes con distribución  $Poisson(\lambda_1)$  y  $Poisson(\lambda_2)$ , respectivamente. Entonces

$$X + Y \sim Poisson(\lambda_1 + \lambda_2).$$

##### 3.1.2. Distribución de Skellam

La distribución de Skellam es una distribución de probabilidad discreta que modela la diferencia entre dos variables aleatorias independientes y de Poisson. Específicamente, si  $X$  e  $Y$  son dos variables aleatorias independientes y de Poisson con parámetros  $\lambda_1$  y  $\lambda_2$ , respectivamente, entonces la variable  $Z = X - Y$  sigue una distribución de Skellam [1]. Esta distribución tiene una función de probabilidad

$$f_{SK}(z) = P(Z = z) = e^{-(\lambda_1 + \lambda_2)} \left( \frac{\lambda_1}{\lambda_2} \right)^{z/2} I_k^1(2\sqrt{\lambda_1 \lambda_2}). \quad (3.1.2.1)$$

Esta distribución presenta las siguientes características:

**Independencia:** Las dos variables aleatorias deben ser independientes entre sí.

**Distribución de Poisson:** Cada variable aleatoria debe seguir una distribución de Poisson, lo que implica que los eventos deben ser contables y aleatorios, y la ocurrencia de un evento no debe influir en la ocurrencia del otro.

Esta distribución se relaciona dos eventos aleatorios, cosa que se aplica directamente con nuestro caso de estudio, la diferencia de variables aplicada a nuestro caso sería la diferencia de goles anotados y goles encagados por partido.

##### 3.1.3. Análisis Bayesiano

El análisis bayesiano es un método estadístico que utiliza el teorema de Bayes para actualizar las creencias sobre la probabilidad de un evento a la luz de nueva evidencia. Este enfoque se basa en la idea de que la probabilidad de un evento es una medida de la creencia en su ocurrencia.

**Teorema 3.1.3.1**  $P(A/B) = P(B/A) * P(A) / P(B)$ ,

donde:

- $P(A|B)$ : es la probabilidad de  $A$  dado  $B$ .

---

<sup>1</sup>Función modificada de Bessel del primer tipo  $I_\alpha(x) = \sum_{k=0}^{\infty} \frac{1}{k! \Gamma(k + \alpha + 1)} \left( \frac{x}{2} \right)^{2k + \alpha}$ .

- $P(B|A)$ : es la probabilidad de  $B$  dado  $A$ .
- $P(A)$ : es la probabilidad marginal<sup>2</sup> de  $A$ .
- $P(B)$ : es la probabilidad marginal de  $B$ .

Adicional al Teorema de Bayes, es necesario tener en cuenta su versión generalizada y términos como la probabilidad total.

**Teorema de Bayes para Conjuntos de Eventos:** Se utiliza para calcular la probabilidad de un conjunto de eventos dado otro conjunto de eventos.

$$P(A_1 \cap A_2 \cap \dots \cap A_n | B_1 \cap B_2 \cap \dots \cap B_n) =$$

$$P(B_1 \cap B_2 \cap \dots \cap B_n | A_1 \cap A_2 \cap \dots \cap A_n) * P(A_1 \cap A_2 \cap \dots \cap A_n) / P(B_1 \cap B_2 \cap \dots \cap B_n). \quad (3.1.3.1)$$

**Probabilidad Total:** Se utiliza para calcular la probabilidad de un evento como las probabilidades de sus eventos mutuamente excluyentes.

$$P(B) = \sum_{i=1}^n P(B|A_i) P(A_i). \quad (3.1.3.2)$$

## 3.2. Librerías

### 3.2.1. Numpy

NumPy es una biblioteca fundamental para computación científica en Python. Proporciona un poderoso objeto de matriz multidimensional (`numpy.array`) y funciones para operar eficientemente en estas matrices. NumPy se utiliza ampliamente en cálculos numéricos, álgebra lineal, transformadas de Fourier, generación de números aleatorios y mucho más.

### 3.2.2. Matplotlib

Matplotlib es una biblioteca de trazado en Python que produce figuras de calidad de publicación en una variedad de formatos. `pyplot` es una interfaz para Matplotlib que proporciona una forma similar a MATLAB de trazar gráficos. Se utiliza para crear gráficos 2D, como histogramas, gráficos de dispersión, líneas, barras, entre otros.

### 3.2.3. Pandas

Pandas es una biblioteca de Python para manipulación y análisis de datos. Proporciona estructuras de datos y herramientas de análisis de alto nivel, como DataFrames y Series, que permiten realizar operaciones de datos rápidas y flexibles. Es útil para la limpieza de datos, la manipulación de datos faltantes, la agregación y la agrupación de datos, entre otras tareas.

### 3.2.4. Scipy

Scipy es una biblioteca de Python para matemáticas, ciencia e ingeniería. Proporciona una amplia variedad de funciones y algoritmos para optimización, álgebra lineal, integración, interpolación, estadísticas, procesamiento de señales y mucho más. Scipy se basa en NumPy y se utiliza comúnmente junto con él para aplicaciones científicas y de ingeniería.

---

<sup>2</sup>La **probabilidad marginal** se refiere a la probabilidad de que un evento ocurra sin tener en cuenta la ocurrencia de otros eventos

### 3.2.5. Scikit-Learn

Scikit-learn es una biblioteca de aprendizaje automático en Python que ofrece herramientas simples y eficientes para el análisis predictivo y la minería de datos. Proporciona algoritmos de clasificación, regresión, agrupación, reducción de dimensionalidad, selección de modelos y preprocesamiento de datos, entre otros. Scikit-learn se integra bien con NumPy y SciPy y es una de las bibliotecas de aprendizaje automático más utilizadas en Python.

### 3.2.6. Seaborn

Seaborn es una biblioteca de visualización de datos en Python basada en Matplotlib. Proporciona una interfaz de alto nivel para crear gráficos estadísticos atractivos e informativos. Seaborn simplifica la creación de gráficos complejos como diagramas de dispersión con distribuciones marginales, diagramas de violín, diagramas de caja y bigotes, entre otros.

### 3.2.7. Extras

Adicional a las librerías principales utilizadas, se tienen librerías que apoyaron a ciertas tareas, como facilitar la búsqueda de archivos para su lectura y posterior análisis, o simplemente ignorar advertencias innecesarias para mejorar la estética del *notebook*.

**Os** El módulo `os` es una parte estándar de Python que proporciona funciones para interactuar con el sistema operativo. Se utiliza para realizar operaciones relacionadas con el sistema de archivos, como navegar por el sistema de archivos, manipular rutas de archivo, crear y eliminar directorios, etc.

**Warnings** El módulo `warnings` proporciona control sobre las advertencias emitidas por Python. Se utiliza para mostrar o suprimir advertencias durante la ejecución del programa. Es útil para depurar código y para controlar el flujo de advertencias en diferentes partes de una aplicación.

## 4. Metodología

### 4.1. Conjunto de Datos

#### 4.1.1. Opciones de Conjuntos de Datos

Como es bien sabido y lógico, es necesario un buen conjunto de datos para poder hacer predicciones y análisis de calidad, para ello se encontraron varias bases/conjuntos de datos que pueden ser útiles. Tales como: [Kaggle \(UEFA Champions League 2016-2022\)](#), [European Soccer Database](#), [Kaggle \(European Soccer Database Supplementary\)](#) y [footystats](#).

#### 4.1.2. Elección del Conjunto de Datos

Para la elección de los conjuntos de datos a utilizar se tomó en cuenta: la limpieza de los datos, la fecha de su última actualización (dado que si fue hace un tiempo considerable, los datos no tendrían validez ni la posibilidad de realizar predicciones reales), la información dada (goles, resultados, posesión, tiros, tiros a puerta, etc.) y las competiciones que incluye. Con esto, se tuvo 3 casos: en el caso de las bases Europeas encontradas en Kaggle, las temporadas tratadas en su base de datos son las comprendidas entre 2008 y 2016 lo que nos es inviable para realizar predicciones en una temporada 8 años después. El segundo caso es el de la página de "footystats" que, a pesar de tener un conjunto de datos algo "sucio", proporciona la mayor cantidad de datos y cuenta con los ámbitos, a priori, más importantes; sin embargo, la información que ofrecen es de pago y lastimosamente, no existe una opción de prueba.

Tomando esto en cuenta, el tercer caso y el mejor conjunto de datos para realizar las predicciones es el proporcionado por la página [football-data](#), la cual nos brinda información acerca de las ligas de: Inglaterra (hasta la 5ta división), Escocia (hasta la 4ta división), Alemania (hasta la 2da división), Italia (hasta la 2da división), España (hasta la 2da división), Francia (hasta la 2da división), Países Bajos, Bélgica, Portugal, Turquía y Grecia. Y temporadas desde la disputada en 1994/95 hasta la jornada 26 de la temporada 2023/24 (disputada entre el 23 y el 26 de febrero del 2024, día en el que se escribió esta sección, tomando como referencia la Liga Española).

Además, la idea que dio vida a este proyecto era predecir el ganador de la UEFA Champions League 2024, y uno de los objetivos de este proyecto sigue siendo este. Para ello se escogió un conjunto de datos que cuenta con estadísticas de las temporadas entre 2016 y 2022, lo que solo nos deja sin las estadísticas de la temporada 2022-2023 (que nos puede ser útil para realizar test extras). Esto se realizará luego de hacer las respectivas pruebas con algunas de las grandes ligas europeas para determinar si el modelo funciona bien. Los datos están bastante limpios (dado que es una competición de tal prestigio como la UCL, no hay extraordinarios como sucede en ligas locales y más en divisiones inferiores).

#### 4.1.3. Extracción y Limpieza

Para el conjunto de *football-data* se tienen las siguientes columnas de interes para nosotros:

- Div: División de liga.
- Date: Fecha del partido.
- HomeTeam: Equipo local.
- AwayTeam: Equipo visitante.
- FTHG: Goles a tiempo cumplido del equipo local.
- FTAG: Goles a tiempo cumplido del equipo visitante.
- FTR: Resultado del partido ('H' Local, 'A' Visitante y 'D' Empate, por sus siglas en inglés).
- HTHG: Goles a medio tiempo del equipo local.
- HTAG: Goles a medio tiempo del equipo visitante.

En la cuales, afortunadamente, no se tienen celdas sin datos y todo está estandarizado, sin valores fuera de lo esperado y utilizables.

|      | Date                | HomeTeam   | AwayTeam   | FTHG | FTAG | FTR | HTHG | HTAG |
|------|---------------------|------------|------------|------|------|-----|------|------|
| 0    | 2005-08-27 00:00:00 | Alaves     | Barcelona  | 0    | 0    | D   | 0    | 0    |
| 1    | 2005-08-27 00:00:00 | Ath Bilbao | Sociedad   | 3    | 0    | H   | 0    | 0    |
| 2    | 2005-08-27 00:00:00 | Valencia   | Betis      | 1    | 0    | H   | 0    | 0    |
| ⋮    | ⋮                   | ⋮          | ⋮          | ⋮    | ⋮    | ⋮   | ⋮    | ⋮    |
| 7078 | 2024-02-11 00:00:00 | Barcelona  | Granada    | 3    | 3    | D   | 1    | 1    |
| 7079 | 2024-02-12 00:00:00 | Almeria    | Ath Bilbao | 0    | 0    | D   | 0    | 0    |

Cuadro 1: Datos de *football-data*.

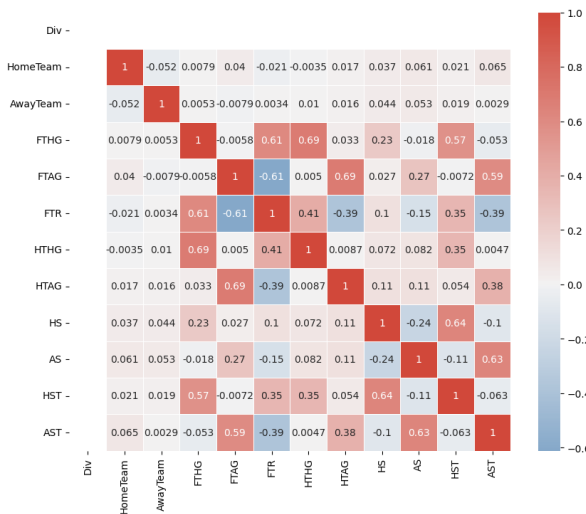
Para el conjunto de datos de la UEFA Champions League 2016-2022, se tienen los partidos ordenados en fase de grupos (96 partidos) al inicio de cada bloque de temporada, luego se tiene la ronda de octavos de final (16 partidos), cuartos de final (8 partidos), semi-finales (4 partidos) y la final (1 partido), a



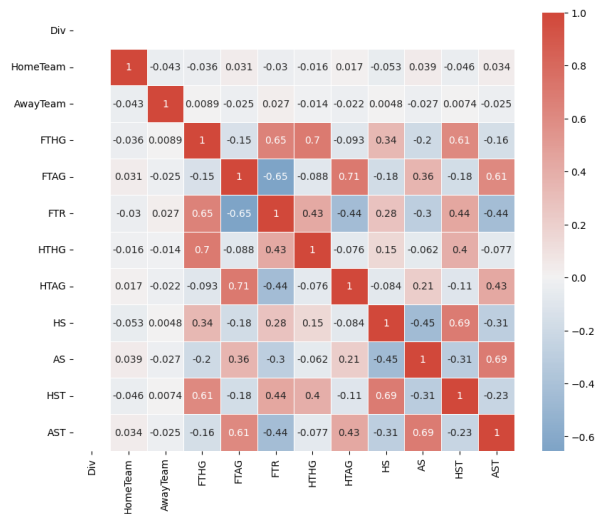
excepción del año 2020, que por la pandemia, únicamente se disputó un partido por ronda (en campo neutral) a partir de los octavos de final, en lugar de lo normal que son partidos ida y vuelta. Fuera de eso, y gracias a la competición, no se tienen valores de campo anormales (como partidos cancelados, partidos ganados por *default* o, directamente, falta de datos).

#### 4.1.4. Elección del Modelo

Para la elección del modelo se inició con una idea de realizarlo por medio de inteligencia artificial y aprendizaje supervisado. Sin embargo, los problemas presentados hicieron buscar otra alternativa y se decidió hacer un modelo puramente estadístico. Por ende, se eligieron las variables más importantes o de mayor peso en los resultados como tal. Para ello, se realizó una matriz de correlación, con esto se pretende entender cuales de las variables son de mayor importancia en el resultado del partido.



(a) Diagrama con Coeficientes de Correlación para La Liga Española



(b) Diagrama con Coeficientes de Correlación para la Premier League

Figura 1: Diagramas de correlación para La Liga y la Premier League desde la temporada 2016-17

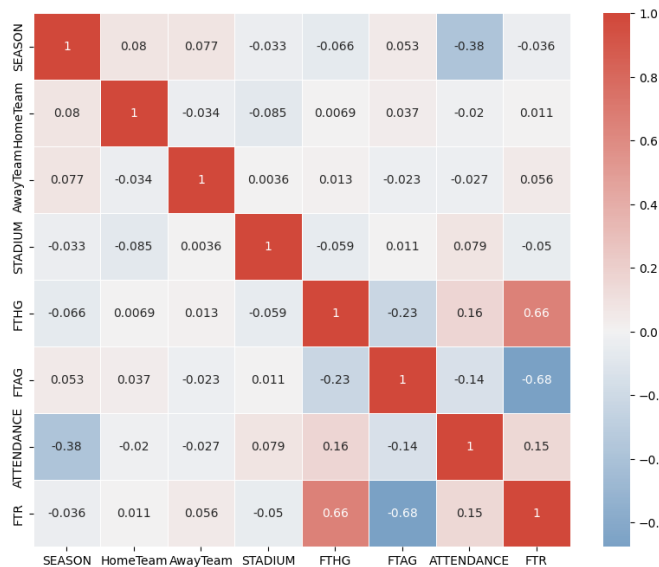
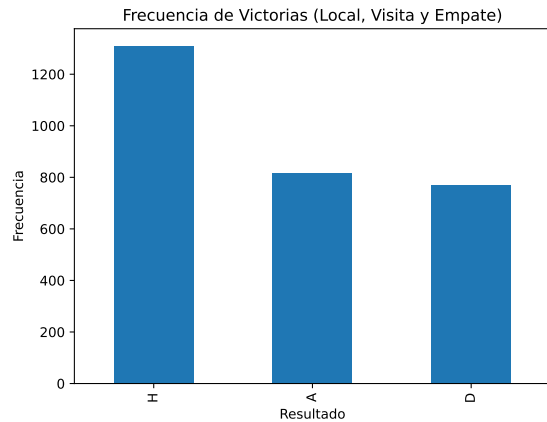


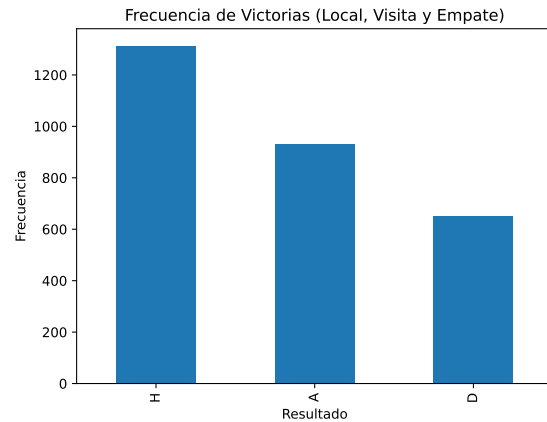
Figura 2: Diagrama de Correlación para la UEFA Champions League, desde la temporada 2016-17

Ambas figuras 1 y 2 vemos que las variables que más relación tienen con la variable objetivo ('FTR') son las de los goles, tanto del equipo visitante, como del local, en ambas divisiones (por medio tiempo y tiempo completo, pero esta era de esperarse).

Ya teniendo cual es la variable con mayor relación con el resultado del encuentro, podemos analizar cual es el resultado que se da con mayor frecuencia.



(a) Cantidad de resultados en La Liga



(b) Cantidad de resultados en la Premier League

Figura 3: Histograma de resultados en La Liga y la Premier League desde la temporada 2016-17.

Dado que el resultado más frecuente (y por una cantidad considerable) es la victoria del equipo local, es claro que esto puede afectar de cierto modo al modelo realizado. Dado que este se construirá en base a los goles anotados, el hecho de que la mayor cantidad de victorias (y, por ende, mayor cantidad de goles) sea del equipo local afectará en cierto modo las predicciones.

Ahora, es necesario elegir que distribución de probabilidad modelará el problema en cuestión. Como ya se vio en las figuras ?? y ??, los goles son las variables que más relacionadas están con el resultado del encuentro, veremos que tipo de distribución cumplen las diferentes cantidades de goles por partido.

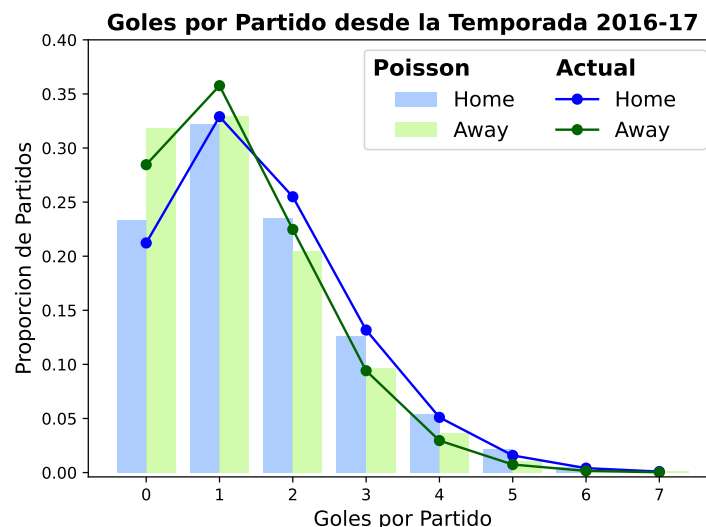


Figura 4: Frecuencia de goles por partido (LaLiga).

En el gráfico anterior (5) se puede ver cual es la cantidad de goles más frecuente por partido (en proporción) y, es bastante claro, que cumple muy bien con la distribución de Poisson. Con esto en mente, podemos analizar el problema de otra forma, por medio de la diferencia de goles, esta nos da una perspectiva mejor, dado que la gama de 3 posibilidades (victoria, derrota y empate), se ve por medio de 3 características (positivo, negativo y neutro, respectivamente). Para esto, existe la distribución de Skellam (3.1.2.1) (como se mencionó en el marco teórico). Para ello se tiene el siguiente diagrama<sup>3</sup>

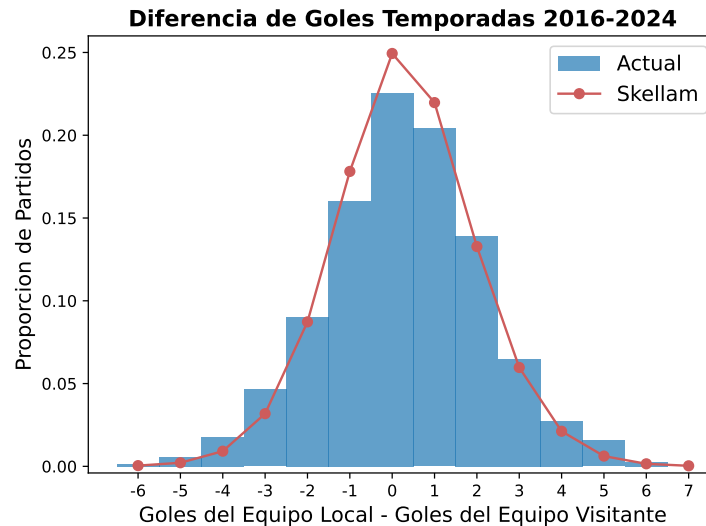


Figura 5: Frecuencia de diferencia de goles por partido (LaLiga).

#### 4.1.5. Creación del Modelo

Para el modelo descrito anteriormente se creo lo siguiente:

```
1 modelo_poisson = smf.glm(formula="goals~home+team+opponent", data=
  =modelo_goles, family=sm.families.Poisson()).fit()
```

Donde se utiliza la librería statsmodels, en donde:

- Esta línea crea un objeto de modelo lineal generalizado (GLM) usando la clase statsmodels.formula.api.GLM
- La fórmula 'goals ~ home + team + opponent' especifica la estructura del modelo:
  - goals es la variable dependiente (lo que se intenta predecir).
  - home, team y opponent son las variables independientes y la tilde ~ separa las variables independientes de las dependientes.
- data=modelo\_goles es un 'dataframe' que incluye la información mencionada anteriormente en forma de tabla.
- family=sm.families.Poisson() especifica que el modelo usa la distribución de Poisson.
- El método .fit() ajusta el modelo a los datos en modelo\_goles. Esto implica estimar los coeficientes para cada variable independiente y evaluar el ajuste general del modelo.

<sup>3</sup>Tanto en este diagrama como en el anterior, se muestran únicamente el de LaLiga dado que es redundante mostrar diagramas prácticamente iguales; sin embargo, se puede revisar en el GitHub del proyecto.

En términos más simples, este código crea un modelo estadístico para predecir la cantidad de goles anotados (**goals**) basado en factores como si el equipo juega en casa (**home**), el equipo en sí (**team**) y el oponente (**opponent**). Utiliza una regresión de Poisson porque se trata de datos de conteo (goles).

Teniendo el modelo ya creado, se utiliza para calcular el promedio de goles por equipo (local y visitante)

```
1 home_goals_avg = model.predict(pd.DataFrame(data={'team': homeTeam, 'opponent': awayTeam, 'home':1}, index=[1])).values[0]
2 away_goals_avg = model.predict(pd.DataFrame(data={'team': awayTeam, 'opponent': homeTeam, 'home':0}, index=[1])).values[0]
3 team_pred = [[poisson.pmf(i, team_avg) for i in range(0, max_goals+1)] for team_avg in [home_goals_avg, away_goals_avg]]
```

y, posteriormente, se calcula la probabilidad de tener cada uno de los posibles resultados hasta 10 goles. Esto, utilizando lo siguiente para calcular la probabilidad de tener empate

$$P(\text{Empate}) = P(0|\text{Local}) \times P(0|\text{Visitante}) + P(1|\text{Local}) \times P(1|\text{Visitante}) + \dots \quad (4.1.5.1)$$

Y así, para calcular la probabilidad de ganar o perder por cierta cantidad de goles.

Con esto, ya se tiene el modelo de predicción, únicamente teniendo el enfrentamiento, el modelo nos da la probabilidad de tener al local como ganador, al visitante como ganador o el empate. Además se tiene una matriz, en la cual las columnas son los resultados del equipo visitante, las filas los del local y la diagonal los empates.

## 5. Resultados

Utilizando el modelo creado, se puede realizar las pruebas para los enfrentamientos deseados. En nuestro caso se tienen dos enfrentamientos para realizar las pruebas y generar las respectivas matrices. Se realizan las predicciones para 10 goles, pero solo se muestran hasta 4 para que se entienda el concepto y mostrar las más relevantes.

| Equipos         |       | FC Barcelona |       |       |       |       |
|-----------------|-------|--------------|-------|-------|-------|-------|
|                 | Goles | 0            | 1     | 2     | 3     | 4     |
| Athletic Bilbao | 0     | 0.070        | 0.115 | 0.094 | 0.052 | 0.021 |
|                 | 1     | 0.071        | 0.117 | 0.096 | 0.053 | 0.022 |
|                 | 2     | 0.036        | 0.059 | 0.049 | 0.027 | 0.011 |
|                 | 3     | 0.012        | 0.020 | 0.017 | 0.009 | 0.004 |
|                 | 4     | 0.003        | 0.005 | 0.004 | 0.002 | 0.001 |

Cuadro 2: Matriz de Probabilidad entre Athletic de Bilbao v FC Barcelona.

| Equipos         |       | Manchester United |       |       |       |       |
|-----------------|-------|-------------------|-------|-------|-------|-------|
|                 | Goles | 0                 | 1     | 2     | 3     | 4     |
| Manchester City | 0     | 0.047             | 0.042 | 0.019 | 0.006 | 0.001 |
|                 | 1     | 0.102             | 0.091 | 0.041 | 0.012 | 0.003 |
|                 | 2     | 0.110             | 0.098 | 0.044 | 0.013 | 0.003 |
|                 | 3     | 0.079             | 0.071 | 0.032 | 0.009 | 0.002 |
|                 | 4     | 0.043             | 0.038 | 0.017 | 0.005 | 0.001 |

Cuadro 3: Matriz de Probabilidad entre Manchester City v Manchester United.

Para el enfrentamiento entre el Athletic de Bilbao y el FC Barcelona los resultados más probables (en formato goles local-goles visita) son: 1-1, 0-1, 1-2, 0-2, 1-0 y 0-0. Todas con probabilidades del 7 % o más. Para el enfrentamiento entre el Manchester City y el Manchester United los resultados más probables son: 2-0, 1-0, 2-1, 1-1, 3-0 y 3-1. Y viendo los resultados del día de hoy (jornada 27), ATH (Athletic de Bilbao) 0 - 0 FCB (FC Barcelona) y MCI (Manchester City) 3 - 1 MNU (Manchester United).

Las probabilidades para los enfrentamientos de esta semana (jornada 27) en La Liga y Premier League son:

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| Celta                 | Almeria                   | Empate   |
| 0.582883              | 0.207118                  | 0.209988 |
| Sevilla               | Sociedad                  | Empate   |
| 0.451223              | 0.287662                  | 0.261114 |
| Vallecano             | Cadiz                     | Empate   |
| 0.477888              | 0.236709                  | 0.285404 |
| Getafe                | Las Palmas                | Empate   |
| 0.554268              | 0.180273                  | 0.265459 |
| Valencia              | Real Madrid               | Empate   |
| 0.235566              | 0.530057                  | 0.234374 |
| Villarreal            | Granada                   | Empate   |
| 0.669479              | 0.143358                  | 0.18714  |
| Ath Madrid            | Betis                     | Empate   |
| 0.66134               | 0.129458                  | 0.209196 |
| Mallorca              | Girona                    | Empate   |
| 0.323353              | 0.4141                    | 0.262547 |
| Ath Bilbao            | Barcelona                 | Empate   |
| 0.234433              | 0.519765                  | 0.2458   |
| Osasuna               | Alaves                    | Empate   |
| 0.430338              | 0.286165                  | 0.283497 |

Cuadro 4: Predicción Jornada 27 La Liga según Modelo de Poisson

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| Newcastle             | Wolves                    | Empate   |
| 0.44046               | 0.280533                  | 0.279007 |
| Tottenham             | Crystal Palace            | Empate   |
| 0.6665                | 0.139883                  | 0.193602 |
| Nott'm Forest         | Liverpool                 | Empate   |
| 0.121262              | 0.703653                  | 0.175052 |
| Brentford             | Chelsea                   | Empate   |
| 0.320771              | 0.42535                   | 0.253878 |
| Everton               | West Ham                  | Empate   |
| 0.43065               | 0.313374                  | 0.255975 |
| Fulham                | Brighton                  | Empate   |
| 0.351262              | 0.374122                  | 0.274616 |
| Luton                 | Aston Villa               | Empate   |
| 0.345772              | 0.431095                  | 0.223129 |
| Burnley               | Bournemouth               | Empate   |
| 0.427936              | 0.309256                  | 0.262808 |
| Man City              | Man United                | Empate   |
| 0.662287              | 0.144342                  | 0.193354 |
| Sheffield United      | Arsenal                   | Empate   |
| 0.155356              | 0.623088                  | 0.221553 |

Cuadro 5: Predicción Jornada 27 Premier League según Modelo de Poisson

Mientras que los resultados reales de estos encuentros son (L: Victoria Local, V: Victoria Visitante y E: Empate):

| Enfrentamientos    |               | Resultado | Victoria/Empate |
|--------------------|---------------|-----------|-----------------|
| Celta              | Almería       | 1-0       | L               |
| Sevilla            | Real Sociedad | 3-2       | L               |
| Rayo Vallecano     | Cádiz         | 1-1       | E               |
| Getafe             | Las Palmas    | 3-3       | E               |
| Valencia           | Real Madrid   | 2-2       | E               |
| Villarreal         | Granada       | 5-1       | L               |
| Atlético de Madrid | Betis         | 2-1       | L               |
| Mallorca           | Girona        | 1-0       | L               |
| Athletic de Bilbao | FC Barcelona  | 0-0       | E               |
| Osasuna            | Alavés        | 1-0       | L               |

Cuadro 6: Resultados Jornada 27 La Liga

| Enfrentamiento    |                   | Resultado | Victoria/Empate |
|-------------------|-------------------|-----------|-----------------|
| Newcastle         | Wolves            | 3-0       | L               |
| Tottenham         | Crystal Palace    | 3-1       | L               |
| Nottingham Forest | Liverpool         | 0-1       | V               |
| Brentford         | Chelsea           | 2-2       | E               |
| Everton           | West Ham          | 1-3       | V               |
| Fulham            | Brighton          | 3-0       | L               |
| Luton             | Aston Villa       | 2-3       | V               |
| Burnley           | Bournemouth       | 0-2       | V               |
| Manchester City   | Manchester United | 3-1       | L               |
| Scheffield United | Arsenal           | 0-6       | V               |

Cuadro 7: Resultados Jornada 27 Premier League

Comparando ambas tablas, se tiene que para La Liga, se acertaron 5/10 y mientras que para la Premier League se acertaron 6/10.

Ahora, realizando el mismo procedimiento para la UEFA Champions League. Desafortunadamente, en esta competencia se realizan sorteos para la fase de 8vos de final y 4tos de final, por ende no se tiene una llave específica de la cual podamos tomar una guía. Sin embargo, dada la aleatoriedad del sorteo también se puede modelar. Por lo que, en base a las probabilidades por partido, se tendrán las diferentes eliminatorias hasta llegar a la gran final y mostrar al ganador.

Los resultados de la vuelta de los octavos de final pueden ser difíciles de predecir, hay probabilidades muy altas por un equipo y resultados en la ida muy diferentes a estas probabilidades. Pero, en este caso, tomaremos el resultado por goles más probable:

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| Arsenal FC            | FC Porto                  | Empate   |
| 0.700038              | 0.152832                  | 0.146703 |
| Atlético Madrid       | Inter                     | Empate   |
| 0.407639              | 0.256554                  | 0.335807 |
| Bayern München        | Lazio Roma                | Empate   |
| 0.802892              | 0.0757905                 | 0.121    |
| Borussia Dortmund     | PSV Eindhoven             | Empate   |
| 0.648557              | 0.152485                  | 0.198945 |
| FC Barcelona          | SSC Napoli                | Empate   |
| 0.613517              | 0.171231                  | 0.215246 |
| Manchester City       | FC København              | Empate   |
| 0.256067              | 0.364221                  | 0.379712 |
| Real Madrid           | RB Leipzig                | Empate   |
| 0.71542               | 0.130634                  | 0.153787 |

Cuadro 8: Predicción 8vos de Final Vuelta UCL

Sin embargo, el resultado más probable es el que tomaremos en cuenta, para los partidos de esta ronda de la UEFA Champions League son:

| Equipo Local      | Equipo Visitante | Resultado |
|-------------------|------------------|-----------|
| Arsenal FC        | FC Porto         | 3-1       |
| Atlético Madrid   | Inter            | 0-0       |
| Bayern München    | Lazio Roma       | 3-0       |
| Borussia Dortmund | PSV Eindhoven    | 2-0       |
| FC Barcelona      | SSC Napoli       | 1-0       |
| Manchester City   | FC København     | 0-0       |
| Real Madrid       | RB Leipzig       | 2-1       |

Cuadro 9: Resultado más probable en los 8vos de Final Vuelta UCL

Los equipos marcados en azul son los que, con el resultado global, pasarán de ronda<sup>4</sup>. Para las siguientes predicciones se realizará el sorteo aleatorio<sup>5</sup> y se tomarán únicamente 1 eliminatoria (representando el resultado de esta como el global). Para los cuartos de final se tiene:

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| Arsenal FC            | Manchester City           | Empate   |
| 0.485363              | 0.327039                  | 0.187514 |
| Bayern München        | Real Madrid               | Empate   |
| 0.590319              | 0.212202                  | 0.197453 |
| FC Barcelona          | Borussia Dortmund         | Empate   |
| 0.633024              | 0.171465                  | 0.195491 |
| Inter                 | Paris Saint-Germain       | Empate   |
| 0.236869              | 0.512878                  | 0.250252 |

Cuadro 10: Predicción 4tos de Final UCL

Dadas estas probabilidades generales, se tienen los siguientes resultados más probables:

| Equipo Local   | Equipo Visitante    | Resultado |
|----------------|---------------------|-----------|
| Arsenal FC     | Manchester City     | 2-2       |
| Bayern München | Real Madrid         | 2-1       |
| FC Barcelona   | Borussia Dortmund   | 2-1       |
| Inter          | Paris Saint-Germain | 0-1       |

Cuadro 11: Resultado más probable en los 4vos de Final UCL

Para el Arsenal v Manchester City, se tomó el segundo resultado más probable, el cual es 1 – 2, victoria para el Manchester City. Ahora, modelando los enfrentamientos de semifinales<sup>6</sup> y su resultado:

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| FC Barcelona          | Manchester City           | Empate   |
| 0.451932              | 0.313684                  | 0.234382 |
| Paris Saint-Germain   | Bayern München            | Empate   |
| 0.364065              | 0.407526                  | 0.228406 |

Cuadro 12: Predicción Semi-Finales UCL

<sup>4</sup>Es claro que falta un enfrentamiento, este es el del Paris Saint-Germain contra la Real Sociedad. Este enfrentamiento tiene una peculiaridad, la Real Sociedad no tiene presedentes en la UCL, y el partido de ida tuvo un resultado de 2 – 0 a favor del PSG, por lo que se le dió por ganador.

<sup>5</sup>Se realizó con el método `random.shuffle(array)`, con la semilla 33.

<sup>6</sup>Semilla 43



Cuyos resultados más probables son:

| Victoria Equipo Local | Victoria Equipo Visitante | Resultado |
|-----------------------|---------------------------|-----------|
| FC Barcelona          | Manchester City           | 1-1       |
| Paris Saint-Germain   | Bayern München            | 1-1       |

Cuadro 13: Resultado más probable en las Semi-Finales UCL

Para ambos partidos se tomó el segundo resultado más probable, en cada caso es FC Barcelona v Manchester United 2 – 1 y para Paris Saint-Germain v FC Bayern 1 – 2. Con esto la final, sorprendente mente es entre Barcelona y Bayern München. Los resultados más probables de la misma son: (el sorteo<sup>7</sup> se realiza únicamente para escoger al equipo local)

| Victoria Equipo Local | Victoria Equipo Visitante | Empate   |
|-----------------------|---------------------------|----------|
| FC Barcelona          | Bayern München            | Empate   |
| 0.304152              | 0.464494                  | 0.231351 |

Cuadro 14: Predicción FINAL UCL 2023-24

Cuyo resultado más probable es:

| Victoria Equipo Local | Victoria Equipo Visitante | Resultado |
|-----------------------|---------------------------|-----------|
| FC Barcelona          | Bayern München            | 1-1       |

Cuadro 15: Predicción FINAL UCL 2023-24

Nuevamente un empate es el resultado más probable, con esto, vemos el segundo más probable y tenemos que el FC Bayern München es el Campeón de la UEFA Champions League 2023-24 y empata al AC Milán con 7 campeonatos continentales, venciendo al FC Barcelona por un 1-2.

## 6. Discusión de Resultados

1. Dadas las características del modelo los resultados más comunes son los de victoria, ya sea por el equipo local o visitante. Esto influye directamente en la predicción del modelo, ya que el empate como resultado más probable es muy poco probable; por ende, en la gran mayoría de partidos que finalizan en empate, no se tendrá una predicción acertada. Esto, viendo la probabilidad máxima en los diferentes posibles resultados puede ser diferente. Por ello se realizaron estas búsquedas en el análisis de la UEFA Champions League.
2. Las limitaciones del modelo son claras, solo se toman en cuenta dos variables, los goles del local y los del visitante. Sin embargo, si se quiere realizar un modelo más eficiente se puede utilizar el modelo basado en distribuciones de Poisson bivariadas [1] o modelos de deep learning y aprendizaje supervisado, como la regresión logística, redes neuronales o catboost. Para ello se podrían tomar en cuenta muchas más variables, como el estadio, la asistencia al partido, historial de enfrentamientos y la fuerza de ataque y defensa [2]. Ojo que para ello se requiere de un mejor conjunto de datos.

## 7. Conclusiones

1. La revisión exhaustiva de la literatura académica y científica relacionada con la teoría estadística aplicada a la predicción de resultados deportivos ha proporcionado una visión profunda y

<sup>7</sup>Semilla 53.

comprensiva de los métodos, técnicas y modelos utilizados en este campo. A través de la identificación y recopilación de artículos, estudios y publicaciones relevantes, hemos obtenido una sólida comprensión de los fundamentos teóricos que sustentan los programas de predicción en el ámbito deportivo.

2. La recopilación de datos para llevar a cabo simulaciones y análisis estadísticos se realizó de manera exhaustiva y meticulosa, garantizando la selección de fuentes confiables de información deportiva, la identificación de variables relevantes y la obtención de datos históricos precisos. Este proceso ha sentado las bases para llevar a cabo análisis estadísticos sólidos y fundamentados en el ámbito deportivo, permitiendo así obtener resultados fiables y significativos para su posterior interpretación y aplicación en la predicción de resultados deportivos.
3. La utilización de los datos recopilados y la teoría revisada ha sido fundamental para realizar simulaciones y experimentos en el ámbito de la predicción de resultados deportivos. Aplicando los métodos estadísticos y probabilísticos aprendidos, en concreto, un modelo lineal generalizado basado en la distribución de Poisson. Estas simulaciones computacionales han permitido evaluar la efectividad y precisión de los programas de predicción desarrollados, proporcionando una visión clara sobre su desempeño en diferentes escenarios deportivos.

## 8. Apéndice

A continuación se muestra la clase creada para realizar las diferentes pruebas mostradas anteriormente. Los conjuntos de datos y las diferentes pruebas se pueden encontrar en el siguiente [drive](#) y las diferentes pruebas se encuentran en el siguiente [Google Colab](#), aquí se pueden encontrar las diferentes gráficas realizadas y algunas tablas no mostradas por su insignificancia en el informe.

## Clase PREDICT

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import glob
import warnings

# SCIPY
from scipy.stats import poisson, skellam
from tabulate import tabulate
# SCIKIT LEARN
# label encoder
from sklearn.preprocessing import LabelEncoder
# Statsmodels
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.iolib.summary import Summary

warnings.filterwarnings(action='ignore')

content = '/content/drive'
from google.colab import drive, files
drive.mount(content)
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

```
[ ]: class predict:
    def __init__(self) -> None:
        self.content = '/content/drive'
        self.folder_data = str
        self.division = str
        self.data = pd.DataFrame

    def look_up(self, folder_data : str, division : str, columns : list, show_info_
    ↪ False) -> pd.DataFrame:
        self.folder_data = folder_data
        patron = '*.xls*'
        nombres_archivos = []
        # columnas = ['Div', 'Date', 'HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'FTR',
    ↪ 'HTHG', 'HTAG', 'HS', 'AS', 'HST', 'AST']
        for archivo in glob.glob(os.path.join(self.content + '/' + folder_data_
    ↪ , patron)):
            nombre_archivo = os.path.basename(archivo)
            nombres_archivos.append(nombre_archivo)
```

```

nombres_archivos.sort()

dfs = []
#for i in range(12, len(nombres_archivos)):
for i in range(len(nombres_archivos)):
    df = pd.read_excel('/content/drive/' + folder_data + nombres_archivos[i],
→sheet_name=division, usecols=columns)
    dfs.append(df)

data = dfs[0]
for i in range(1, len(dfs)):
    data = pd.concat([data, dfs[i]], ignore_index=True)

if show_info:
    print(data.info())

self.data = data
return data

def distribution(self, show_histo = False, show_description = True) -> tuple:
    data = self.data
    data['FTR'].value_counts().plot(kind='bar')
    plt.title('Frecuencia de Victorias (Local, Visita y Empate)')
    plt.xlabel('Resultado')
    plt.ylabel('Frecuencia')

    plt.savefig('/content/histogram.pdf', format='pdf')
    files.download('/content/histogram.pdf')
    if show_histo:
        plt.show()
    #data.value_counts(data['FTR'])
    cat_cols = data.select_dtypes(include=['datetime64[ns]', 'object']).columns
    nums_cols = data.select_dtypes(exclude=['datetime64[ns]', 'object']).columns

    if show_description:
        print(data[nums_cols].describe())
    return nums_cols, cat_cols

def correlation_plot(self, full_description = False, show_corr = False) ->
→None:
    data = self.data
    data = data.copy()
    for col in data.select_dtypes(include='O'):
        data[col] = LabelEncoder().fit_transform(data[col])
    plt.figure(figsize=(10,8))

```

```

if full_description:
    cmap = sns.diverging_palette(245, 15, as_cmap=True)
    sns.heatmap(data.corr(), cmap=cmap, center=0, annot=True, linewidths=.5)
else:
    sns.heatmap(data.corr(), cmap='Reds')

def goals_info(self, title : str, show_histogram = False) -> None:
    spld = self.data[['HomeTeam', 'AwayTeam', 'FTHG', 'FTAG']]
    spld = spld.rename(columns={'FTHG' : 'HomeGoals', 'FTAG' : 'AwayGoals'})

    # para cada valor medio de goles
    poisson_pred = np.column_stack([[poisson.pmf(i, spld.mean()[j]) for i in
    ↪range(8)] for j in range(2)])

    plt.hist(spld[['HomeGoals', 'AwayGoals']].values, range(9),
             alpha=0.7, label=['Home', 'Away'], density=True, color=["xkcd:carolina_
    ↪blue", "xkcd:pistachio"])

    # add lines for the Poisson distributions
    pois1, = plt.plot([i-0.5 for i in range(1,9)], poisson_pred[:,0],
                      linestyle='-', marker='o',label="Home", color = 'Blue')
    pois2, = plt.plot([i-0.5 for i in range(1,9)], poisson_pred[:,1],
                      linestyle='-', marker='o',label="Away", color = '#006400')

    leg=plt.legend(loc='upper right', fontsize=13, ncol=2)
    leg.set_title("Poisson          Actual          ", prop = {'size':'14',
    ↪'weight':'bold'})

    plt.xticks([i-0.5 for i in range(1,9)], [i for i in range(8)])
    plt.xlabel("Goles por Partido",size=13)
    plt.ylabel("Proporcion de Partidos",size=13)
    # title = "Goles por Partido desde la Temporada 2005-06"
    plt.title(title,size=14,fontweight='bold')
    plt.ylim([-0.004, 0.4])
    plt.tight_layout()

    plt.savefig('/content/histogram_poisson.pdf', format='pdf')
    files.download('/content/histogram_poisson.pdf')
    if show_histogram:
        plt.show()
        plt.close()

    # Ahora utilizando la distribución de Skellam
    skellam_pred = [skellam.pmf(i, spld.mean()[0], spld.mean()[1]) for i in
    ↪range(-6,8)]

```

```

plt.hist(spld[['HomeGoals']].values - spld[['AwayGoals']].values,
→range(-6,8),
        alpha=0.7, label='Actual',density=True)
plt.plot([i+0.5 for i in range(-6,8)], skellam_pred,
        linestyle='-', marker='o',label="Skellam", color =
→'#CD5C5C')
plt.legend(loc='upper right', fontsize=13)
plt.xticks([i+0.5 for i in range(-6,8)], [i for i in range(-6,8)])
plt.xlabel("Goles del Equipo Local - Goles del Equipo Visitante",size=13)
plt.ylabel("Proporcion de Partidos",size=13)
plt.title("Diferencia de Goles Temporadas
→2016-2024",size=14,fontweight='bold')
plt.ylim([-0.004, 0.26])
plt.tight_layout()

plt.savefig('/content/histogram_skellam.pdf', format='pdf')
files.download('/content/histogram_skellam.pdf')
if show_histogram:
    plt.show()
plt.close()

self.model_data = spld

def model_generator(self):
    modelo_goles = pd.concat([self.model_data[['HomeTeam', 'AwayTeam',
→'HomeGoals']].assign(home=1).rename(
        columns={'HomeTeam':'team', 'AwayTeam':'opponent','HomeGoals':
→'goals'})],
        self.model_data[['AwayTeam', 'HomeTeam', 'AwayGoals']].
→assign(home=0).rename(
        columns={'AwayTeam':'team', 'HomeTeam':'opponent', 'AwayGoals':
→'goals'})])

    modelo_poisson = smf.glm(formula="goals ~ home + team + opponent",
→data=modelo_goles,
        family=sm.families.Poisson()).fit()

    return modelo_poisson

def simulate_match(self, model, homeTeam : str, awayTeam : str, max_goals=10)
→-> np.array:
    self.homeTeam = homeTeam
    self.awayTeam = awayTeam
    home_goals_avg = model.predict(pd.DataFrame(data={'team': homeTeam,

```

```

                                'opponent':␣
↪awayTeam, 'home':1},
                                index=[1])).values[0]
    away_goals_avg = model.predict(pd.DataFrame(data={'team': awayTeam,
                                'opponent':␣
↪homeTeam, 'home':0},
                                index=[1])).values[0]
    team_pred = [[poisson.pmf(i, team_avg) for i in range(0, max_goals+1)] for␣
↪team_avg in [home_goals_avg, away_goals_avg]]
    return(np.outer(np.array(team_pred[0]), np.array(team_pred[1])))

def probability(self, prob_matrix : np.array) -> pd.DataFrame:
    prob = pd.DataFrame({self.homeTeam : [np.sum(np.tril(prob_matrix, -1))],
                        self.awayTeam : [np.sum(np.triu(prob_matrix, 1))],
                        "Empate" : [np.sum(np.diag(prob_matrix))])})
    return prob

```

## Referencias

- [1] Karlis, Dimitris y Ioannis Ntzoufras: *Analysis of sports data by using bivariate Poisson models*. Journal of the Royal Statistical Society: Series D (The Statistician), 52(3):381–393, 2003.
- [2] Canela Ribas, Javier: *Estimación de resultados deportivos mediante modelos lineales generalizados*. 2023.
- [3] Rincón, Luis: *Introducción a la probabilidad*. Universidad Nacional Autónoma de México, Facultad de Ciencias, 2014.
- [4] Developers, Numpy: *NumPy: User Guide*, 2023. <https://numpy.org/doc/>.
- [5] Developers, Matplotlib: *Matplotlib: Plotting with Python*, 2023. <https://matplotlib.org/>.
- [6] Developers, Pandas: *Pandas: Python Data Analysis Library*, 2023. <https://pandas.pydata.org/pandas-docs/stable/>.
- [7] Developers, SciPy: *SciPy: Open-source Software for Mathematics, Science, and Engineering*, 2023. <https://docs.scipy.org/doc/scipy/>.
- [8] Developers, Scikit learn: *scikit-learn: Machine Learning in Python*, 2023. <https://scikit-learn.org/stable/>.
- [9] Developers, Seaborn: *Seaborn: Visualization Library for Python*, 2023. <https://seaborn.pydata.org/>.