

Herramienta de predicción de partidos de la EPL



CENTRO
COLABORADOR



UNIVERSIDAD
NEBRIJA

Daniel González Millán



Índice

1. Problema a resolver
2. Objetivos
3. Metodología
4. Resultados
5. Conclusiones
6. Next Steps



Problema a resolver

Queremos desarrollar una herramienta que sea capaz de predecir los resultados de los partidos de la English Premier League (EPL).

Para ello, trataremos de acertar los partidos planteándolo desde dos problemas distintos:

- Sistema de Victoria - Empate - Derrota (W-D-L)
- Sistema de Victoria - Empate/Derrota (W-D/L)



Objetivos







- La herramienta debe ser capaz de descargarse y cargar en un Dataframe los datos que necesitemos para ejecutar los modelos.
- Usaremos dos modelos (Random Forest y XGboost) para determinar cuál de los dos modelos es mejor. Nuestra herramienta debe ser capaz de tener un porcentaje de acierto mucho mayor que el azar.
- Determinar qué tipos de variables para cuales favorecen más a nuestro modelo.



Web Scrapping

- Consiste en obtener datos de una página web mediante HTTP.
- Para ello hemos usado la página:

<https://fbref.com/en/comps/9/Premier-League-Stats>

Rk	Squad	MP	W	D	L	GF	GA	GD	Pts	Pts/MP	xG	xGA	xGD	xGD/90	Last 5
1	 Arsenal	14	12	1	1	33	11	+22	37	2.64	26.2	11.8	+14.3	+1.02	W D W W W
2	 Manchester City	14	10	2	2	40	14	+26	32	2.29	27.6	11.2	+16.4	+1.17	L W W W L
3	 Newcastle Utd	15	8	6	1	29	11	+18	30	2.00	24.3	14.3	+9.9	+0.66	W W W W W
4	 Tottenham	15	9	2	4	31	21	+10	29	1.93	24.1	16.5	+7.6	+0.50	L L W L W
5	 Manchester Utd	14	8	2	4	20	20	0	26	1.86	18.8	17.3	+1.5	+0.11	W D W L W
6	 Liverpool	14	6	4	4	28	17	+11	22	1.57	24.2	19.5	+4.7	+0.34	W L L W W



Dataset

Usaremos los resultados de los partidos de la Premier League desde la temporada 2017 - 2018 hasta la actual, que es la 2022 - 2023. Es decir, 5 temporadas y 15 jornadas, lo que hace un total de 4092 partidos.

	Date	Time	Comp	Round	Day	Venue	Result	GF	GA	Opponent	...	Dist	FK	PK	PKatt	xG	npG	npG/Sh	G-xG	npG-xG	Match Report
0	2022-08-05	20:00	Premier League	Matchweek 1	Fri	Away	W	2	0	Crystal Palace	...	14.6	1.0	0	0	1.0	1.0	0.10	0.0	0.0	Match Report
1	2022-08-13	15:00	Premier League	Matchweek 2	Sat	Home	W	4	2	Leicester City	...	13.0	0.0	0	0	2.7	2.7	0.16	1.3	1.3	Match Report
2	2022-08-20	17:30	Premier League	Matchweek 3	Sat	Away	W	3	0	Bournemouth	...	14.8	0.0	0	0	1.3	1.3	0.10	1.7	1.7	Match Report
3	2022-08-27	17:30	Premier League	Matchweek 4	Sat	Home	W	2	1	Fulham	...	15.5	1.0	0	0	2.6	2.6	0.12	-0.6	-0.6	Match Report
4	2022-08-31	19:30	Premier League	Matchweek 5	Wed	Home	W	2	1	Aston Villa	...	16.3	1.0	0	0	2.4	2.4	0.12	-0.4	-0.4	Match Report



Variables a utilizar

Variables que conocemos antes del partido:

- Round: Hace referencia a la jornada en la que se disputa el partido.
- Day: El día de la semana en el que se disputa el partido.
- Venue: Si el partido se disputa jugando en su estadio o en el del rival.
- Opponent: El nombre del rival
- Referee: Nombre del árbitro
- Season: Temporada en la que nos encontramos:
- Team: Equipo que juega el partido.



Variables a utilizar

Variables que conocemos después del partido:

- GF: Goles a favor
- GA: Goles en contra
- xG: Goles esperados a favor
- xGA: Goles esperados en contra
- Poss: Posesión del balón (porcentaje del tiempo total que un equipo tuvo el balón)
- Attendance: Asistencia al estadio
- Sh: Tiros realizados
- SoT: Tiros a puerta (aquellos cuya trayectoria inicial va hacia la portería)
- FK: Faltas tiradas.
- PK: Penaltis anotados
- PKatt: Penaltis ejecutados



Random Forests

- Tiene árboles de decisión como algoritmo base.
- Cada uno de estos árboles se entrena a partir de un subconjunto de datos usando un muestreo de tipo bootstrap.
- Cada uno de estos árboles se entrenará también con un subconjunto de las variables.

Usaremos este algoritmo primero y principalmente porque **no es un algoritmo lineal**.



Random Forests

Las ventajas que nos va a ofrecer este modelo son básicamente:

- Es un algoritmo que **no tiende a sobreajustar**.
- Es un algoritmo con un **gran rendimiento y muy rápido de calcular**, debido a que es una combinación de Decision Trees corriendo en paralelo.

La principal desventaja es que **no es interpretable**.



XGBOOST

- Es un algoritmo de ***boosting***
- Utiliza árboles de decisión en paralelo.
- Alto rendimiento

Su principal desventaja es que **no es interpretable**.



Métricas a utilizar

Accuracy Score:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$



Cálculo de Medias

¿Qué podemos hacer si queremos usar datos como el número de faltas, la posesión o los goles anotados?

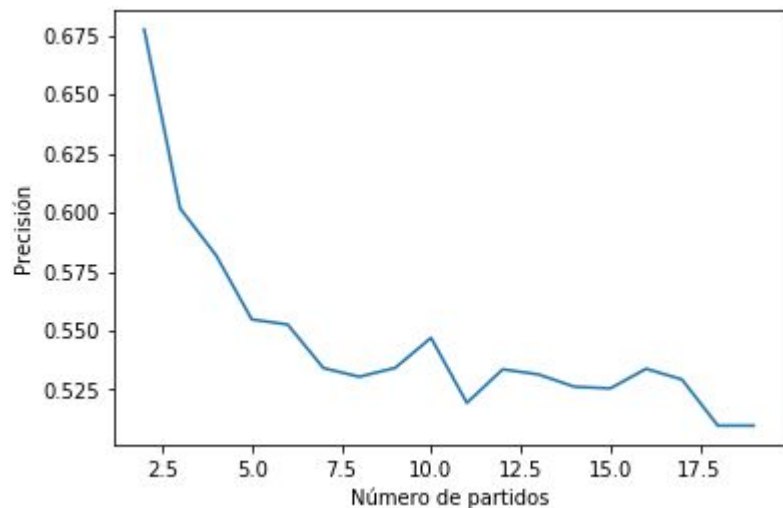
```
def calculo_medias(group, cols, new_cols, n):  
    group = group.sort_values("Date")  
    stats_medias = group[cols].rolling(n, closed='left').mean()  
    group[new_cols] = stats_medias  
    group = group.dropna(subset=new_cols)  
    return group
```

```
cols = ["GF", "GA", "Sh", "SoT", "Dist", "FK", "PK", "PKatt"]  
new_cols = [f"{c}_rolling" for c in cols]
```

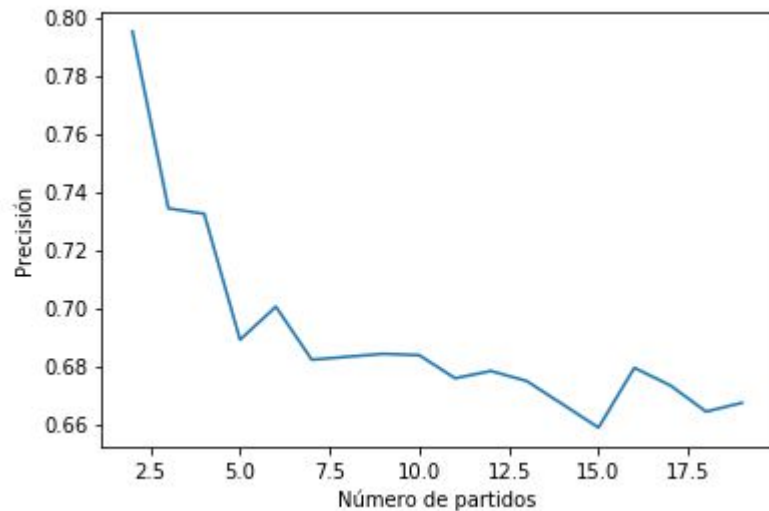


Cálculo de Medias

Para el sistema W-D-L:



Para el sistema W-D/L:





Resultados (Sistema W-D-L)

Precisión	Sin cálculo de medias	Con cálculo de medias
Random Forests	50,3%	68%
XGBoost	52,5%	69%



Resultados (Sistema W-D/L)

Precisión	Sin cálculo de medias	Con cálculo de medias
Random Forests	65,2%	80,1%
XGBoost	68,7%	80,2%



Conclusiones

1. Aunque se pueden hacer predicciones solamente usando datos de ítems que podemos saber antes del partido, los modelos obtenidos con este tipo de datos son bastante pobres.
2. Al usar datos de partidos, hemos observado que la mayor precisión la encontramos cuando usamos datos que tienen solamente una ventana de 1 partido anterior, mientras que, si vamos ampliando esta ventana, nuestra precisión va bajando.



Conclusiones

3. Aunque el modelo de W-D/L tiene una precisión mayor que el de W-D-L, teniendo en cuenta que, pese a haber simplificado el problema, haciéndolo pasar de dos a 3 opciones, solo hemos mejorado nuestra precisión un 12%.
4. Finalmente, hemos comparado dos algoritmos ensamblados en nuestro proyecto para nuestro problema, que son Random Forests y XGBoost, hemos visto que ambos ofrecen una precisión muy similar pero **Random Forests** se ejecuta de manera más rápida.



Next Steps

Cómo podríamos mejorar esta herramienta:

- Utilizar datos de otras competiciones.
- Utilizar una métrica que sea porcentaje de victorias contra los equipos de tabla alta y de tabla baja.
- Introducir una métrica que pondere con mayor peso los datos más recientes y con menor peso los datos más antiguos.
- Utilizar una base de datos con los jugadores.
- Incluir una serie de métricas en referencias al oponente.

Muchas gracias por su atención

