

Universidad Del Valle De Guatemala

Deep Learning

Javier Fong



**Proyecto Final: Predicción del resultado de partidos de la liga**

Javier Mombiola 20067

Roberto Ríos 20979

Guatemala, 22 de noviembre del 2023

## Índice

<b>Índice.....</b>	<b>1</b>
<b>Introducción.....</b>	<b>2</b>
<b>Antecedentes.....</b>	<b>3</b>
Contexto del Dataset.....	3
Modificaciones en el dataset.....	5
Tecnología Utilizada.....	6
<b>Análisis Exploratorio.....</b>	<b>7</b>
<b>Metodología.....</b>	<b>10</b>
Modelo 1: Red Neuronal Recurrente (RNN) con Capa SimpleRNN.....	11
Modelo 2: Red Neuronal con Capas Densas y Promedios de Datos.....	12
Modelo 3: Red Neuronal Recurrente con Capas LSTM.....	12
Modelo 4: Red Neuronal Recurrente con Capa SimpleRNN.....	13
Modelo 5: Red Neuronal con Capas Densas para Predicción del FC Barcelona.....	13
Observaciones Adicionales.....	14
<b>Resultados.....</b>	<b>15</b>
Resultados de los modelos.....	15
Resumen de los resultados.....	17
Predicciones de los modelos.....	17
<b>Discusión.....</b>	<b>18</b>
Resultados obtenidos.....	19
Significado de los resultados.....	19
¿Son los mejores resultados?.....	20
Descubrimiento sobre los resultados.....	20
<b>Conclusiones.....</b>	<b>21</b>
<b>Apéndice.....</b>	<b>22</b>
Implementaciones de los modelos.....	22
<b>Bibliografía.....</b>	<b>24</b>

## Introducción

Como proyecto final, nos embarcamos en la creación de una serie de modelos diseñados para predecir con precisión los resultados de los partidos en la Liga BBVA. Cada modelo presenta características y estructuras distintas, aunque comparten el mismo objetivo, con la excepción del último modelo. Esta variedad fue estratégicamente seleccionada para facilitar una comparación exhaustiva y determinar cuál de ellos se destacaba como el más eficaz en la tarea de predicción.

El quinto y último modelo introduce un giro único en la temática, al enfocarse no en predecir los resultados generales de un partido, sino en anticipar específicamente si el F.C. Barcelona obtendrá la victoria en un encuentro. Este enfoque diferente nos brinda una dimensión adicional a nuestra investigación y diversifica las aplicaciones potenciales de los modelos desarrollados.

Estos modelos se crearon con Tensor Flow (keras), y se entrenaron con un dataset combinado de varias décadas de datos de la liga española, estos datasets se obtuvieron de football-data.co.uk (ver bibliografía) y se limpiaron y manejaron acorde a las necesidades de cada modelo usando Pandas. Los resultados van bastante acorde a la realidad, ya que el modelo predice resultados similares a la realidad actual, como por ejemplo que el Girona es posible candidato a ganar la liga este año, o que el Barcelona, Real Madrid y Atlético de Madrid son siempre los primeros tres puestos.

Decidimos abarcar este tema, porque ambos somos fanáticos del fútbol. Toda la vida hemos jugado el deporte y siempre hemos sido seguidores de la Liga BBVA y más específicamente del F.C. Barcelona. Por lo tanto, se nos hizo una muy buena idea mezclar la inteligencia artificial con la liga española, ya que esto mezcla dos áreas que nos interesan, y podría ser de utilidad para poder obtener información de la temporada actual, lo cual nos podría servir incluso para apostar. Predecir resultados de la liga española es beneficioso ya que existe mucho interés por parte de las distintas aficiones, a su vez, permite aprender de los patrones numéricos que se extraen de los datos históricos para tomarlos en consideración en estrategias e incluso apuestas. Desarrollar este proyecto con redes neuronales hace posible que el proyecto sea adaptable a los cambios que se producen en el rendimiento de los equipos y los factores que influyen en los resultados.

De este estudio se puede concluir que los modelos entrenados sobre sets de datos de partidos de fútbol son buenos para predecir victorias, sin embargo, los empates suelen confundirlos y afectar su accuracy. El posible ganador para este año es el Girona según el modelo 4 y el Barcelona según el modelo 1, el Almería según el modelo 3 y el Girona según el 4.

## Antecedentes

### Contexto del Dataset

El dataset utilizado en este proyecto es una recolección de las últimas 10 temporadas de la Liga BBVA (desde la temporada 2013-2014 hasta la 2022-2023). Para este proyecto, se descargaron dichas temporadas en *Football-Data.co.uk* en formato csv. Pero cada temporada venía en un archivo separado, por lo cual hicimos un script en python que pudiera hacerle merge a las 10 temporadas en un solo archivo. Adicionalmente, se eliminaron varias columnas originales, ya que estos archivos tenían más de 100 columnas, siendo la mayoría sobre información de apuestas, por lo que no las consideramos importantes y las descartamos.

*Script para unir las 10 temporadas y descartar columnas innecesarias*

```
1 import pandas as pd
2
3 # Lista de archivos a unir
4 files = [
5     '13-14.csv', '14-15.csv', '15-16.csv',
6     '16-17.csv', '17-18.csv', '18-19.csv',
7     '19-20.csv', '20-21.csv', '21-22.csv',
8     '22-23.csv'
9 ]
10
11 # Crear lista para guardar los DataFrames
12 dfs = []
13
14 # Recorrer la lista de archivos
15 for file in files:
16     df = pd.read_csv('temporadas/' + file)
17     dfs.append(df)
18
19 # Unir los DataFrames
20 df = pd.concat(dfs, ignore_index=True)
21
22 # Eliminar columnas que no se van a utilizar
23 columns_to_drop = [
24     'Div', 'B365H', 'B365D', 'B365A', 'BWH', 'BWD', 'BWA', 'IWH', 'IWD', 'IWA',
25     'LBH', 'LBD', 'LBA', 'PSH', 'PSD', 'PSA', 'WHH', 'WHD', 'WHA', 'SJH', 'SJD',
26     'SJA', 'VCH', 'VCD', 'VCA', 'Bb1X2', 'BbMxH', 'BbAvH', 'BbMxD', 'BbAvD', 'BbMxA',
27     'BbAvA', 'BbOU', 'BbMx>2.5', 'BbAv>2.5', 'BbMx<2.5', 'BbAv<2.5', 'BbAH', 'BbAHH',
28     'BbMxAHH', 'BbAvAHH', 'BbMxAHA', 'BbAvAHA', 'PSCH', 'PSCD', 'PSCA', 'MaxH', 'MaxD',
29     'MaxA', 'AvgH', 'AvgD', 'AvgA', 'B365CH', 'B365CD', 'B365CA', 'BWCH', 'BWCD', 'BWCA',
30     'IWCH', 'IWCD', 'IWCA', 'PSCH', 'PSCD', 'PSCA', 'WHCH', 'WHCD', 'WHCA', 'VCCH', 'VCCD',
31     'Time', 'B365>2.5', 'B365<2.5', 'P>2.5', 'P<2.5', 'Max>2.5', 'Max<2.5', 'Avg>2.5', 'Avg<2.5',
32     'AHH', 'B365AHH', 'B365AHA', 'PAHH', 'PAHA', 'MaxAHH', 'MaxAHA', 'AvgAHH', 'AvgAHA', 'VCCA',
33     'MaxCH', 'MaxCD', 'MaxCA', 'AvgCH', 'AvgCD', 'AvgCA', 'B365C>2.5', 'B365C<2.5', 'PC>2.5',
34     'PC<2.5', 'MaxC>2.5', 'MaxC<2.5', 'AvgC>2.5', 'AvgC<2.5', 'AHCh', 'B365CAHH', 'B365CAHA',
35     'PCAHH', 'PCAHA', 'MaxCAHH', 'MaxCAHA', 'AvgCAHH', 'AvgCAHA'
36 ]
37
38 df = df.drop(columns=columns_to_drop, axis=1)
39
40 # Guardar el DataFrame resultante en un archivo CSV
41 df.to_csv('temporadas.csv', index=False)
42
```

### Head del dataset resultante al ejecutar el script

```
1 Date,HomeTeam,AwayTeam,FTHG,FTAG,FTR,HTHG,HTAG,HTR,HS,AS,HST,AST,HF,AF,HC,AC,HY,AY,HR,AR
2 17/08/13,Sociedad,Getafe,2,0,H,1,0,H,16,15,6,2,13,6,6,5,1,1,0,0
3 17/08/13,Valencia,Málaga,1,0,H,0,0,D,9,11,1,2,15,23,9,6,3,5,0,0
4 17/08/13,Valladolid,Ath Bilbao,1,2,A,1,1,D,8,13,2,3,10,8,5,5,1,0,0,0
5 18/08/13,Barcelona,Levante,7,0,H,6,0,H,22,4,13,1,15,16,9,3,1,3,0,0
6 18/08/13,Osasuna,Granada,1,2,A,0,2,A,14,13,5,4,15,17,7,6,1,4,0,0
7 18/08/13,Real Madrid,Betis,2,1,H,1,1,D,20,11,9,4,11,20,5,7,1,2,0,0
```

Tabla 1: Información general del dataset

Descripción	Cantidad
Filas	3800
Columnas	21
Datos Nulos	0

Es importante mencionar que se hizo una codificación para ciertas columnas como; HomeTema, AwayTeam y FTR, ya que estas se pasaron a valores enteros para poder usarlas en los modelos. Las variables clave en el conjunto de datos final incluyen:

- Date: La fecha del partido.
- HomeTeam y AwayTeam: Los equipos locales y visitantes respectivamente.
- FTHG y FTAG: Los goles marcados por el equipo local y visitante, respectivamente, al final el partido.
- FTR: El resultado final del partido (H: Victoria del equipo local, A: Victoria del equipo visitante, D: Empate).
- HTHG y HTAG: Los goles marcados por el equipo local y visitante, respectivamente, al medio tiempo.
- HTR: El resultado final del partido al medio tiempo (H: Victoria del equipo local, A: Victoria del equipo visitante, D: Empate).

- Otras estadísticas: Variables adicionales como tiros a puerta (HS, AS), tiros al arco (HST, AST), faltas (HF, AF), córners (HC, AC), tarjetas (HY, AY), y expulsiones (HR, AR).

### Modificaciones en el dataset

Como se mencionó anteriormente, cada uno de los modelos toma diferentes características como input, por lo cual el dataset se modifica para cada uno de ellos.

El modelo 1, modelo 3 y el modelo 4 utilizan columnas extras similares. El modelo 1 utiliza todas las columnas que se muestran a continuación, el modelo 3 solo utiliza PH, PA, y el modelo 4 excluye HHHW, HHAW:

PH	PA	GDH	GDA	HHHW	HHAW
14	6	1	-8	7	6
15	19	5	6	6	10
12	6	-1	-11	2	1
12	5	-3	-12	1	2
14	6	-4	-7	4	1

- PH, PA: Puntos sumados de los últimos 10 partidos, para local y visita, respectivamente.
- GDH, GDA: Diferencia de goles de los últimos 10 partidos, para local y visita, respectivamente.
- HHHW, HHAW: Victorias head-to-head entre ambos equipos para local y visita, respectivamente.

El modelo 2 y el modelo 5 utilizan las siguientes columnas extras:

HST_avg	AST_avg	HF_avg	AF_avg	HC_avg	AC_avg	HY_avg	AY_avg	HR_avg	AR_avg
5.2	2.9	14.1	13.4	8.6	2.5	2.6	3.2	0.0	0.3
4.6	3.1	14.3	13.7	8.0	2.6	2.4	2.8	0.0	0.3
4.6	3.2	13.8	14.0	8.7	2.8	2.4	2.9	0.0	0.3
4.3	3.6	14.7	13.9	7.0	3.5	2.8	2.6	0.1	0.3
4.4	4.0	14.7	14.2	6.7	3.1	2.7	2.8	0.1	0.3

- Promedio de los últimos 10 partidos de las variables adicionales como tiros a puerta (HS, AS), tiros al arco (HST, AST), faltas (HF, AF), córners (HC, AC), tarjetas (HY, AY), y expulsiones (HR, AR).

El modelo 5 también tiene una nueva columna con la variable objetivo:

Resultado
1
0
0
1
0

- Resultado: Resultado del partido en donde participa el Barcelona, en donde 1 significa que gana el partido y 0 que no gana

## Tecnología Utilizada

En este proyecto, hemos empleado una variedad de técnicas y herramientas de aprendizaje profundo para explorar y modelar los datos:

**Keras y TensorFlow:** Utilizamos la biblioteca Keras, que se ejecuta sobre el motor de TensorFlow, para construir y entrenar nuestras redes neuronales. La flexibilidad y la eficiencia de TensorFlow nos permitieron implementar modelos complejos y experimentar con diversas arquitecturas.

**Redes Neuronales Recurrentes (RNN) y Capas Densas:** Exploramos modelos que van desde redes neuronales recurrentes (LSTM y SimpleRNN) hasta capas densas para abordar diferentes aspectos de la predicción de resultados.

**Pandas y NumPy:** Empleamos Pandas y NumPy para el preprocesamiento eficiente de datos, manipulación de variables categóricas y la creación de nuevas características.

**Scikit-learn:** Utilizamos Scikit-learn para la división de conjuntos de datos, la codificación de etiquetas, y la evaluación del rendimiento de los modelos.

Este enfoque tecnológico nos brindó la capacidad de implementar modelos y realizar análisis detallados sobre el comportamiento de los equipos en la Liga BBVA. En las secciones siguientes, detallaremos nuestras experiencias y los resultados obtenidos al aplicar estas tecnologías a nuestro conjunto de datos.

## Análisis Exploratorio

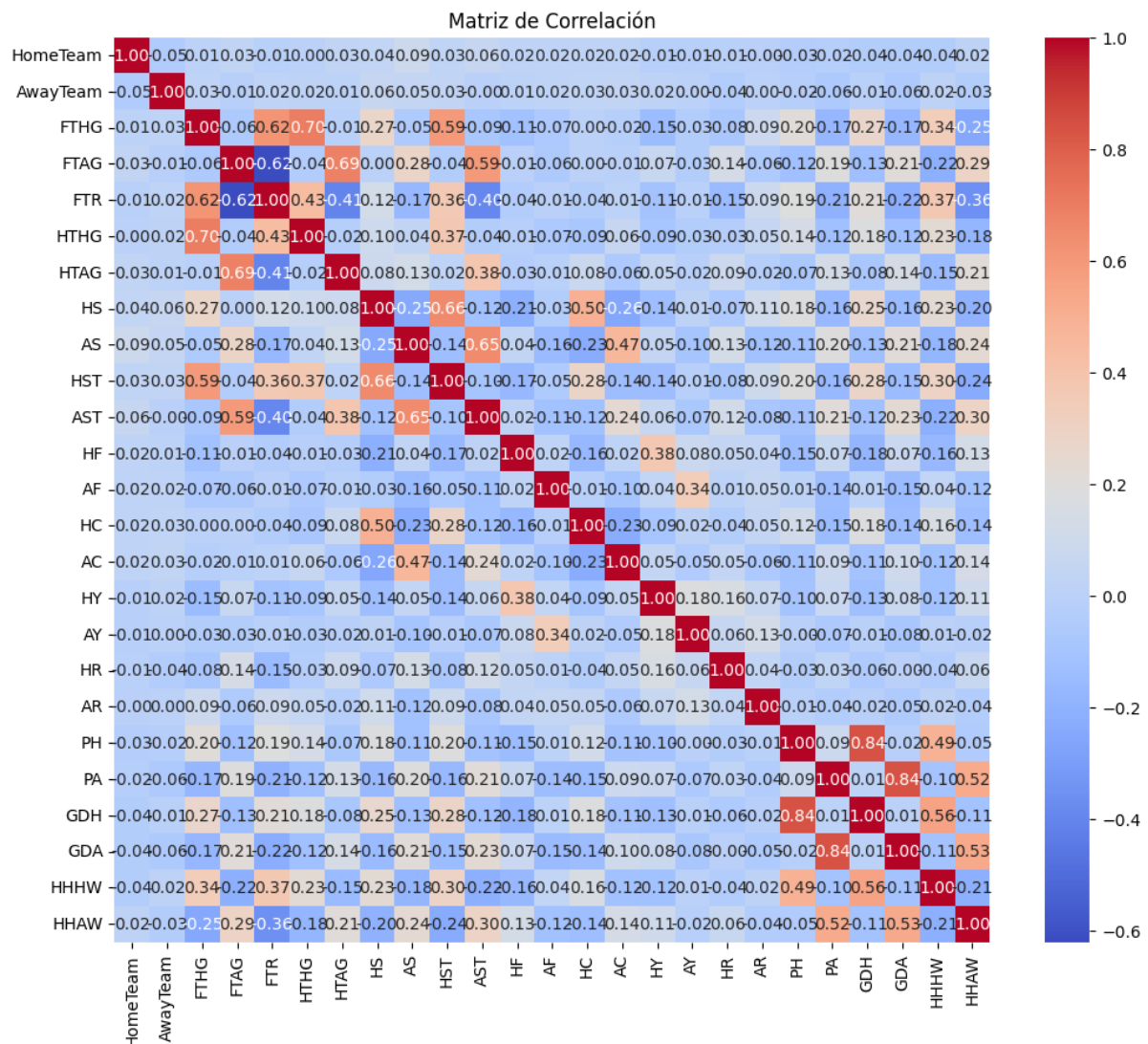
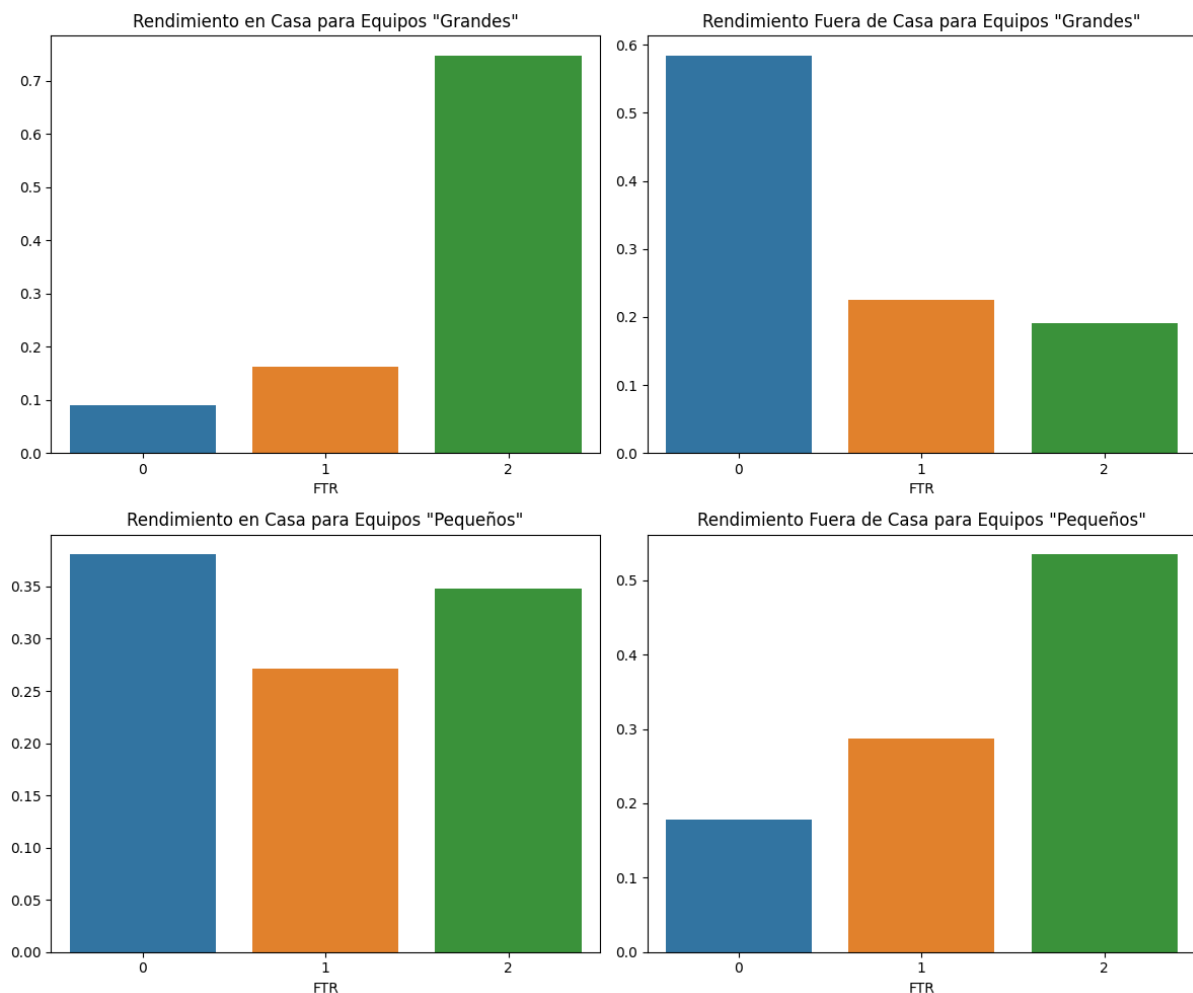


Gráfico 1: Matriz de Correlación

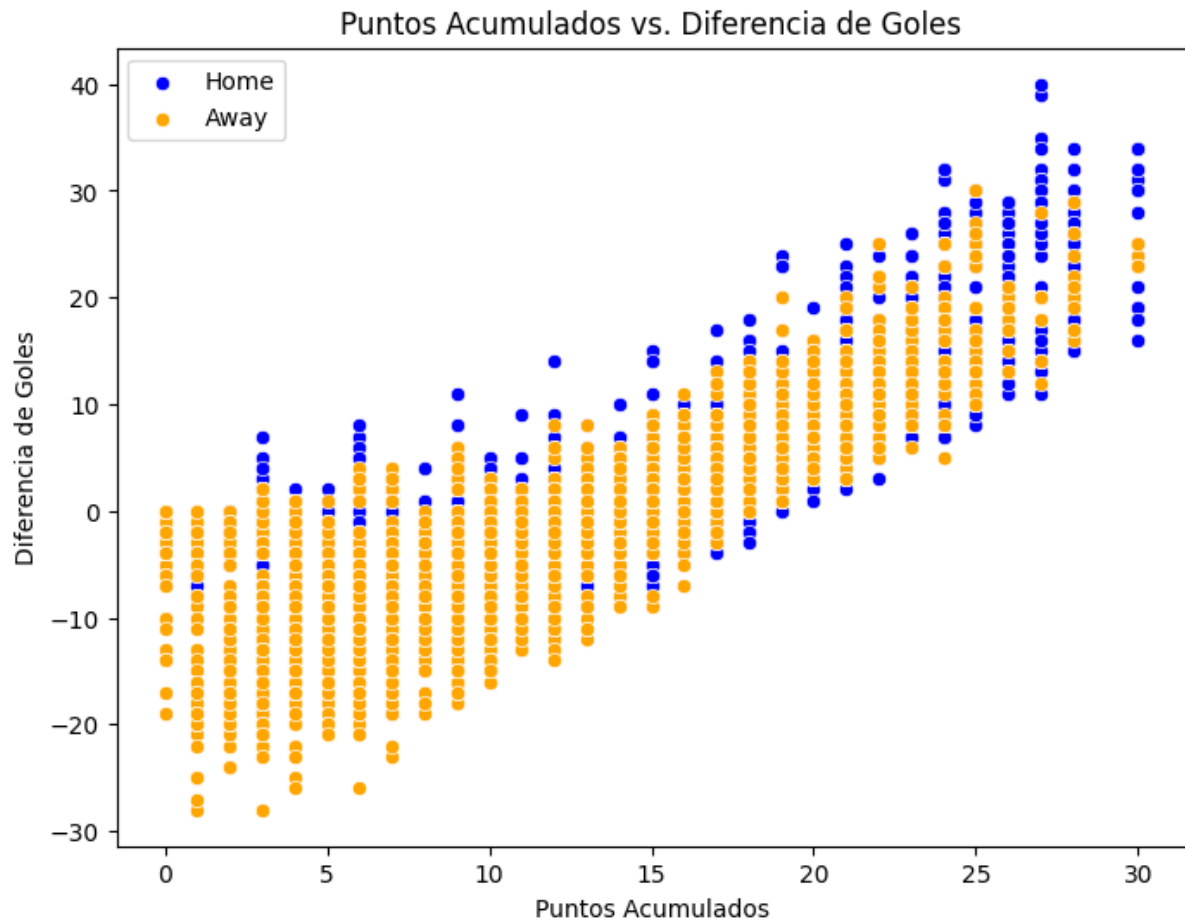


La matriz de correlaciones es una herramienta muy importante para poder elegir las características a utilizar en nuestros modelos. Tomando en cuenta que nuestra variable objetivo es “FTR” (Full Time Result), podemos observar que hay varias variables que se pueden utilizar para el modelo como; las estadísticas head-to-head, puntos a favor, diferencia de goles y otras estadísticas importantes como tiros a puerta y tiros al arco.



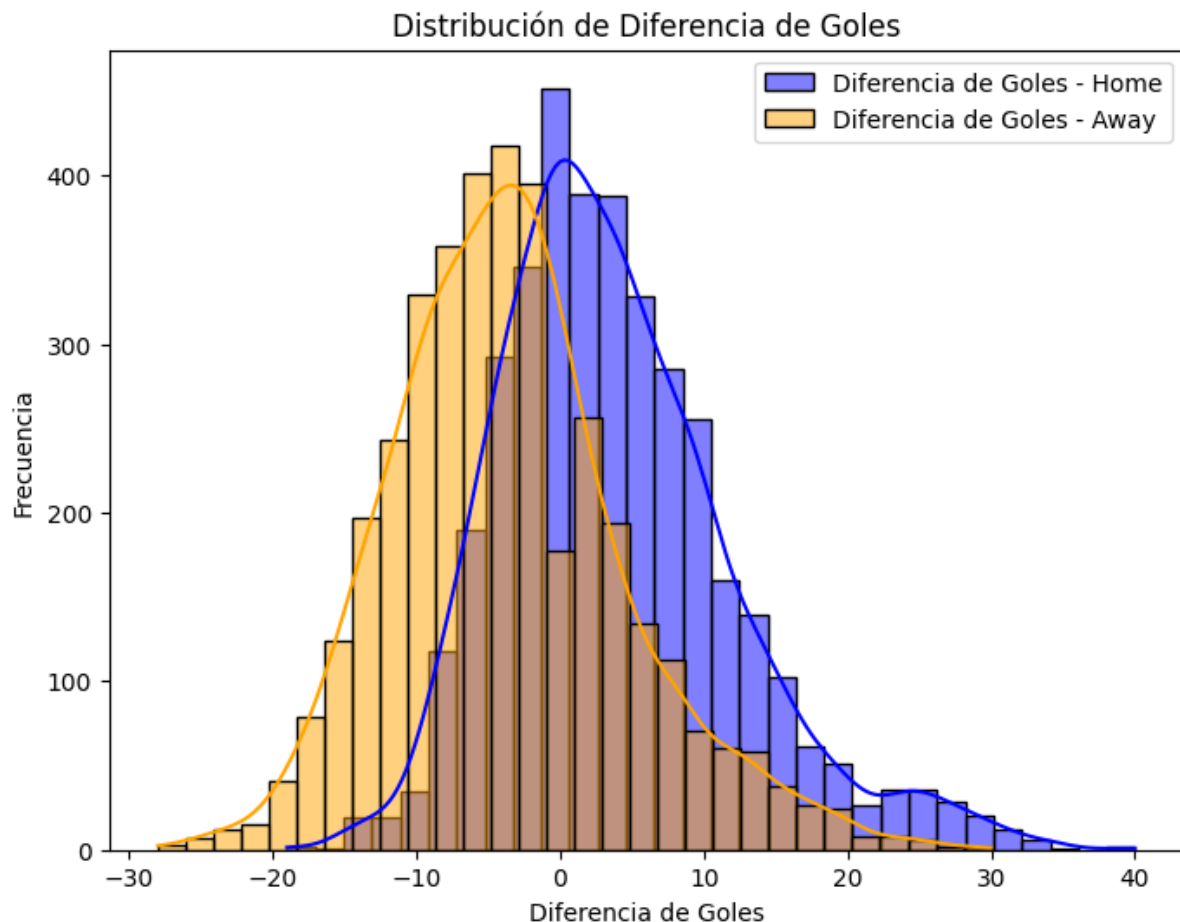
*Gráfico 2: Rendimiento de Equipos*

El gráfico de rendimiento de equipos es muy útil para poder observar el rendimiento de los equipos grandes (Barcelona, Madrid y Atlético) vs. el rendimiento de los equipos pequeños (Eibar, Leganes, Huescas). Y como podemos observar, los equipos grandes siempre tienden a tener más victorias que derrotas cuando están jugando de local y viceversa. Por el otro lado, los equipos pequeños sí tienen más derrotas cuando juegan de visitante, pero su rendimiento de local no mejora mucho.



*Gráfico 3: Diagrama de Dispersión, puntos vs. goles*

El diagrama de dispersión ilustra la relación entre dos variables, en este caso, puntos acumulados vs. diferencia de goles. Como podemos observar, este gráfico es importante, ya que nos demuestra que si hay una correlación entre puntos acumulados y diferencia de goles, entre mayor sea uno, mayor será el otro.



*Gráfico 4: Distribución de diferencia de goles*

El gráfico de distribución muestra la frecuencia de valores diferentes en una variable particular. En este caso la variable es la diferencia de goles, y podemos observar que si afecta si el equipo está jugando de local o visitante, ya que a la hora que juegan de local, tienden a anotar más goles.

## Metodología

En esta sección, proporcionaremos un desglose detallado de la implementación de la tecnología en cada uno de nuestros modelos de deep learning, destacando los parámetros e hiperparámetros clave utilizados. Cabe destacar que la división de los conjuntos de datos de entrenamiento y prueba se realizó de manera consistente para todos los modelos mediante la siguiente línea de código:

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=True)
```

Además, para los modelos que incorporaron capas recurrentes (1 y 4), se aplicó un reshape a los conjuntos de datos de entrada de la siguiente manera:

```
1 X_train = X_train.values.reshape((X_train.shape[0], 1, X_train.shape[1]))
2 X_test = X_test.values.reshape((X_test.shape[0], 1, X_test.shape[1]))
```

Estas decisiones metodológicas proporcionaron consistencia en la evaluación de los modelos y aseguraron la comparabilidad entre ellos.

### **Modelo 1: Red Neuronal Recurrente (RNN) con Capa SimpleRNN**

El primer modelo es una red neuronal que utiliza estadísticas como los puntos, diferencia de goles y Head-to-Head wins de los últimos 10 partidos. Este modelo predice el 'FTR' o resultado de un partido de la liga.

Descripción de la Arquitectura:

- Entrada:
  - Características seleccionadas: 'HomeTeam', 'AwayTeam', 'PH', 'PA', 'GDH', 'GDA', 'HHHW', 'HHAW'.
  - Codificación de etiquetas para 'HomeTeam' y 'AwayTeam'.
- Capas:
  - Capa SimpleRNN con 32 unidades y activación ReLU.
  - Capa de Dropout con 20% de desconexión.
  - Capa Densa con 16 unidades y activación ReLU.
  - Capa de Dropout con 20% de desconexión.
  - Capa Densa de salida con activación softmax para la clasificación.
- Compilación:
  - Función de pérdida: 'sparse\_categorical\_crossentropy'.
  - Optimizador: 'adam'.
  - Métricas: 'accuracy'.
- Entrenamiento:
  - Épocas: 25.
  - Tamaño de lote: 10.

## Modelo 2: Red Neuronal con Capas Densas y Promedios de Datos

El segundo modelo es una red neuronal simple que utiliza los promedios de estadísticas extras como; tiros a puerta, tiros al arco, faltas y tarjetas de los últimos 10 partidos. Este modelo predice el 'FTR' o resultado de un partido de la liga.

Descripción de la Arquitectura:

- Entrada:
  - Características seleccionadas: 'HomeTeam', 'AwayTeam', 'FTHG\_avg', 'FTAG\_avg', 'HTHG\_avg', 'HTAG\_avg', 'HS\_avg', 'AS\_avg', 'HST\_avg', 'AST\_avg', 'HF\_avg', 'AF\_avg', 'HC\_avg', 'AC\_avg', 'HY\_avg', 'AY\_avg', 'HR\_avg', 'AR\_avg'.
  - Codificación de etiquetas para 'HomeTeam' y 'AwayTeam'.
- Capas:
  - Capa Densa con 64 unidades y activación ReLU.
  - Capa de Dropout con 10% de desconexión.
  - Capa Densa con 32 unidades y activación ReLU.
  - Capa de Dropout con 10% de desconexión.
  - Capa Densa de salida con activación softmax para la clasificación.
- Compilación:
  - Función de pérdida: 'sparse\_categorical\_crossentropy'.
  - Optimizador: 'adam'.
  - Métricas: 'accuracy'.
- Entrenamiento:
  - Épocas: 25.
  - Tamaño de lote: 32.

## Modelo 3: Red Neuronal Recurrente con Capas LSTM

El tercer modelo es una red neuronal recurrente con LSTM, que utiliza únicamente los puntos de ambos equipos, de los últimos 10 partidos. Este modelo predice el 'FTR' o resultado de un partido de la liga.

Descripción de la Arquitectura:

- Entrada:
  - Características seleccionadas: 'HomeTeam', 'AwayTeam', 'HomePointsSum', 'AwayPointsSum'.
  - Codificación de etiquetas para 'HomeTeam' y 'AwayTeam'.
- Capas:
  - Capa LSTM con 64 unidades y activación ReLU.
  - Capa de Dropout con 20% de desconexión.
  - Capa Densa con 32 unidades y activación ReLU.

- Capa de Dropout con 20% de desconexión.
- Capa Densa de salida con activación softmax para la clasificación.
- **Compilación:**
  - Función de pérdida: 'sparse\_categorical\_crossentropy'.
  - Optimizador: 'adam'.
  - Métricas: 'accuracy'.
- **Entrenamiento:**
  - Épocas: 25.
  - Tamaño de lote: 32.

#### **Modelo 4: Red Neuronal Recurrente con Capa SimpleRNN**

El cuarto modelo, al igual que el primero, es una red neuronal recurrente, pero este omite la característica de los Head-to-Head wins. Este modelo predice el 'FTR' o resultado de un partido de la liga.

Descripción de la Arquitectura:

- **Entrada:**
  - Características seleccionadas: 'HomeTeam', 'AwayTeam', 'PH', 'PA', 'GDH', 'GDA'.
  - Codificación de etiquetas para 'HomeTeam' y 'AwayTeam'.
- **Capas:**
  - Capa SimpleRNN con 32 unidades y activación ReLU.
  - Capa de Dropout con 20% de desconexión.
  - Capa Densa con 16 unidades y activación ReLU.
  - Capa de Dropout con 20% de desconexión.
  - Capa Densa de salida con activación softmax para la clasificación.
- **Compilación:**
  - Función de pérdida: 'sparse\_categorical\_crossentropy'.
  - Optimizador: 'adam'.
  - Métricas: 'accuracy'.
- **Entrenamiento:**
  - Épocas: 25.
  - Tamaño de lote: 10.

#### **Modelo 5: Red Neuronal con Capas Densas para Predicción del FC Barcelona**

El quinto y último modelo es una red neuronal que utiliza los promedios de las estadísticas de los últimos 10 partidos para poder predecir 'Resultado', que es una nueva columna en donde 1 significa que el Barcelona gana y 0 que pierde o empatara.

Descripción de la Arquitectura:

- Entrada:
  - Características seleccionadas: 'HomeTeam', 'AwayTeam' y estadísticas promedio ('FTHG\_avg', 'FTAG\_avg', ...).
  - Codificación de etiquetas para 'HomeTeam' y 'AwayTeam'.
- Capas:
  - Capa Densa con 32 unidades y activación ReLU.
  - Capa Densa con 16 unidades y activación ReLU.
  - Capa Densa de salida con activación sigmoid para la clasificación binaria.
- Compilación:
  - Función de pérdida: 'binary\_crossentropy'.
  - Optimizador: 'adam'.
  - Métricas: 'accuracy'.
- Entrenamiento:
  - Épocas: 10.
  - Tamaño de lote: 32.

## Observaciones Adicionales

### Preprocesamiento de Datos:

- Codificación de etiquetas: Utilizamos LabelEncoder para transformar las variables categóricas en valores numéricos.
  - 'HomeTeam' y 'AwayTeam' (nombres de los equipos) valores numéricos (del 0 al 30).
  - 'FTR' (resultado final) en valores numéricos. {A: 0, D: 1, H: 2}
  - Para el modelo 5 no hubo codificación de 'FTR', ya que en este se agregó una columna extra, llamada 'Resultado', en donde {Gana: 1, Pierde/Empata: 0}
- Cálculo de Puntos y Diferencias de Goles: Creamos nuevas características que representan la suma de puntos y las diferencias de goles acumuladas por cada equipo en los últimos 10 partidos.
- Ventana Deslizante: Implementamos una ventana deslizante para calcular victorias head-to-head y otras estadísticas acumuladas.
- Rolling averages: Creamos nuevas características que representan los promedios de las estadísticas extras de los últimos 10 partidos por cada equipo.

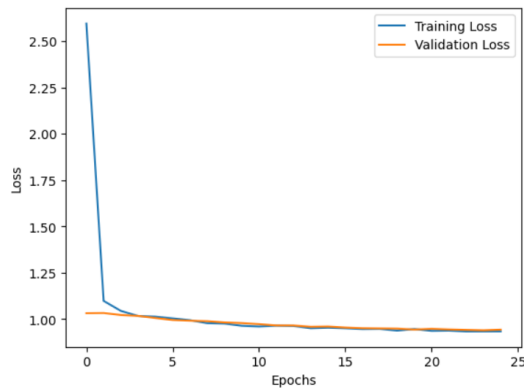
Esta metodología nos permitió abordar diferentes aspectos de la predicción de resultados en la Liga BBVA, desde el rendimiento general de los equipos hasta la predicción específica de partidos del FC Barcelona.

## Resultados

### Resultados de los modelos

Modelo 1: Red Neuronal Recurrente (RNN) con Capa SimpleRNN

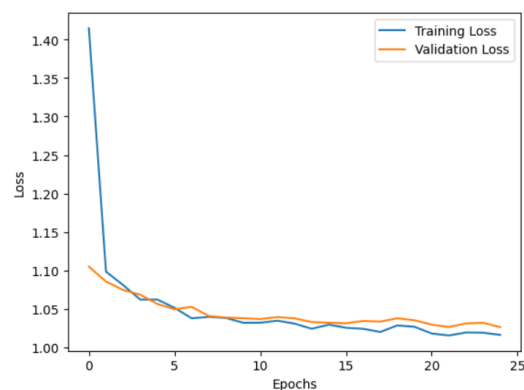
- Precisión: 56.28%
- Pérdida: 0.9340
- Pérdida de validación: 0.9437
- Precisión de validación: 0.5605



Prediccion	0	1	2
Real			
0	122	2	88
1	59	3	147
2	38	0	301

Modelo 2: Red Neuronal con Capas Densas y Promedios de Datos

- Precisión: 50.63%
- Perdida: 1.0164
- Pérdida de validación: 1.0264
- Precisión de validación: 0.5079

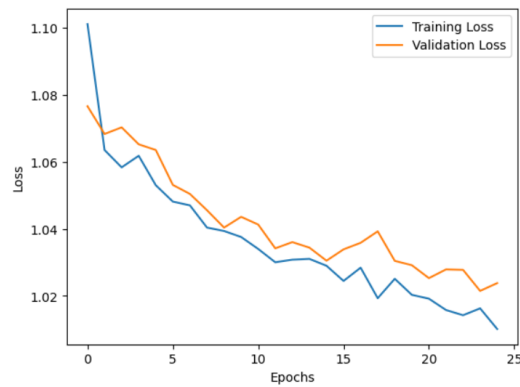


Prediccion	0	2
Real		
0	72	118
1	39	123
2	33	251

Modelo 3: Red Neuronal Recurrente con Capas LSTM

- Precisión: 50.90%
- Pérdida: 1.0101
- Pérdida de validación: 1.0238
- Precisión de validación: 0.4937

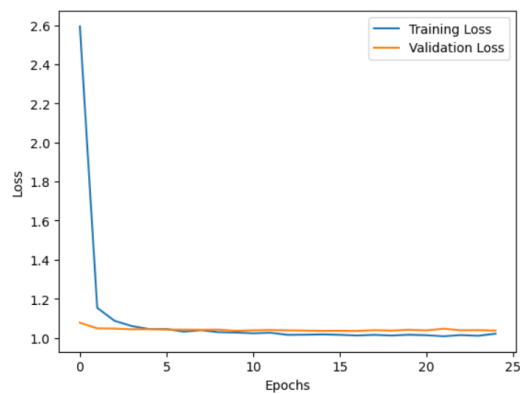




Prediccion	0	1	2
Real			
0	51	0	131
1	29	2	142
2	18	2	261

#### Modelo 4: Red Neuronal Recurrente con Capa SimpleRNN

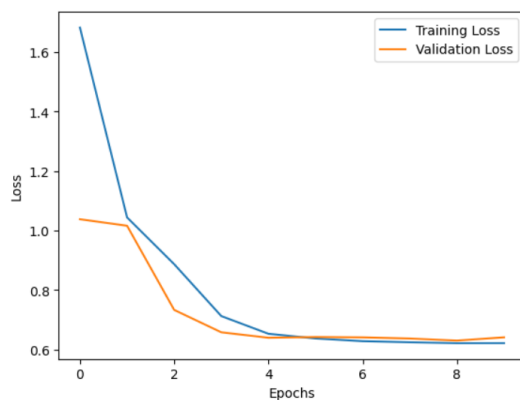
- Precisión: 49.90%
- Perdida: 1.0210
- Pérdida de validación: 1.0368
- Precisión de validación: 0.4908



Prediccion	0	2
Real		
0	58	154
1	19	190
2	24	315

#### Modelo 5: Red Neuronal con Capas Densas para Predicción del FC Barcelona

- Precisión: 69.26%
- Pérdida: 0.6221
- Pérdida de validación: 0.6415
- Precisión de validación: 0.6757



Prediccion	0	1
Real		
0	0	23
1	1	50

## Resumen de los resultados

*Tabla 2: Resumen de los resultados*


Modelo	Precisión	Pérdida
Modelo 1	56.28%	0.9340
Modelo 2	50.63%	1.0164
Modelo 3	50.90%	1.0101
Modelo 4	49.90%	1.0210
Modelo 5	69.26%	0.6221

## Predicciones de los modelos

Para poder probar nuestros modelos, decidimos ir un paso más adelante e intentar predecir la tabla de la Liga BBVA prediciendo las 13 jornadas que se han jugado hasta la fecha. Para poder realizar este experimento hicimos un par de cosas. Primero, descargamos el calendario de cada uno de los partidos que se han jugado esta temporada. Luego descargamos la tabla de resultados para cada una de las jornadas jugadas.

Con esta información, ya pudimos predecir los resultados con los modelos 1,3 y 4. Ya que lo que hicimos a continuación fue obtener el equipo local y visitante del calendario de la Liga y usamos esos datos como características del modelo. Adicionalmente, utilizamos la información de la jornada anterior para poder ver el rendimiento del equipo y en este paso, obtuvimos información como los puntos de cada uno de los equipos y la diferencia de goles. Luego, solo dejamos que nuestros modelos hicieran sus predicciones y comparamos las tablas de la última jornada, la jornada 13, para ver que tanto aceptaban nuestros modelos. A continuación podemos ver los primeros 5 equipos en 4 diferentes tablas. La primera tabla muestra la tabla real de la liga hasta la fecha y las siguientes 3 tablas, son las predicciones que hicieron nuestros modelos.

### Top 5, tabla real de la liga, jornada 13




1									
2	Pos	Equipo	Puntos	Juegos	Ganados	Empatados	Perdidos	Favor	Contra
3									
4	1	Girona	34	13	11	1	1	31	16
5									
6	2	Real Madrid	32	13	10	2	1	28	9
7									
8	3	Barcelona	30	13	9	3	1	26	13
9									
10	4	Ath Madrid	28	12	9	1	2	29	12
11									
12	5	Ath Bilbao	24	13	7	3	3	25	17

Top 5, predicciones modelo 1, jornada 13



1		
2	Equipo	Puntos
3		
4	Barcelona	39
5		
6	Ath Madrid	36
7		
8	Real Madrid	30
9		
10	Villarreal	27
11		
12	Sevilla	27
13		

Top 5, predicciones modelo 3, jornada 13



1		
2	Equipo	Puntos
3		
4	Almeria	27
5		
6	Barcelona	27
7		
8	Celta	24
9		
10	Betis	24
11		
12	Villarreal	21
13		

Top 5, predicciones modelo 4, jornada 13



1		
2	Equipo	Puntos
3		
4	Girona	36
5		
6	Real Madrid	30
7		
8	Ath Bilbao	30
9		
10	Barcelona	30
11		
12	Ath Madrid	27
13		

## Discusión

Para poder predecir los resultados de los partidos de la Liga BBVA, hicimos 5 modelos diferentes para poder comparar los resultados y ver cuál de los 5 era el más apropiado para utilizar y poder predecir, de manera precisa, los resultados de los partidos. Analizaremos entonces de manera detallada los resultados obtenidos para poder comprender la eficacia de cada modelo.

### Resultados obtenidos

Para evaluar la eficacia de nuestros modelos, observamos las métricas clave en términos de pérdida y precisión para el conjunto de entrenamiento y validación. A continuación, detallamos los resultados obtenidos por cada modelo:

Para el Modelo 1, basado en una Red Neuronal Recurrente (RNN) con Capa SimpleRNN, la estructura recurrente se diseñó para predecir el resultado de un partido dados ambos equipos. Este modelo alcanzó una precisión del 56.28% en el conjunto de entrenamiento y del 56.05% en el conjunto de validación. Estos resultados sugieren una capacidad razonable para anticipar el desempeño de los equipos en la liga a partir de la información histórica.

Para el Modelo 2, que utiliza una Red Neuronal con Capas Densas y Promedios de Datos, la arquitectura incorpora promedios móviles para suavizar las estadísticas. Aunque la precisión lograda fue del 50.63% en el conjunto de entrenamiento y del 50.79% en la validación, este modelo destaca la importancia de considerar promedios móviles para obtener una visión más equilibrada de las estadísticas a lo largo del tiempo.

Para el Modelo 3, basado en una Red Neuronal Recurrente (RNN) con Capa LSTM, la capa LSTM se introdujo para modelar patrones temporales en la suma de puntos. Aunque la precisión alcanzada fue del 50.90% en el conjunto de entrenamiento y del 49.37% en la validación, este modelo destaca la capacidad de las LSTM para capturar dependencias a largo plazo.

Para el Modelo 4, que utiliza una Red Neuronal Recurrente (RNN) con Puntos y Diferencia de Goles, la estructura se enfoca en la suma de puntos y la diferencia de goles, incorporando estadísticas acumuladas en ventanas deslizantes. Alcanzó una precisión del 49.90% en el conjunto de entrenamiento y del 49.08% en la validación.

Para el Modelo 5, basado en una Red Neuronal con Capas Densas para Predecir Victorias del Barcelona, la arquitectura predice si el Barcelona ganará un partido y utiliza promedios móviles de diversas estadísticas. Logró una precisión del 69.26% en el conjunto de entrenamiento y del 67.57% en la validación. Estos resultados indican una capacidad significativa para prever victorias específicas de equipos.

### **Significado de los resultados**

Podemos observar que la precisión de los modelos está dentro de un rango de un 49% a un 67%. Aunque ninguno de los modelos logró alcanzar niveles excepcionales de precisión, la diversidad de enfoques y la variabilidad en los resultados resaltan la complejidad de predecir resultados de fútbol. La modesta precisión general muestra lo difícil que es la tarea y sugiere que, si bien estos modelos proporcionan información valiosa, se necesitarán ajustes y consideraciones adicionales para mejorar su capacidad predictiva. Este análisis conjunto demuestra la importancia de explorar y ajustar constantemente enfoques para abordar la imprevisibilidad inherente en el mundo del fútbol.

### **¿Son los mejores resultados?**

Como mencionamos anteriormente, los resultados de nuestros modelos están alrededor del 60% de precisión. Y si bien, podríamos ajustar ciertas cosas de los modelos para que estos

mejoraran de alguna manera u otra, podemos mencionar que los resultados son de muy buena calidad.

Esto es debido a que en el fútbol hay demasiados factores que afectan el resultado de un partido, lo que hace que el deporte sea muy impredecible. Adicionalmente, los empates son un resultado muy común en el deporte, en la temporada 2021-2022 el 29% de los partidos de la liga fueron empates. Como se puede ver, en otros deportes, los empates no son muy comunes, por lo que solo se debe predecir el ganador, pero en este caso, los empates son un factor que hacen que la predicción de los resultados sea aún más complicada, ya que introducen una tercera posibilidad en la predicción.

Por ende, tomando en cuenta que la predicción de resultados de partidos de futbol es algo complejo, podemos ver que los mejores modelos creados en la vida real tienen al rededor del 60% de precisión (Rahman, A. 15 de octubre del 2020). Uno de los mejores modelos desarrollados fue uno utilizado para la copa del mundo del 2018, y este logro obtener una precisión del 63.3%.

Dados estos datos, podemos concluir que nuestros resultados no son perfectos, pero sí se acercan al 60%, el cual se considera un buen porcentaje de precisión para este deporte, por lo cual, nuestros resultados sí son de buena calidad.

### **Descubrimiento sobre los resultados**

A la hora de analizar dichos resultados, hemos llegado a tres grandes descubrimientos. El primero es algo que descubrimos a la hora de observar nuestras matrices de confusión de los modelos. Como podemos observar en dichas matrices, a la hora que los modelos que predicen resultados de partidos, predicen una victoria, el porcentaje de que esto esté correcto es más alto que el que esté incorrecto, pero a la hora de que predice un empate, el porcentaje es casi igual cuando acierta y cuando falla, por lo que podemos concluir que los modelos no son tan precisos a la hora de predecir empates. Adicionalmente, con el modelo 5, el que predice un resultado de partidos del Barcelona, cuando este predice una victoria, la probabilidad de que esté bien la predicción es alta, pero cuando predice una derrota, es muy baja, 50% o menos. Por lo tanto, podemos concluir que no podemos confiar mucho en las predicciones de nuestros modelos cuando predicen derrotas o empates, pero sí podemos confiar más a la hora de que predicen victorias.

Otro descubrimiento que vimos, es que a la hora de predecir partidos, como lo hicimos con los resultados de la temporada actual de la Liga BBVA, no siempre una mayor precisión significa mejores resultados. ¿Por qué? Esto lo podemos confirmar al observar las predicciones que realizamos con los modelos 1 y 4 (ver las tablas de jornada 13 en la sección de resultados). Como podemos observar, el modelo 1 tiene una precisión de un 56%, mientras que el 4 tiene una precisión de un 49%, aun así, al ver la imagen de la tabla real de la jornada 13, podemos ver que el modelo 4, logró predecir los mismos 5 líderes, con puntos muy

similares y casi el mismo orden, mientras que el modelo 1 solo logró predecir 3 de estos líderes con puntos alejados, y un orden no muy similar.

La razón subyacente a este fenómeno radica en que el Modelo 1 se centra demasiado en la historia de los equipos, prediciendo a menudo que los equipos con más head-to-head wins ganarán, lo cual es común en la liga con equipos como el Barcelona, el Real Madrid y el Atlético de Madrid. En contraste, el Modelo 4 se enfoca en el rendimiento actual, prescindiendo de las estadísticas head-to-head. Al proporcionar datos de la jornada anterior, este modelo logra predicciones más precisas al priorizar el desempeño actual de los equipos, revelando así la importancia de considerar la dinámica de la temporada en curso.

El tercer y último descubrimiento que vimos, fue en el modelo 5. Como mencionamos anteriormente, una de las cosas que hace que predecir el fútbol sea tan difícil, son los empates. Por lo tanto, en este modelo, que solo predecía si el Barcelona ganaría un partido o no, logramos remover esa tercera probabilidad del modelo, lo cual hace que el modelo pueda predecir de una manera más sencilla. Como podemos observar, la precisión de este modelo fue de un 67%, un 12% más que el mejor modelo de los que tenían las 3 posibilidades. Pero aun así, este modelo aún es más confiable a la hora de predecir una victoria que una derrota.

## Conclusiones

1. **Satisfactoria Precisión Dentro del Contexto Futbolístico:** Nuestros resultados, con precisiones en el rango del 40% al 60%, demuestran un desempeño considerable, especialmente considerando la complejidad del fútbol como un deporte impredecible. La tasa de éxito de los mejores modelos en esta área se encuentra típicamente alrededor del 60%, lo que resalta la dificultad intrínseca de predecir los resultados de los partidos.
2. **La Importancia de No Enfocarse Solo en la Historia:** Nuestra observación de que el Modelo 4, a pesar de tener una precisión inferior (49%) en comparación con el Modelo 1 (56%), logró predecir mejor la tabla de la liga en una temporada específica demuestra que una mayor precisión no siempre es sinónimo de mejores resultados. Al enfocarse en el rendimiento actual de los equipos y prescindir de las estadísticas head-to-head, el Modelo 4 logró resultados más precisos, destacando la necesidad de equilibrar el enfoque histórico con la realidad actual.
3. **Mayor Confianza en Predicciones de Victorias:** Nuestro análisis revela que podemos tener más confianza en las predicciones de victorias realizadas por nuestros modelos. Cuando se trata de prever victorias, la tasa de aciertos es significativamente más alta en comparación con empates y derrotas. Este hallazgo sugiere que nuestros modelos pueden proporcionar información más precisa y confiable en el escenario de victorias.

4. **Eliminación de Empates para Incrementar la Precisión:** La exclusión de empates en el Modelo 5 resultó en un aumento significativo en la precisión. Al centrarse específicamente en predecir si el Barcelona ganaría o no, el modelo logró alcanzar una precisión del 69.26% en el conjunto de entrenamiento y del 67.57% en el conjunto de validación. Esta estrategia destaca la importancia de adaptar el enfoque del modelo según los objetivos específicos de predicción y cómo la eliminación de ciertos resultados puede mejorar significativamente la capacidad predictiva.

## Apéndice

Ya que hemos desarrollado 5 diferentes modelos y varios scripts para poder modificar la data y hacer las predicciones, pensamos que sería mejor ver el código por separado, ya que es bastante. Para poder acceder al código fuente completo de este proyecto y explorar en detalle la implementación de los modelos, puede visitar nuestro repositorio en [GitHub](#). Además, para mayor comodidad, proporcionaremos un archivo ZIP separado que contiene todos los archivos necesarios y el código utilizado en este proyecto.

### Implementaciones de los modelos

#### *Implementación modelo 1*

```
1 model = Sequential()
2 model.add(SimpleRNN(32, input_shape=(X_train.shape[1], X_train.shape[2]), activation='relu'))
3 model.add(Dropout(0.2))
4 model.add(Dense(16, activation='relu'))
5 model.add(Dropout(0.2))
6 model.add(Dense(len(y.unique()), activation='softmax'))
```

#### *Implementación modelo 2*

```
1 model = Sequential()
2 model.add(Dense(64, activation='relu', input_shape=(X_train.shape[1],)))
3 model.add(Dropout(0.1))
4 model.add(Dense(32, activation='relu'))
5 model.add(Dropout(0.1))
6 model.add(Dense(3, activation='softmax'))
```

### *Implementación modelo 3*

```
1 model = Sequential()
2 model.add(LSTM(64, activation='relu', input_shape=(None, X_train.shape[1])))
3 model.add(Dropout(0.2))
4 model.add(Dense(32, activation='relu'))
5 model.add(Dropout(0.2))
6 model.add(Dense(3, activation='softmax'))
```

### *Implementación modelo 4*

```
1 model = Sequential()
2 model.add(SimpleRNN(32, input_shape=(X_train.shape[1], X_train.shape[2]), activation='relu'))
3 model.add(Dropout(0.2))
4 model.add(Dense(16, activation='relu'))
5 model.add(Dropout(0.2))
6 model.add(Dense(len(y.unique()), activation='softmax'))
```

### *Implementación modelo 5*

```
1 model = Sequential()
2 model.add(Dense(32, input_dim=len(X_train.columns), activation='relu'))
3 model.add(Dense(16, activation='relu'))
4 model.add(Dense(1, activation='sigmoid'))
```

## **Bibliografía**

Futbol.com. (s.f.). Resultados, primera división. Recuperado de:

<https://www.resultados-futbol.com/primer/grupo1/jornada14>

Football-Data.co.uk. (20 de noviembre del 2023). La Liga Season Results. Recuperado de:

<https://football-data.co.uk/spainm.php>



Pinnacle. (1 de agosto del 2018). Cómo realizar predicciones acertadas de fútbol. Recuperado de:

<https://www.pinnacle.com/es/betting-articles/soccer/how-to-make-accurate-soccer-predictions/xux2kmey9jl8m8r9>

Rahman, A. (15 de octubre del 2020). A deep learning framework for football match prediction. Recuperado de:

<https://link.springer.com/article/10.1007/s42452-019-1821-5>