Trabajo de Predicción

Adrián Sánchez-Miguel Ortega
Adrian.Sanchez14@alu.uclm.es

Julio Sánchez de las Heras Martín Consuegra
Julio.Sanchez6@alu.uclm.es

1. Introducción

En el presente trabajo se expone el diseño de diversos modelos de aprendizaje supervisado entrenados con el objetivo de pronosticar el resultado de cualquier encuentro no disputado durante el Campeonato Mundial de balonmano masculino de 2021. Para ello, se parte de las estadísticas generadas en dicho campeonato, las cuales están compuestas por 3071 registros correspondientes a la actuación de cada jugador en cada partido disputado. Además, esto se ve complementado con 31 características de diversa naturaleza: identificativas, defensivas y de tipos de lanzamiento. Cabe destacar que, se realizará un proceso de depuración y estandarización de los datos, eliminando las características identificativas junto con otras más, con el objetivo de realizar un análisis más profundo y preciso.

Por otra parte, para cumplir el objetivo descrito, se hará uso de diversos algoritmos de aprendizaje supervisado, tales como *Decision Tree*, *KNN*, *Random Forest o SVM*, entre otros, junto con una red neuronal, no solo para desarrollar un modelo que pronostique el resultado de los partidos, si no también, para realizar un estudio sobre las ventajas e inconvenientes de cada algoritmo en función del objetivo que se pretende cumplir.

Cabe destacar que, la metodología de trabajo que se seguirá para cada análisis posterior consistirá, en primer lugar, en una limpieza de los datos junto con una reducción de la dimensionalidad. Posteriormente, se normalizarán los datos en un intervalo de 0-1 gracias al *scaler MinMax*. Una vez se hayan procesado los datos, se entrenará cada modelo con los datos de entrenamiento, para, finalmente, evaluar el modelo con los datos de pruebas y realizar las predicciones deseadas.

2. Procesamiento de los datos

Tal y como se ha comentado en el apartado anterior, el primer paso a realizar en este trabajo es el procesamiento de los datos, con el objetivo de suministrar a los modelos información de alta calidad.

Para ello, inicialmente y gracias al asesoramiento de la opinión experta, se han eliminado características irrelevantes para el análisis, tales como todas las identificativas, aquellas que contengan información agregada y las tarjetas, entre otras. Además, gracias a la aplicación del algoritmo *Decision Tree*, se han eliminado características como las pérdidas y recuperaciones de balón, que no son relevantes para el correcto funcionamiento de los modelos. Gracias a este procedimiento se ha prevenido la *maldición de la dimensionalidad.*

En cuanto a cada registro del *Dataset* original, debido a que representa la actuación de cada jugador y el objetivo de este trabajo es predecir el resultado a nivel de equipo, se han agrupado dichos datos de jugadores por cada selección teniendo, así, el cómputo global del rendimiento de cada equipo.

Por último, gracias al *scaler MinMax* se han normalizado todos los datos en un rango de 0-1 para evitar que algunas características tengan más peso que otras en los modelos debido a su dimensionalidad.

Los datos de entrenamiento serán los resultados obtenidos en el mundial en aquellos partidos que no formen parte del conjunto de 24 sobre los cuales se tendrán que probar el modelo antes de realizar predicciones.

3. Definición conceptual de los modelos

Una vez conseguidos unos datos de calidad para nutrir a los algoritmos gracias a su procesamiento, se procede a definir los modelos empleados.

3.1. *Decision Tree*

Como primer modelo a desarrollar, se ha decidido optar por Decision Tree, debido a que es sencillo de construir y, si el árbol es pequeño, gracias a su buena interpretabilidad, podría servir como guía para el resto de modelos.

Para poder parametrizar de manera precisa este modelo se ha realizado un proceso de análisis en el cuál, se ha seleccionado como criterio de selección la entropía para cada una de las variables. Así, el

modelo elegirá aquella variable cuya proporción de casos positivos esté más alejada de 0,5.

Por otro lado, se han establecido otros parámetros relevantes para el modelo: no se ha limitado el número máximo de ramas que un nodo puede tener, se ha establecido a 1 el número mínimo de observaciones por nodo final (*leaf size*) y a 2 el número mínimo de observaciones para dividir un nodo (*split size*).

Además, para evitar el sobreaprendizaje, se decidió añadir una poda probando en un rango de 1-30 la profundidad máxima que se debe establecer para obtener el menor *Mean Absolute Error* (MAE), la cual, se ha establecido a 5. Así, el modelo solo considera los nodos más importantes, eliminando aquellos irrelevantes y redundantes.

3.2. KNN

Se ha decidido emplear KNN (*K-Nearest Neighbours*) debido a que es un modelo muy versátil, eficaz y robusto al ruido.

Para su parametrización, se ha seguido el mismo proceso que con el Decision Tree. Antes de aplicar el modelo de manera definitiva, se ha hecho uso de una validación cruzada de 5 cv para probar el parámetro de la distancia con Manhattan y Euclidea, además, para cada una de ellas se ha probado de manera secuencial un rango de 1-35 para el número de vecinos con el objetivo de intentar evitar un problema de sensibilidad al ruido y que las fronteras de decisión no sean dispersas. Esto ha resultado en que el menor MAE se obtiene con 3 vecinos y la distancia Manhattan.

3.3. Red neuronal

Para desarrollar una red neuronal que se adecúe al objetivo del presente trabajo, en primer lugar, se ha creado un modelo secuencial donde las capas son añadidas una tras otra.

Después, teniendo como base la opinión experta de la charla impartida por Pablo Pérez, se han añadido cuatro capas densamente conectadas con una función de activación *Rectified Linear Unit* (ReLu) con 128, 64, 32 y 16 neuronas, respectivamente y una última capa con una función de activación *softmax* para obtener una distribución de probabilidad sobre las clases. Cabe destacar que, entre cada una de las capas previamente mencionadas, se ha añadido una capa de *dropout* para evitar un posible sobreaprendizaje.

Por último, para compilar el modelo, se ha establecido una función de pérdida basada en la entropía cruzada categórica, adecuada para problemas de clasificación con más de dos clases, también se ha añadido el optimizador Adam y se emplea la precisión como métrica para evaluar los resultados del modelo.

Por otra parte, a la hora de entrenar el modelo se han establecido un total de 20 *epochs*, un tamaño de *batch* de 32 y un 20% para la división de validación.

3.4. SVM

Se ha decidido emplear SVM (*Support Vector Machine)* debido a que es un algoritmo muy versátil, siendo eficaz en una amplia gama de problemas. Además, es interesante para el problema de este trabajo, ya que SVM es efectivo incluso cuando el conjunto de datos es cuenta con pocos registros y muchas características.

En cuanto a su parametrización, teniendo un kernel de tipo lineal, se ha empleado una validación cruzada con una búsqueda en *Grid* (*GridSearch*) probando distintos valores de C para buscar un equilibrio entre los márgenes y que no haya *overfitting*.

3.5. *Random Forest*

Para evitar el sobreajuste producido por modelos más simples, tales como el *Decision Tree*, se ha decidido emplear conjuntos de modelos, en este caso *Random Forest*, el cual tiene una parametrización sencilla y no necesita poda. Así, se puede tener una gran cantidad de modelos *Decision Tree* independientes, los cuales son entrenados con un subespacio aleatorio y se coordinan para obtener el resultado final.

Por otra parte, para realizar una correcta parametrización del *Random Forest*, se ha empleado una búsqueda en *Grid* (*GridSearch*) analizando distintos valores para el número de estimadores, la máxima profundidad, el número mínimo de observaciones para un nodo hoja y para dividir un nodo de decisión y, por último, el criterio de selección.

4. Validación de los modelos

Una vez se han definido, parametrizado y entrenado los modelos establecidos, se han empleado los 24 primeros partidos llevados a cabo en el campeonato para evaluar dichos modelos, calculando el grado de acierto conseguido habiéndose medido con la precisión media del modelo y el *f1-score*. Los respectivos resultados se muestran a continuación:

- ***Decision Tree***: Se ha conseguido un 62,5% de acierto.
- **KNN**: Se ha conseguido un 79% de acierto.
- **Red neuronal**: Se ha conseguido un 83,3% de acierto.
- **SVM**: Se ha conseguido un 88% de acierto.

- ***Random Forest***: Se ha conseguido un 83% de acierto.

5. Pronóstico de resultados

Los resultados obtenidos una vez habiendo entrenado y validado los modelos se muestra en la siguiente tabla. Cabe destacar que, cuando se muestra un 1, simboliza la victoria del equipo local, una X, simboliza el empate entre los dos equipos y un 2 simboliza la victoria del equipo visitante.

Model	ESP Vs SWE	HUN Vs DEN	DEN Vs FRA	BRN Vs BLR	ISL Vs CRO
Decision Tree	1: 66% X: 0% 2: 33%	1: 0% X: 100% 2: 0%	1: 66% X: 0% 2: 33%	1: 0% X: 0% 2: 100%	1: 100% X: 0% 2: 0%
Red Neuronal	1: 40% X: 20% 2: 40%	1: 33% X: 25% 2: 42%	1: 46% X: 18% 2: 36%	1: 34% X: 26% 2: 40%	1: 50% X: 18% 2: 32%
SVM	1: 36% X: 16% 2: 48%	1: 8% X: 10% 2: 82%	1: 62% X: 16% 2: 22%	1: 18% X: 8% 2: 74%	1: 66% X: 14% 2: 20%
Random Forest	1: 88% X: 88% 2: 88%	1: 88% X: 88% 2: 88%	1: 88% X: 88% 2: 88%	1: 88% X: 88% 2: 88%	1: 88% X: 88% 2: 88%
KNN	1: 33% X: 0% 2: 66%	1: 0% X: 33% 2: 66%	1: 66% X: 0% 2: 33%	1: 0% X: 0% 2: 100%	1: 66% X: 33% 2: 0%

6. Conclusiones

La dimensionalidad, ya en la primera parte de este trabajo, fue un gran escollo que requirió de varios intentos y formas de encontrar los mejores resultados. A la hora de reducir dimensionalidad ha sido fundamental atender a las indicaciones de una opinión experta en la materia como la del profesor Eusebio Angulo, que nos indicó la poca importancia que tenían las tarjetas amarillas o azules, y otras indicaciones más específicas por posición que han ayudado a filtrar los datos.

Otra de las dificultades encontradas en la primera parte del trabajo fue ver que los datos que salían no concordaban con los resultados del torneo, hasta que se descubrió en la fuente de datos la existencia de la *President Cup*, al igual que la falta de un día entero de datos, el del 21 de enero de 2021, donde se disputaron 6 partidos, lo cual distorsionaba significativamente los datos.

En lo que respecta a este trabajo, en la validación de sus resultados, al no saber qué habría ocurrido en los partidos a predecir, se ha intentado recurrir al máximo número de algoritmos de aprendizaje supervisado estudiados en la materia, con el objetivo de comparar resultados y ver cuál arroja unos que se podrían asemejar más a la realidad.

Finalmente, resaltar la importancia de una correcta parametrización de cada uno de los algoritmos, lo cual resulta igual o más importante que el procesamiento previo de los datos a analizar, ya que ambas acciones ayudarán a crear un modelo más robusto.

Trabajo Final

Adrián Sánchez-Miguel Ortega
Adrian.Sanchez14@alu.uclm.es

Julio Sánchez de las Heras Martín Consuegra
Julio.Sanchez6@alu.uclm.es

1. Primera pregunta

Uno de los aspectos más destacables que se han aprendido de este trabajo es que, si bien hay técnicas más enfocadas a algún objetivo concreto, la clave para obtener unos buenos resultados a la hora de trabajar con problemas de aprendizaje automático no es tanto el algoritmo que se emplea, sino el procesamiento de los datos que se realiza previo a la ejecución del modelo. Esto es, pese a tener, teóricamente, el algoritmo enfocado a un objetivo concreto, si no se realiza un estudio y análisis del dominio del problema, no se cuenta con una opinión experta, no se evita la maldición de la dimensionalidad y no se normalizan los datos, se van a obtener unos resultados pésimos.

Dicho esto, si bien es cierto que no es un algoritmo para detectar *outliers*, como podría ser *DBSCAN*, el cual sería una opción poco apropiada para este objetivo, entre otras opciones como el clustering jerárquico o *k-means y el Expectation-Maximization* no hay tanta diferencia en los resultados obtenidos, llegando a realizar una comprobación empírica en nuestro código.

Pese a que es computacionalmente más costoso, al no ser esférica la distribución de los datos en los distintos gráficos creados, *k-means o c-means* serían menos efectivos que el *Expectation-Maximization*. Además, el *k-means* no es bueno con grupos de distintos tamaños como los que se forman al dividir en clusters en este caso. Con respecto al clustering jerárquico, será más apropiado si se quieren formar muchos clusters, pero en este problema, se ha observado que con un número reducido de, en torno a 4 o 5 clusters, puede ser más que suficiente.

2. Segunda pregunta

Pese a que se ha conseguido crear un modelo que a partir de las estadísticas recabadas a lo largo del torneo es capaz de realizar una predicción bastante creíble de lo que podría ocurrir en aquellos partidos que no se habían disputado, es cierto que se podría haber entrenado un modelo mejor para la predicción. El principal escollo que superar es el reducido número de partidos que se le proporcionan al modelo para entrenar, debido a la necesidad de ajustarse a los encuentros que tuvieron lugar en el mundial de balonmano. Normalmente, los modelos suelen tener bastantes más datos de partida, llegando, en algunos casos a miles o cientos de miles, en contraposición con los menos de 70 partidos con los que se han contado para entrenar el modelo.

Por otra parte, el modelo se entrenó utilizando datos globales de todo el torneo para cada selección, sin pormenorizar en qué pasó en cada uno de los partidos y teniendo en cuenta que se pierde información de esta manera, al faltar partidos en el *dataset* original, ya que no se podría predecir luego un encuentro al no tener sus respectivos datos. Un modelo que utilizara datos de cada partido podría dar mejores resultados.

Por último, los parámetros que se tienen en cuenta en cada uno de los registros del *dataset* original carecen de un aspecto fundamental en el balonmano, como es la defensa. Esto nos ha llevado a tener que cribar los porteros, al no haber registros de sus paradas, o a que jugadores de campo de talla mundial que brillan por sus labores defensivas pasen un poco más desapercibidos en los resultados obtenidos al carecer de estadísticas en las que destaquen más allá de los bloqueos o recuperaciones.

Como otras técnicas no vistas en clase a aplicar en este problema, se listan las siguientes:

- **Validación cruzada:** La validación cruzada es una técnica utilizada para evaluar el rendimiento de un modelo de aprendizaje supervisado. En lugar de evaluar el modelo en un solo conjunto de datos, se divide el conjunto de datos en varias partes y se realiza el entrenamiento y evaluación en múltiples combinaciones. Esto ayuda a obtener una estimación más confiable del rendimiento del modelo y reducir la posibilidad de sobreajuste.
- **Optimización de hiperparámetros:** Los modelos de aprendizaje supervisado suelen tener hiperparámetros, que son configuraciones ajustables que no se aprenden directamente del conjunto de datos. La optimización de hiperpa-

rámetros implica buscar la combinación óptima de valores de hiperparámetros que maximicen el rendimiento del modelo. Esto se puede lograr mediante técnicas como la búsqueda en cuadrícula, la búsqueda aleatoria o la optimización bayesiana.

- **SMOTE (*Synthetic Minority Over-sampling Technique*):** Un inconveniente destacable que se ha tenido durante el desarrollo de este trabajo, no solo ha sido el reducido tamaño del conjunto de datos, sino que, además, estos estaban desbalanceados. Esto es, del conjunto de encuentros disputados con los que se contaban, había muy pocos empates y el número de victorias del equipo local superaba a las victorias del equipo visitante, lo cual ha afectado notablemente las predicciones de los modelos desarrollados. Debido a esto, se ha considerado necesaria la utilización de SMOTE, una técnica que soluciona el desequilibrio de clases mediante la generación de datos sintéticos de las clases minoritaritas, esto con el objetivo de equilibrar el conjunto de datos.

3. Tercera pregunta

Aparte de los modelos utilizados, se añaden algunas más para el modelo de aprendizaje no-supervisado visto en la primera entrega de este trabajo:

- **Propagación de Afinidad (*Affinity Propagation*):** Con este algoritmo se determina automáticamente el número de clusters y se asigna puntos "ejemplares", los cuales representan a los clusters y son como centroides para el *k-means*, o "responsables" que se asignan a los distintos ejemplares dependiendo de la similitud entre ellos. Sabiendo que el resto de los algoritmos son capaces de detectar diferencias por posición, este algoritmo destaca cuando la similitud entre los puntos es significativa a la hora de asignarlos a ejemplares. Esto es bueno ya que da menor lugar a error humano a la hora de declarar el número de clusters a utilizar, al poder detectarlos automáticamente dependiendo del parecido entre los puntos.
- **Desplazamiento de Media (*Mean Shift*):** Este algoritmo también detecta el número de clusters de manera automática, encontrando modas o máximos locales en los datos. En este algoritmo se mueven los datos hacia regiones de puntos más densas hasta que se cumple un criterio de convergencia. Una vez movidos, se asignan etiquetas a los clusters según su posición en el espacio, aún con formas y tamaños arbitrarios. Esto podría resultar útil en este trabajo, ya que el algoritmo podría adaptarse bien a las formas arbitrarias en las que se presentan los datos en nuestro modelo de datos.
- **BIRCH (Reducción Iterativa Equilibrada y Agrupación Utilizando Jerarquías):** Este modelo es similar al clustering jerárquico, pese a ser más apropiado para cantidades de datos más grandes que las presentadas en este trabajo, podría ser otra opción válida. Clasifica los datos constantemente conforme se van agregando, ajustado el árbol de agrupamiento. Una vez formado, se pueden buscar los grupos jerárquicos y analizarlos. Como en el anterior, podría ser útil al poder detectar grupos con tamaños y formas arbitrarias.
- ***Ensemble learning*:** El *ensemble learning* implica combinar las predicciones de múltiples modelos individuales para obtener una predicción final más precisa. Esto se puede lograr mediante técnicas como el promedio de predicciones, la votación o el *stacking*. Los enfoques de *ensemble* pueden mejorar la generalización del modelo y reducir el riesgo de sobreajuste. Pese a que en este trabajo se ha utilizado una técnica de *ensemble learning*, el *Random Forest*, también se podrían emplear otras técnicas complejas como el *Boosting*.