UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

UN ENFOQUE DE MACHINE LEARNING PARA PREDECIR EQUIPO GANADOR EN JUEGOS DE LA LIGA PROFESIONAL DE BALONCESTO NBA

POR:

ING. JESÚS IGNACIO RODRÍGUEZ SIBAJA

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERÍA EN COMPUTACIÓN

CHIHUAHUA, CHIH., MÉXICO **12/2022**

Un enfoque de machine learning para predecir qué equipo gana en un encuentro de la NBA. Tesis presentada por Jesús Ignacio Rodríguez Sibaja como requisito parcial para obtener el grado de Maestro en Ingeniería en Computación, ha sido aprobada y aceptada por:

M.I. Fabián Vinicio Hernández Martínez
Director de la Facultad de Ingeniería

Dr. Fernando Martínez Reyes
Secretario de Investigación y Posgrado

M.S.I. Karina Rocío Requena Yáñez
Coordinador(a) Académico

Dr. Luis Carlos González Gurrola
Director(a) de Tesis

Fecha
24 de Enero del 2023

Comité:
DR. LUIS CARLOS GONZÁLEZ GURROLA
DR. JESUS ROBERTO LÓPEZ SANTILLÁN
DR. FERNANDO MARTÍNEZ REYES
DR. RAYMUNDO CORNEJO GARCÍA

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

27 de febrero de 2023.

ING. JESÚS IGNACIO RODRÍGUEZ SIBAJA
Presente.

En atención a su solicitud relativa al desarrollo del trabajo de tesis para obtener el grado de Maestro en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Luis Carlos González Gurrola** para que lo desarrolle con el título **"UN ENFOQUE DE MACHINE LEARNING PARA PREDECIR EQUIPO GANADOR EN JUEGOS DE LA LIGA PROFESIONAL DE BALONCESTO NBA"**.

Índice de Contenido

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

Capítulo 3: Metodología

3.1 Extracción de los datos

3.2 Procesamiento de los datos

3.3 Modelos

3.4 Predicciones en base a datos históricos

3.5 Predicciones en tiempo real

3.6 Importancia de las características

Capítulo 4: Resultados

4.1 Resultados históricos

4.2 Resultados en tiempo real

4.3 Shap values

Capítulo 5: Conclusiones

5.1 Discusiones

5.2 Trabajos futuros

Bibliografía

Apéndice

Currículum Vitae

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares del documento terminal.

A T E N T A M E N T E

"Naturam subiecit aliis"

EL DIRECTOR

M.I. FABIÁN VINICIO HERNÁNDEZ MARTÍNEZ

FACULTAD DE INGENIERÍA U.A.CH.

DIRECCIÓN

SECRETARIO DE INVESTIGACIÓN Y POSGRADO

DR. FERNANDO MARTÍNEZ REYES

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

Dedicatoria

Dedico este proyecto, con gratitud y alegría, a mis seres queridos.

A mi madre, Rosa, quién siempre ha sabido guiarme y apoyarme en todos mis planes.

A Ignacio, mi padre, por ayudarme a solucionar mis dudas con su conocimiento.

A mi hermana, Karelia, con quien siempre me ha servido como referencia

A mi familia en general, quienes a la distancia me han apoyado de inicio a fin.

A mi amada novia, Cassandra, quien con todos sus recursos ha estado apoyándome tanto en lo material como lo emocional.

A Pancake, mi mascota, quién se quedaba a mi lado todas las noches en vela.

Agradecimientos

Quiero dedicar este espacio para agradecer a todas y cada una de las personas que, a lo largo de este proyecto, han apoyado para volverlo posible.

Primero a mis padres, Ignacio Rodríguez y Rosa Sibaja, quienes han dedicado toda su vida en educarme, amarme y apoyarme. Quienes me han demostrado con actos que todo es posible y que han sacrificado de todo por mí, para forma y criar a un hombre de bien. Con ellos estoy y estaré infinitamente agradecido.

También me gustaría agradecer a mi hermana, quién ha sido, desde su poca experiencia en la vida, una guía y referencia de los pasos a seguir y quien, al igual que mis padres, siempre me han apoyado a la distancia.

A mi amada novia, Cassandra, quién ha sido un pilar en mi vida desde el inicio hasta al final de este proyecto, siempre sabiendo que hacer y que decir cuando ya no creía poder más, quién siempre creyó en mi y me incitó a superarme y quién, sin importar qué, estuvo para mí.

A mi asesor de tesis Luis Carlos Gonzalez, quién me brindo su apoyo, conocimiento y paciencia a lo largo del desarrollo de este proyecto, a su vez por confiar en mi talento y habilidades.

A todos mis maestros a lo largo de mi formación, por brindar siempre su apoyo y conocimiento para alcanzar todas mis metas.

Y por último y no menos importante, a mi mascota Pancake, quién llegó a mi vida un día y se volvió un gran apoyo emocional.

Con estas palabras solo logró demostrar una fracción de lo agradecido que estoy con todas y cada una de las personas mencionadas anteriormente, las palabras no alcanzan para demostrar el agradecimiento verdadero.

¡Gracias!

Resumen

Gracias por el crecimiento acelerado del *big data* deportivo en los últimos años y con técnicas de minería de datos y *machine learning* efectuadas con éxito en diversas áreas de estudio, surge la inspiración en la pregunta ¿Será posible predecir con gran precisión los resultados de eventos deportivos?

El área de investigación de eventos deportivos es muy atractiva y estudiada y la simple tarea de predecir los resultados en los deportes es desafiante por sí misma

En este proyecto, se aborda el uso de modelos de *machine learning* combinado con técnicas de minería de datos, para lograr un procesamiento de datos de la NBA de forma innovadora, e intentar alcanzar un porcentaje de predicción competitivo a comparación de investigaciones previas.

Analizamos los resultados obtenidos por cada uno de los clasificadores involucrados, KNN, Regresión Logistica, Support Vector Machine, XGBoost, etc. Haciendo experimentos con datos históricos, experimentos en tiempo real y analizando las características que más aportan a nuestros resultados.

También, en una búsqueda del entendimiento de las predicciones realizadas por la computadora y la importancia dadas por las mismas a cada una de las características, se realiza un análisis de valores shapley para dos fechas en puntos estratégicos de una temporada ya finalizada.

Después de la realización de los experimentos se logró obtener hasta un 75% de precisión, encontrando así que los resultados de este proyecto han sido bastante alentadores y pudiendo ser referenciado en trabajos futuros.

Palabras clave: Machine learning, data mining, outcomes, prediction, data processing, data management.

Índice de Tablas

Índice de Figuras

CAPÍTULO 1: INTRODUCCIÓN

1.1 Introducción al proyecto

Los seres humanos sentimos curiosidad por conocer el futuro, debido a que tener el control de lo que se avecina nos genera seguridad. Esto se ve reflejado en nuestra vida cotidiana al jugar con nuestros amigos a la piedra papel o tijeras y en los deportes, en donde siempre queremos saber quién será el equipo ganador.

La industria del deporte es una de las que más ganancias aportan en la economía de los países. Tan solo en México, para 2019, la industria deportiva generó 114 mil millones de pesos anuales [1], siendo esta cifra la combinación de todas las ligas nacionales. A pesar de ser México un país que es bastante apasionado en cuanto a deportes se refiere, no se encuentra ni de cerca a producir las ganancias vistas en las ligas de otros países como pueden ser la Premier League (5.300 millones de dólares al año), La liga de futbol española (2.200 millones de dólares al año) la NBA (4.800 millones de dólares al año) la MLB (9.500 millones de dólares al año) y la NFL (13.000 millones de dólares al año) [2].

Con el rápido crecimiento de las tecnologías de la información y los deportes, el análisis de la información deportiva se ha convertido en un tema cada vez más desafiante. El big data deportivo que proviene del internet muestra una tendencia de rápido crecimiento.

La gran cantidad de datos relacionados con los eventos deportivos que se producen hoy en día, ha generado un incremento en el interés por los deportes en el público general [3].

La implementación de nuevas tecnologías en la industria deportiva no es algo extraño, sí lo puede llegar a ser en el sector de las predicciones en donde resalta la incógnita: ¿Será posible generar predicciones más acertadas de una forma segura?

1.2 Antecedentes

1.2.1 Antecedentes de investigación en el área deportiva

Debido a la naturaleza deportiva de la investigación propuesta en este documento, podemos encontrar diversas investigaciones que abarcan dicha problemática en diversas ligas/deportes como la investigación presentada por Li, Hang [4] en donde abordaba la problemática en la UEFA, donde haciendo uso de redes neuronales alimentadas por un vector con 20 características, alcanzando un 66.5% de precisión en promedio.

Otra investigación destacable en esta área (pero con diferente enfoque) es la realizada por McCullagh [5], en 2009, donde combina técnicas de Data Mining con redes neuronales para hacer un análisis, en la Australian Football League (AFL), enfocado en determinar a los jugadores que serán seleccionados anualmente.

1.2.2 Antecedentes enfocados en la NBA

Si nos enfocamos en el análisis de la National Basketball Association (NBA), la liga analizada en esta investigación, encontramos investigaciones destacables con distintos enfoques como los siguientes.

E. Zdravevski & A. Kulakov [6] en 2009 desarrollaron varios modelos para intentar predecir los juegos de la NBA, dichos modelos fueron alimentados con vectores de 10 características de las cuales 2 eran características externas a las estadísticas de los partidos. Dichos modelos fueron puestos a prueba con 930. EL mejor resultado obtenido fue de 72.8% de precisión y 51.8% el peor resultado.

La investigación de Cao [7] obtuvo un promedio de precisión del 69.67%, estudiando 6 temporadas (5 de entrenamiento y 1 de prueba). La precisión fue obtenida alimentando un modelo con 36 características.

Por otro lado, con la investigación realizada por Loeffelholz, Bednar y Bauer [8] (2009) en la cual procesaron 22 características para alimentar una Red Neuronal Prealimentada (FFNN) se obtuvo una precisión en promedio de 71.67% pero, haciendo combinaciones menos densas de dichas características (utilizando solamente 4) lograron alcanzar hasta un 74.33% de precisión con datos históricos.

Ref	Autor	Año	Temporadas analizadas	Variables externas	No. variables	Mejor desempeño
[6]	E. Zdravevski & A. Kulakov	2009	1	Si	10	72.80%
[7]	C. Cao	2012	6	No	36	69.67
[8]	Loeffelholz, Bednar & Bauer	2009	1	No	4	71.67%
[9]	R. A. Torres	2013	7	No	8	68.44%
[10]	J. Guzman	2016	1	Si	218	80.22%
[11]	J. Lin, L. Short & V. Sundaresan	2014	7	No	17	65.15

Tabla 1.- Comparativa de antecedentes enfocados en NBA

1.3 Enfoque

Esta investigación se enfoca en abordar la incógnita planteada anteriormente mediante la inteligencia artificial, usando técnicas de Minería de datos y Machine Learning , realizando e implementando modelos de predicción a un vector de características con las variables (puntos, win rate, jugadas, etc.) que más influyan en el resultado de los partidos, intentando así obtener mejores resultados a partir de la innovando en la forma de procesar el vector de características. Estas variables, dependerán de la liga a la que estemos viendo, pudiendo ser canastas, rebotes, bloqueos, etc., para la NBA por lo que será necesario ir determinando las combinaciones de características que mejor se adapten a nuestro objetivo.

A su vez, y debido a que, en un escenario más cercano a la realidad, se espera poder profundizar no solo en las predicciones con datos históricos, sino también con datos en tiempo real, pudiendo analizar así un caso de uso más apegado a la naturaleza de este problema.

1.4 Justificación

Este proyecto está cimentando en la necesidad de hacer uso de los datos generados por el basquetbol, los cuales han visto incrementados debido al rápido crecimiento de las tecnologías de la información en los deportes, ascendiendo a cifras de cientos de millones de datos generados al día (Bai,2021). Logrando transformar los datos en información que ayude a la toma de decisiones.

Este proyecto pretende ser una mejora notable en las predicciones deportivas, en base a un tratamiento distinto de los datos, lo cual deriva en una mejor toma de decisiones tanto por parte de los aficionados, al poder saber quién ganará, como de los entrenadores, que podrán tomar decisiones en base a los resultados del modelo.

1.5 Hipótesis.

- El modelo desarrollado tendrá una precisión de al menos 70% en sus predicciones.
- El desempeño del equipo mejora si juega en casa.
- Los partidos “milagro” no son predecibles.

1.6 Preguntas de investigación

- ¿Podremos mejorar la precisión de las predicciones con variables externas?
- ¿Es determinante el status del equipo? Es decir, si juega como local o visitante
- ¿El agregado de variables externas resultaran en una mejor precisión al predecir?
- ¿Se podrá hacer un modelo mejor que el de una casa de apuestas?

1.7 Metas

1.7.1 Objetivo general

Alcanzar una predicción competitiva en juegos de la NBA mediante un enfoque original en el tratamiento de los datos y algoritmos de *Machine Learning*.

1.7.2 Objetivos específicos

- Hacer uso de técnicas de minería de datos para extraer estadísticas de las temporadas.
- Analizar características para determinar si puede contribuir a mejorar el vector de características.
- Lograr automatizar todo el proceso de extracción y predicción mediante uso de Python.
- Determinar puntos de mejora mediante análisis de errores en las predicciones del modelo.

1.8 Limitaciones

Debido a la naturaleza competitiva del baloncesto, este deporte combina muchos factores de incertidumbre, los cuales son imposibles de contemplar como las lesiones, los traspasos de jugadores a otros equipos, situaciones personales, etc. Que pueden afectar el resultado del encuentro. Por lo que la precisión de la investigación no anhela a la perfección.

A su vez, las temporadas se segmentan en 3, la pre-temporada, la temporada regular y los playoffs. Las pre-temporadas no serán contempladas porque suele ser más un segmento de tiempo en la que los entrenadores prueban a sus nuevos jugadores o no suelen contar con el equipo completo.

Los playoffs, cuentan con una dinámica distinta, en donde se disputa el mejor de 5 encuentros, por lo cual los equipos suelen jugar de distintas maneras y con distintas estrategias, por lo cual construir un modelo para los playoffs no es adecuado.

Basándonos en lo anterior, esta investigación se enfocará solamente en predecir el resultado de los juegos de la temporada regular.

CAPÍTULO 2: MARCO TEÓRICO

Este apartado aborda conceptos y herramientas que son necesarias conocer para el completo entendimiento de esta investigación.

2.1 Machine Learning

Machine learning es una forma de la IA que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. [12] Conforme el algoritmo ingiere datos de entrenamiento, es posible producir modelos más precisos basados en datos. Un modelo de machine learning es la salida de información que se genera cuando entrena su algoritmo de machine learning con datos. Después del entrenamiento, al proporcionar un modelo con una entrada, se le dará una salida. Por ejemplo, un algoritmo predictivo creará un modelo predictivo. A continuación, cuando proporcione el modelo predictivo con datos, recibirá un pronóstico basado en los datos que entrenaron al modelo [12].

Las técnicas de machine learning son necesarias para mejorar la precisión de los modelos predictivos. Dependiendo de la naturaleza del problema que se está atendiendo, existen diferentes enfoques basados en el tipo y volumen de los datos [12].

2.1.1 Aprendizaje supervisado

El aprendizaje supervisado comienza típicamente con un conjunto establecido de datos y una cierta comprensión de cómo se clasifican estos datos. El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso de analítica. Estos datos tienen características etiquetadas que definen el significado de los datos. Por ejemplo, se puede crear una aplicación de machine learning con base en imágenes y descripciones escritas que distinga entre millones de animales [12].

2.1.2 Aprendizaje no supervisado

El aprendizaje no supervisado se utiliza cuando el problema requiere una cantidad masiva de datos sin etiquetar. Por ejemplo, las aplicaciones de redes sociales, tales como Twitter, Instagram y Snapchat, tienen grandes cantidades de datos sin etiquetar. La comprensión del significado detrás de estos datos requiere algoritmos que clasifican los datos con base en los patrones o clústeres que encuentra. El aprendizaje no supervisado lleva a cabo un proceso iterativo, analizando los datos sin intervención humana. Se utiliza con la tecnología de detección de spam en e-mails. Existen demasiadas variables en los e-mails legítimos y de spam para que un analista etiquete una cantidad masiva de e-mail no solicitado. En su lugar, los clasificadores de machine learning, basados en clustering y asociación, se aplican para identificar e-mail no deseado [12].

2.2 Regresión Logística

La regresión logística es un modelo estadístico que permite estudiar si una variable binomial depende, o no, de una u otras variables no necesariamente binomiales. [13] Se le denomina binomial a una variable que solo tiene dos posibles valores: verdadero o falso.

El modelo asume que la variable dependiente (Y) es binaria, para lo cual tomará valores de: Y=1 a la "presencia" del evento y Y=0 a la "ausencia" del evento [14], utiliza la siguiente función logística:

$$h(\mathrm{z}) = \frac{1}{1 + e^{-z}} \; Donde$$

$$Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Donde $X_1 X_2 \ldots X_K$ son covariables y la probabilidad de pertenecer a una población se calcula utilizando las siguientes funciones:

$$P_i = \frac{1}{1 + e^{-Z}}$$

$$1 - P_i = \frac{1}{1 + e^{Z}}$$

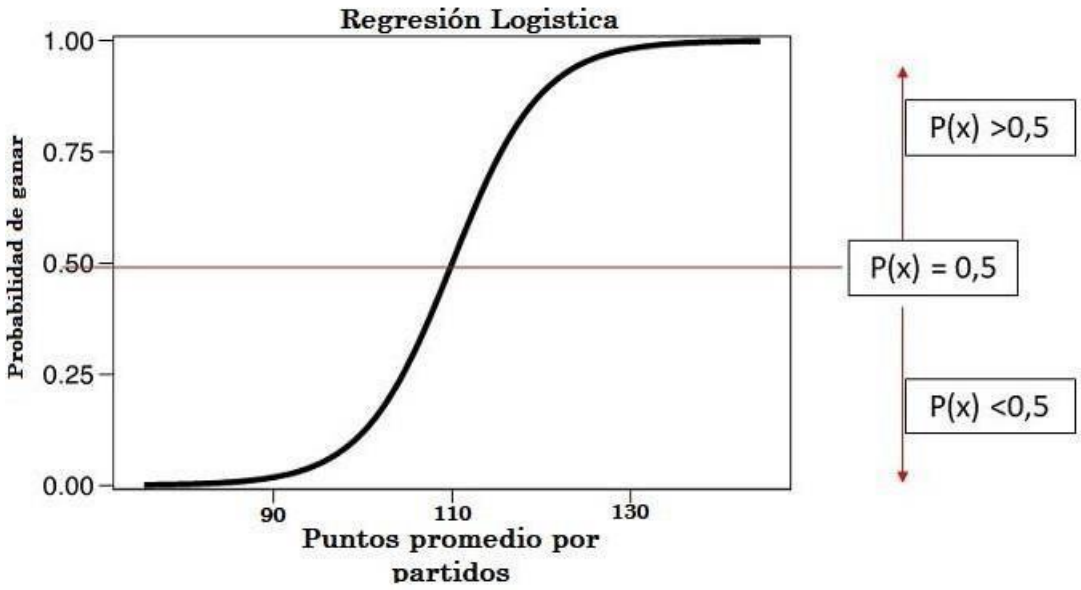


Figura 1.- Diagrama de regresión logística

2.2.1 Toma de decisiones a partir de regresión

El análisis de regresión proporciona una forma objetiva y sistemática de analizar datos. Como resultado, es menos probable que las decisiones basadas en la regresión estén sujetas a sesgos, son consistentes, la base de las decisiones puede explicarse completamente y, en general, son útiles. [15] Las ganancias están especialmente bien documentadas cuando se comparan con decisiones de juicio basadas en los mismos datos. [16] La regresión logística resulta útil para los casos en los que se desea predecir el valor de la variable binomial según los valores de un conjunto de predictores, siendo esto último el objetivo de esta investigación.

2.3 Boosting

El boosting es un método utilizado en el machine learning para reducir los errores en el análisis predictivo de datos. [17] Un único modelo de machine learning puede cometer errores de predicción según la precisión del conjunto de datos de entrenamiento. Por ejemplo, si un modelo de identificación de gatos se entrenó solo con imágenes de gatos blancos, es posible que en ocasiones no identifique un gato de color negro. El boosting intenta solucionar este problema mediante el entrenamiento secuencial de varios modelos para mejorar la precisión del sistema en general. [17]

Para comprender cómo funciona el boosting, describiremos cómo toman decisiones los modelos de machine learning. Si bien existen muchas variaciones en la implementación, los científicos de datos muchas veces utilizan algoritmos de árbol de decisión: [17]

2.3.1 Árboles de decisión

Los árboles de decisión son estructuras de datos en el machine learning que dividen el conjunto de datos en subconjuntos cada vez más pequeños en función de sus características. La idea es que los árboles de decisión dividan los datos reiteradamente hasta que solo quede una clase. Por ejemplo, es posible que el árbol haga una serie de preguntas de sí o no y divida los datos en categorías en cada paso. [17]

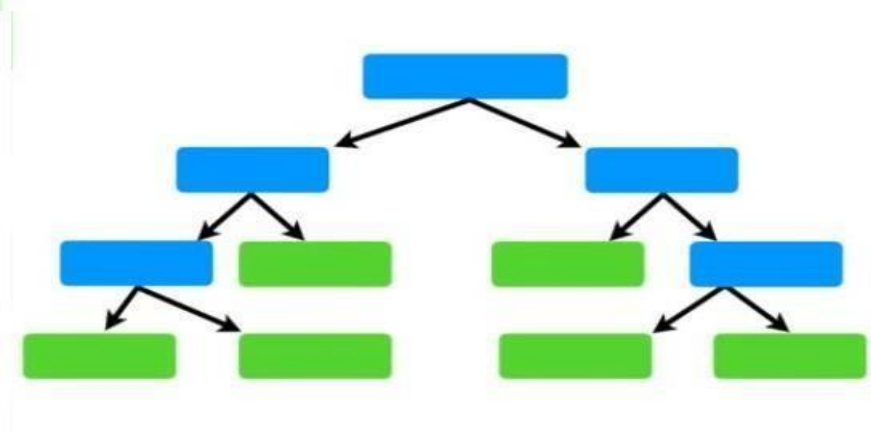


Figura 2.- Diagrama de árbol de decisión

Los árboles de decisión tienen la tendencia de sobre ajustar (*overfit*) [18]. Esto quiere decir que tienden a aprender muy bien los datos de entrenamiento, pero su generalización no es tan buena. Para mejorar mucho más la capacidad de generalización de los árboles de decisión, deberemos combinar varios árboles.

2.3.1.1 Random Forest

Un Random Forest es un conjunto de árboles de decisión combinados con bagging. Al usar bagging, lo que en realidad está pasando, es que distintos árboles ven distintas porciones de los datos. Ningún árbol ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y tenemos una predicción que generaliza mejor.

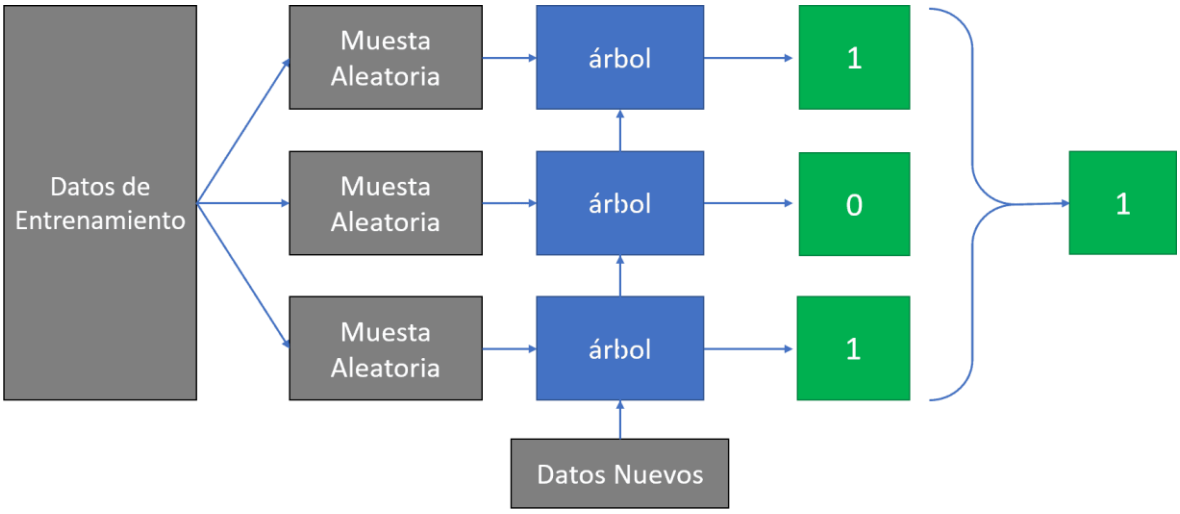


Figura 3.- Diagrama Random Forest

2.3.2 Método de conjunto boosting

El boosting crea un modelo de conjunto mediante la combinación secuencial de varios árboles de decisión débiles. Asigna ponderaciones a las salidas de los árboles individuales. Luego, a las clasificaciones incorrectas del primer árbol de decisión le da una ponderación más alta y una entrada al árbol siguiente. Después de numerosos ciclos, el método boosting combina estas reglas débiles en una única regla de predicción poderosa. [17]

2.3.3 Adaptative Boosting

Se adapta e intenta autocorregirse en cada iteración del proceso de boosting. Al principio AdaBoost brinda la misma ponderación a cada conjunto de datos. Luego, ajusta de forma automática las ponderaciones de los puntos de datos después de cada árbol de decisión. Otorga más ponderación a los elementos clasificados de forma incorrecta para corregirlos en la siguiente ronda. Repite el proceso hasta que el error residual, o la diferencia entre los valores reales y los previstos, cae por debajo de un límite aceptable. [17]

AdaBoost se puede utilizar con muchos predictores y, por lo general, no es tan sensible como otros algoritmos de boosting. Este enfoque no funciona bien cuando existe una correlación entre las características o una dimensionalidad alta de los datos. En general, AdaBoost es un tipo de boosting adecuado para problemas de clasificación. [17]

2.3.4 Gradient boosting

Gradient Boosting (GB) es similar a AdaBoost, ya que también es una técnica de entrenamiento secuencial. La diferencia entre el AdaBoost y el GB es que el GB no asigna más ponderación a los elementos que se clasificaron de forma incorrecta. En cambio, el software del GB optimiza la función de pérdida mediante la generación secuencial de estudiantes base, de manera que el estudiante base presente sea siempre más efectivo que el anterior. Este método intenta generar resultados precisos desde el principio en lugar de corregir errores a lo largo del proceso, como

AdaBoost. Por este motivo, el software de GB puede conducir a resultados más precisos. Gradient Boosting puede ayudar con problemas basados en la clasificación y en la regresión. [17]

2.4 Modelos Boosting

2.4.1 XGBoost Classifier

XGBoost es una librería de Gradient Boosting distribuida, optimizada y diseñada para ser altamente eficiente, flexible y portátil. Implementa algoritmos de aprendizaje automático bajo el Framework Gradient Boosting. XGBoost proporciona un árbol boosting paralelo (también conocido como GBDT, GBM) que resuelve muchos problemas de ciencia de datos de una manera rápida y precisa. [8]

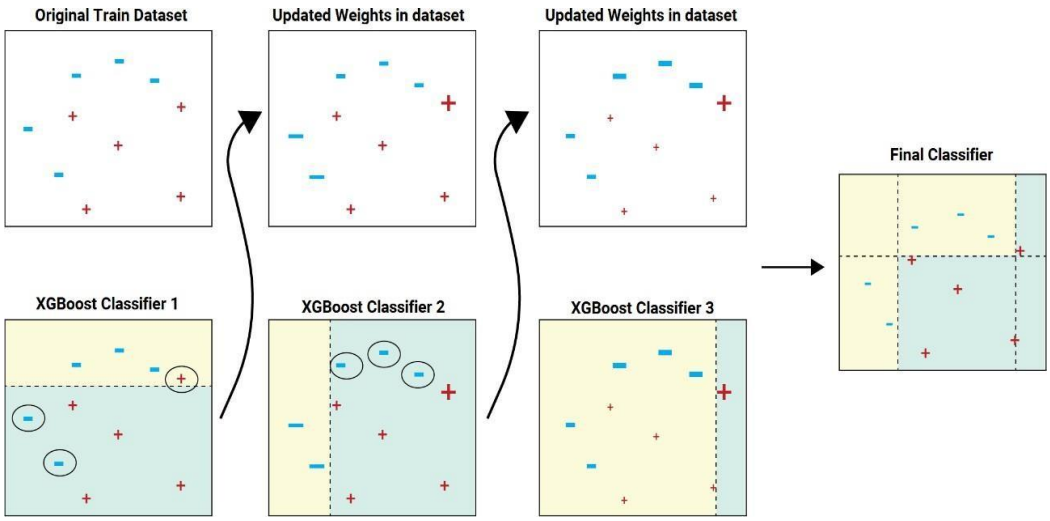


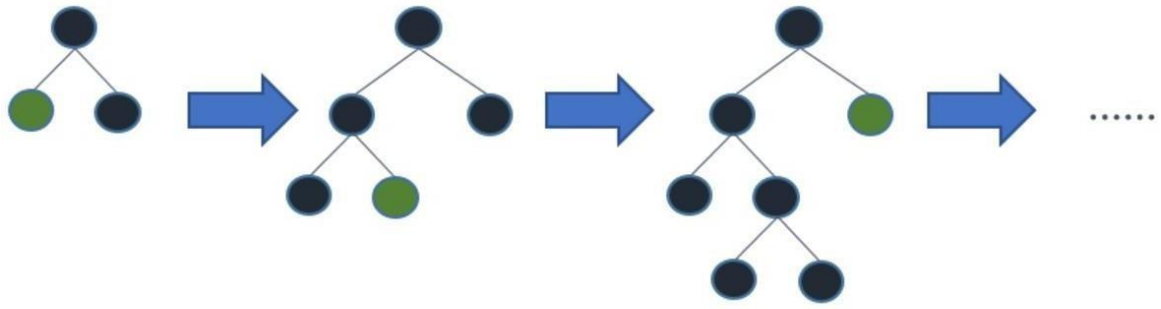
Figura 4.- Diagrama de XGBoost

2.4.2 LightGBM

LightGBM es un framework de gradiente boosting que usa algoritmos de aprendizajes basados en árboles. Está diseñado para ser distribuido y eficiente con las siguientes ventajas:

- Mayor velocidad de entrenamiento y mayor eficiencia.
- Menor uso de memoria.
- Mejor precisión.
- Soporte de aprendizaje paralelo, distribuido y GPU.
- Capaz de manejar datos a gran escala.

La mayoría de los algoritmos de aprendizaje basado en árboles de decisión hacen crecer los árboles primero por nivel (profundidad), LightGBM cultiva árboles priorizando el crecimiento por hojas. [9]



Crecimiento de las hojas del árbol

Figura 5.- Diagrama de LightGBM

2.4.2 Catboost

CatBoost es un algoritmo para potenciar gradientes en árboles de decisión que intenta resolver las características categóricas utilizando una alternativa impulsada por permutación. Se utiliza para búsqueda, sistemas de recomendación, asistente personal, automóviles autónomos,predicción del tiempo y muchas otras tareas.[21]

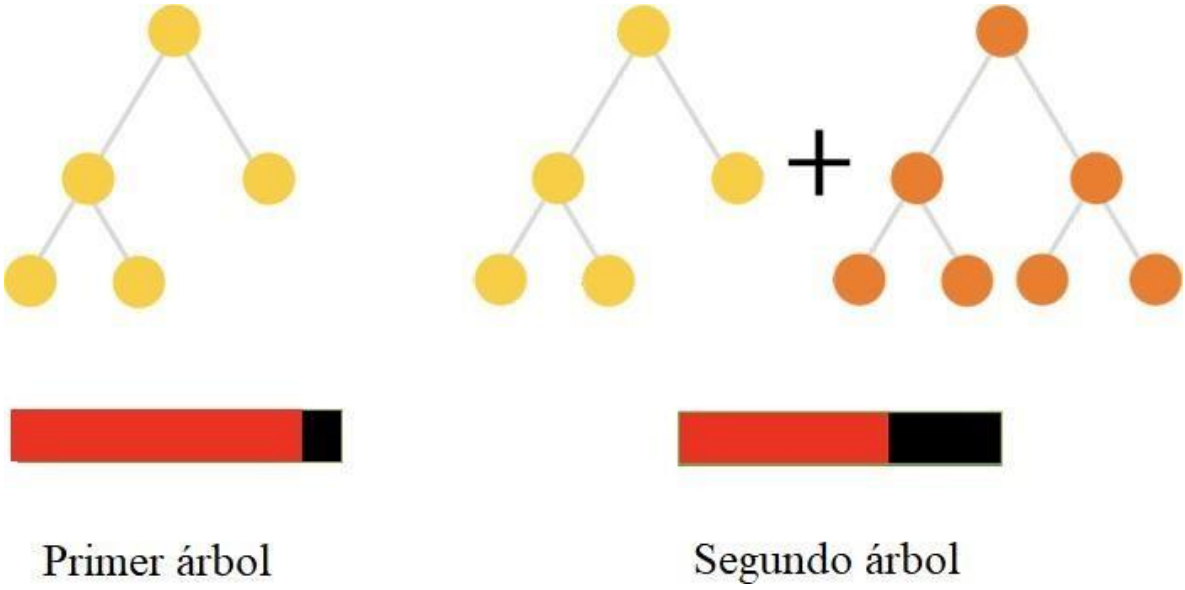


Figura 6.- Diagrama de Catboost

Sus principales características son:

- Manejo nativo para características categóricas
- Entrenamiento rápido de GPU
- Visualizaciones y herramientas para análisis de características y modelos
- Uso de árboles olvidados o árboles simétricos para una ejecución más rápida
- Impulso ordenado para superar el sobreajuste

2.5 Shapley Additive exPlanations

SHAP (Shapley additive explanations) es un modelo basado en la teoría de juegos utilizado para aumentar la transparencia, interpretabilidad y explicar el resultado de cualquier modelo de aprendizaje automático. [22] El método fue propuesto por Lloyd Shapley [23] en 1953.

Los valores de Shapley surgen del contexto donde "n" jugadores participan colectivamente obteniendo una recompensa "p" que se pretende repartir equitativamente a cada uno de los "n" jugadores de acuerdo a la contribución individual, dicha contribución es un valor de Shapley.

En la figura 7, podemos ver como se calcula la contribución individual del jugador A en una coalición compuesta por los jugadores A, B, C y D.

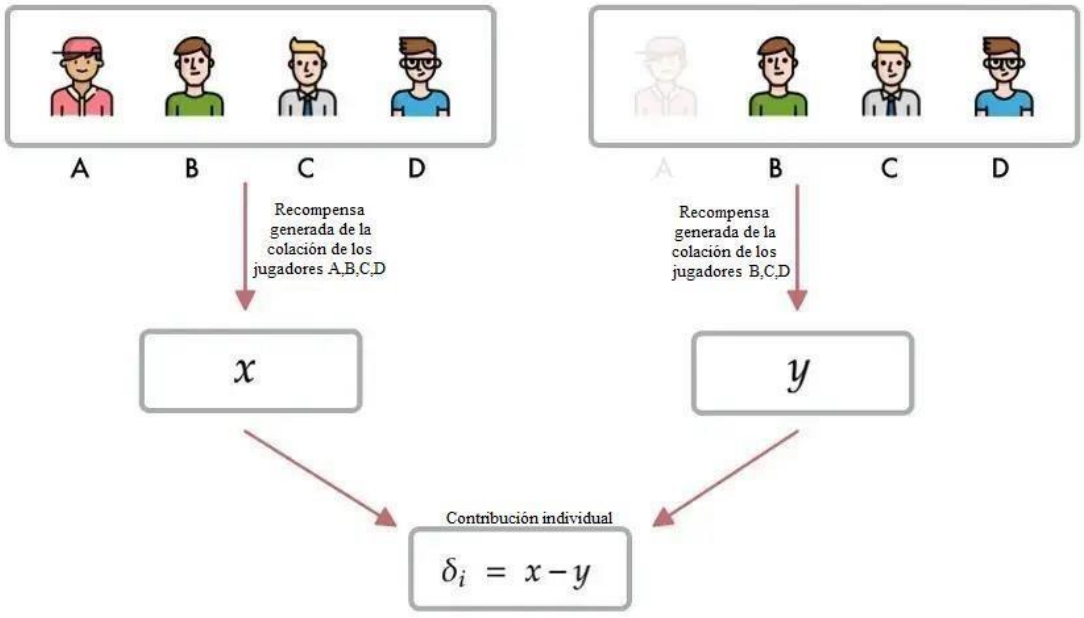


Figura 7.- Calculo de la contribución individual del jugador A en la coalición A, B, D, C

Para calcular el valor Shapley del jugador A, se necesita calcular la contribución individual encada una de las coaliciones posibles y la suma de cada una de las contribuciones individuales nos daría como resultado el valor Shapley del jugador A. [24]

En la figura 8 podemos ver ilustrado el cálculo del valor Shapley del jugador A.

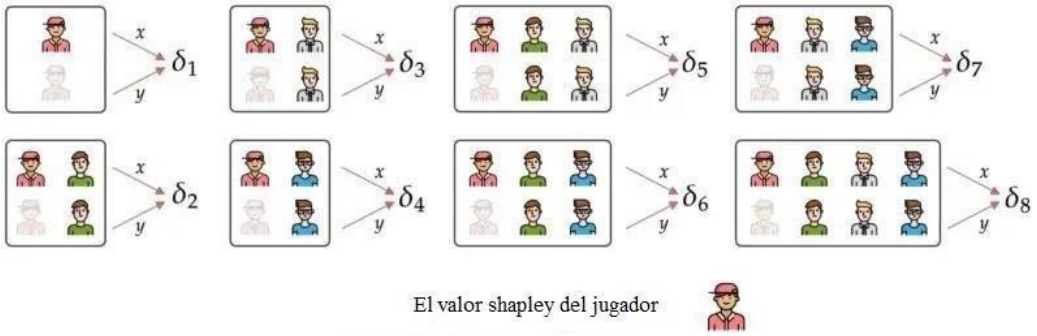


Figura 8.- Calculo del Shap Value del jugador A.

Esto mismo puede ser trasladado al resultado de un modelo de aprendizaje automático, en donde cada uno de los jugadores sería la representado por cada una de las características con la que alimentamos nuestro clasificador.

CAPÍTULO 3: METODOLOGÍA

3.1 Extracción de los datos

Los datos son la parte más importante de los modelos de ML, debido a que a partir de ellos los modelos aprenden a tomar decisiones. Este proyecto se enfoca en la predicción de los ganadores en los partidos de la NBA, por lo cual se priorizó que los datos fuesen obtenidos de su página web oficial.

Para la extracción de los datos se hizo uso de un api de python de uso público llamada **nba_api** [22] con el cuál podremos hacer conexión directa con la base de datos de la NBA obteniendo así bases de datos por temporadas.

Dichas bases de datos contienen las estadísticas grupales hechas por los equipos en cada partido, encontrando características como: fecha, game_id, ganador, puntos, rebotes, robos de balón, bloqueos, asistencia, etc.

TEAM_NAME	GAME_DATE	WL	PTS	FGM	FG3M	OREB	DREB	REB	AST	STL	BLK
Golden State Warriors	14/11/2022	W	132	47	23	5	35	40	35	9	3
Golden State Warriors	13/11/2022	L	115	42	16	8	35	43	27	5	5
Golden State Warriors	11/11/2022	W	106	40	15	8	36	44	29	5	3

Tabla 2.- Muestra de los datos crudos

La información de apuestas fue extraída del sitio web de la casa de apuestas 5DIMES originaria de las vegas, a través de SPORT REVIEW ONLINE; Esto para los experimentos con datos históricos.

Team	Date	VH	1st	2nd	3rd	4th	Open	Close	ML	2H
Philadelphia	1018	V	29	34	25	29	229	216	135	107
Boston	1018	H	24	39	35	28	7	3	-155	2
LALakers	1018	V	22	30	19	38	229.5	223.5	260	114
GoldenState	1018	H	25	34	32	32	6.5	7.5	-310	2

Tabla 3.- Muestra de los datos de la casa de apuestas

3.2 Procesamiento de los datos

Todo el procesamiento de los datos explicado a continuación se realizó en el lenguaje de programación Python, debido a su accesibilidad y librerías para el manejo de los datos.

Teniendo la base de datos empezamos por dividir la base de datos de la temporada a analizar en 30 archivos csv, un archivo correspondiente por cada equipo de la liga.

Tras esto, debido a que el desempeño de los equipos en pretemporada y playoffs es muy errático, nos quedamos solamente con la temporada regular. Para este filtrado nos apoyamos en la columna llamada "**GAME_ID**", dicha columna cuenta con un formato de 8 dígitos la cual se describe en la figura 9.

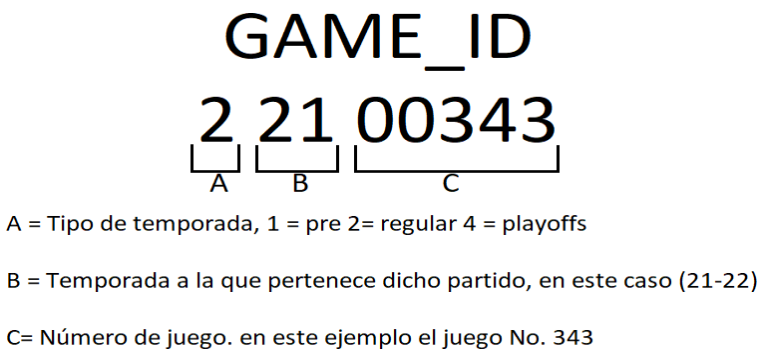


Figura 9.- Formato de la columna "GAME_ID"

Filtramos los partidos de forma en que solo nos quedamos con los juegos cuyos GAME_ID empiecen con un número 2, es decir, solo con juegos de la temporada regular.

Tras el filtrado, Analizamos las características con las que contamos para seleccionar las que, bajo nuestro criterio, son las de mayor importancia o las más influyentes en el resultado del partido, intentando reflejar el desempeño ofensivo y defensivo de los equipos se seleccionaron 5 características por equipos: puntos, rebotes totales, robos de balón, puntos hechos en contra y rebotes hechos en contra.

Cada vector de nuestro vector de características representará el desempeño del equipo hasta la fecha del partido en cuestión, siendo estos promedios de cada uno de las características. En el intento de responder a la hipótesis planteada anteriormente, se procesan las características tomando en cuenta si el partido en cuestión se está jugando como local o como visitante.

Generamos una sexta característica correspondiente al win-rate del equipo, esto promediando los últimos seis juegos jugados por cada equipo, siendo 6 el número de juegos que mejor desempeñó mostró al momento de generar los modelos de ML.

Se utilizó la base de datos de la casa de apuestas para generar una séptima característica, la cuál es la probabilidad dada por la casa de apuestas a cada equipo de ganar el encuentro, para ello hicimos uso de la siguiente ecuación:

$$Prob = \frac{100}{ML + 100} \quad Si\ ML \geq 0$$

$$Prob = \frac{-ML}{(-ML) + 100} \quad Si\ ML < 0$$

Donde Prob es la probabilidad, ML es el *Money Line* o momio de la casa de apuestas.

Los procedimientos anteriores se aplican a cada uno de los 30 archivos csv, los cuales concatenaremos mediante el game_id.

Tras lo anterior, y debido a la naturaleza de la concatenación en Python, desordenamos de forma aleatoria cada uno de los vectores, asegurándonos así que ningún equipo se encuentre siempre de un lado del vector (evitando así posibles ruidos en los modelos).

Por último, generamos una nueva columna en nuestro vector correspondiente al target que los modelos predecirá, teniendo un valor de 0 si gana el equipo que se encuentra del lado derecho de nuestro vector y 1 el equipo del lado izquierdo.

3.3 Modelos

A partir de la investigación y sumando lo visto en los trabajos relacionados, se seleccionaron modelos con potencial buen desempeño para esta tarea. Los seleccionados fueron:

- *K-Nearest Neighbor* (KNN)
- *Support Vector Machine* (SVM)
- *Multi Layer Perceptron* (MLP)
- Random Forest (RF)
- Gradiet Boost (GB)
- Regresión Logística (RL)
- XGBoost
- LightGBM
- Catboost

3.3.1 La mejor versión de los modelos

Si bien, cada uno de los modelos cuentan con parámetros predefinidos a la hora de utilizarlos, antes de empezar a poner a prueba a los modelos, se decidió buscar la mejor combinación de parámetros para cada uno de los modelos.

Para la búsqueda de los mejores parámetros disponibles se hace uso de la función GridSearchCV de la librería Sklearn la cual, como explicamos en el marco teórico, es una búsqueda de hiper parámetros de con fuerza bruta, es decir, prueba todas las combinaciones de hiperparametros hasta encontrar la que mejor desempeño muestra en nuestro objetivo.

Para implementar esta función, primero recurrimos a la documentación del modelo a optimizar, para así visualizar los hiperparametros con los que cuenta y sus valores.

En la figura 10 se ilustran los hiperparametros del modelo regresión logística de la librería **sklearn**.

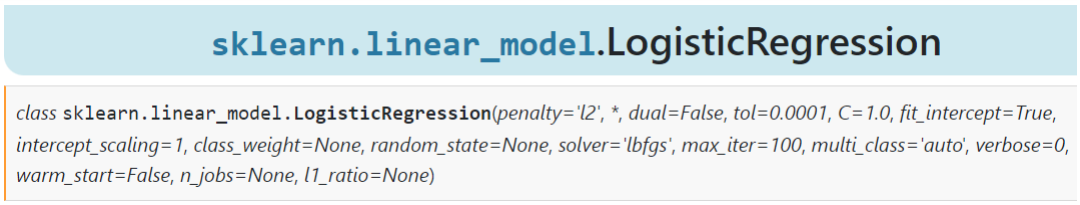


Figura 10.- Hiper parámetros de regresión logística

Conociendo los hiperparametros podemos determinar nuestro espacio de búsqueda, en la figura 11 se muestra el espacio de búsqueda para el modelo regresión logística.

```
space = {'solver':['newton-cg', 'lbfgs', 'liblinear'],
         'penalty':['none', 'l1', 'l2', 'elasticnet'],
         'C':[1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1, 10, 100]}
```

Figura 11.- Espacio de búsqueda del GridSearch para regresión Logística

Por último, ejecutamos la función GridSearchCV, dándole los parámetros del modelo, el espacio de búsqueda y cual será la métrica de evaluación, en este caso diremos que los mejores hiperparametros serán los que nos den la mejor precisión.

En la figura 12 vemos la ejecución y el resultado dado para la regresión logística.

```
search = GridSearchCV(LogisticRegression(), space, scoring='accuracy')

result = search.fit(data, target)
# summarize result
print('Best Score: %s' % result.best_score_)
print('Best Hyperparameters: %s' % result.best_params_)
```

```
Best Score: 0.6756582633053224
Best Hyperparameters: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
```

Figura 12.- Ejecución de la función GridSearchCV

Los resultados los usaremos para tener un modelo de regresión logística mejorado para predecir partidos de la NBA, en la figura 13 vemos el modelo ya con los parámetros ajustados.

```
model = LogisticRegression(penalty='l2', C=10, solver='lbfgs')
```

Figura 13.- Modelo con los mejores hiper parámetros

Este procedimiento se seguirá con todos los demás modelos a utilizar, para así asegurarnos de que estén en su mejor versión.

3.4 Predicciones en base a datos históricos.

Para este apartado, se procesó las bases de datos de las temporadas 2018-2019,2019-2020 y 2020-2021, siendo estas las últimas finalizadas para el momento de la realización de estos experimentos.

Haciendo uso de la librería Scikitlearn, dividimos de forma aleatoria nuestro vector de características, los datos ya pre procesados, en un 80-20, en donde 80% será nuestro data_train (entrenamiento del modelo) y el 20% restante nuestro data_test (prueba del modelo).

Entrenamos cada uno de los modelos, previamente mejorados, con el data_train y procedemos a predecir nuestro data_test, debido a las pizcas de aleatoriedad que hay en la realizaciónen nuestro vector y en la división del mismo, el entrenamiento y prueba se realizó 100 veces en cadamodelo, esto con la finalidad de obtener resultados más certeros.

Por último, de esas 100 iteraciones, rescatamos los *accuaracy* de cada modelo en cada iteración, el cual será nuestro parámetro para determinar los resultados obtenidos.

3.5 Predicciones en tiempo real.

En base a los resultados obtenidos con el experimento de los datos históricos. Se descartó 4 de los 9 modelos de clasificación, quedándonos solamente con: regresión logística, *random forest*, Catboost, XGBoost y LightGBM; los 5 modelos con mejor desempeño.

Para el desarrollo de este experimento, se continuo con el mismo procesamiento de los datos.

Al ser un experimento en tiempo real, el conjunto de datos con los que se entrenó el modelo eran todos los juegos anteriores a los partidos a predecir y el conjunto de prueba (en este caso los partidos a predecir) eran los que se iban a jugar el día presente.

3.6 Importancia de las características.

Con la intención de entender mejor las predicciones hechas por los modelos y la manera en la que aprenden, se decidió hacer un análisis de la importancia que le dan los modelos a cada una de las características que se le dan para predecir.

Para realizar este análisis, se hizo uso de la librería SHAP, con la cual empezamos especificando el explicador.

En el caso de los modelos basados en árbol la librería cuenta con una función nativa llamada TreeExplainer, a la cuál solo hace falta definirle el modelo a utilizar, esto se muestra en la figura 14.

```
model = xgb.XGBClassifier(colsample_bytree = 0.7,
model.fit(data_train1, target_train1)
explainer = shap.TreeExplainer(model)
```

Figura 14.- Explainer para modelos basados en arboles

En el caso de modelos lineales, como es el caso de la regresión logística mostrada en la figura 15, debemos hacer uso de la función Explainer, en donde debemos especificar manualmente los datos de entrenamiento con los que se alimentaron al modelo y los nombres de las características.

```
model = LogisticRegression(penalty='l2', C=10, solver='lbfgs')
model.fit(data_train3, target_train3)
explainer = shap.Explainer(model, data_train3, feature_names=data_train3.columns)
```

Figura 15.- Explainer para modelos lineales

Una vez definido el Explainer, solamente hace falta alimentarlo con los datos de muestra y definir la grafica con la que se mostrarán los resultados, como se puede apreciar en la figura 16.

```
shap_values = explainer(data_test3)
shap.summary_plot(shap_values, data_test3)
```

Figura 16.- Graficación de los Shap Value

CAPÍTULO 4: RESULTADOS

Los experimentos realizados con los vectores de características generados a partir de las estadísticas rescatadas de la NBA arrojaron resultados diversos. Este conjunto de datos fue utilizado en distintos clasificadores.

A continuación, se mostrarán los resultados obtenidos en los dos experimentos principales, predicción con datos históricos y predicciones en tiempo real.

4.1 Resultados históricos.

Los experimentos para las predicciones con datos históricos fueron realizados con a partir de las últimas 3 temporadas concluidas a fecha de esta investigación (18-19,19-20, 20-21). Se decidió realizar con más de una temporada, debido a que ninguna temporada es igual, esperando obtener variaciones, pequeñas, en cada una de las temporadas.

Cabe destacar que dado que nuestro objetivo es hacer que el modelo aprenda y no solo encuentre solamente patrones, se ha programado de forma que se reorganice el *dataset* de forma aleatoria. Por este factor de aleatoriedad agregado se ha decidido iterar 100 veces por experimento.

Los modelos a analizar son: K nearest neighboor, Support Vector Machine, Random Forest, Gradient Boost, Multi Layer Perceptron, Logistic Regression, XGB, Catboost y LightGBM

Por último, los *datasets* se dividen en 80% para el entrenamiento de los modelos y 20% para la prueba de los mismos.

4.1.1 Temporada 18-19.

El *dataset* de esta temporada (los partidos jugados) fue de 1212 datos. El resultado de los modelos a lo largo de las iteraciones realizadas se presenta en la tabla 4.

Modelo	Mejor desempeño (%)	Peor desempeño (%)	Promedio (%)
KNN	0.67	0.55	0.61
SVC	0.70	0.52	0.62
RF	0.73	0.62	0.67
GB	0.70	0.57	0.63
MLP	0.69	0.49	0.62
RL	0.74	0.62	0.67
XGB	0.68	0.58	0.63
CAT	0.73	0.62	0.67
LIGHT	0.68	0.58	0.64

Tabla 4.- Resultados temporada 18-19

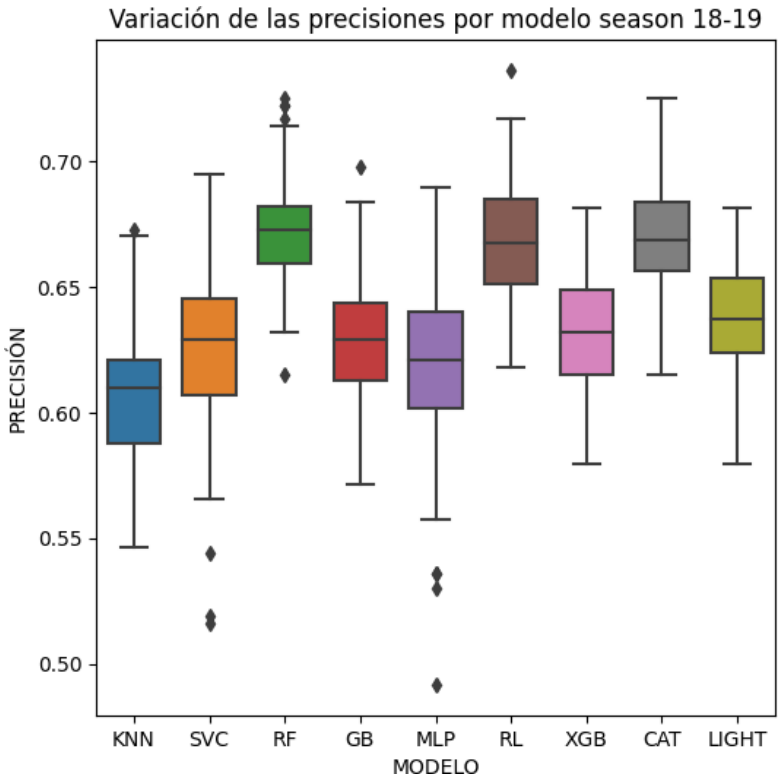


Figura 17.- Varianza temporada 18-19

Para complementar la visualización de los resultados de cada temporada, también se realizó una gráfica tipo *boxplot*, ilustrada en la figura 14, que contiene la varianza en los resultados de los modelos en las iteraciones realizadas.

Los resultados en esta temporada denotan una superioridad por parte de 3 de los modelos, siendo catboost, random forest y regresión logisitca, este último con el mejor desempeño general, los 3 con mejores desempeños promediados, con 67% de precisión.

4.1.2 Temporada 19-20.

Para esta temporada, el *dataset* contó con 1043 datos (partidos), y de igual forma se generó la tabla 5 y una gráfica *boxplot*, figura 15, conteniendo los desempeños de los 9 modelos.

Modelo	Mejor desempeño (%)	Peor desempeño (%)	Promedio (%)
KNN	0.66	0.52	0.59
SVC	0.66	0.46	0.59
RF	0.72	0.60	0.66
GB	0.68	0.55	0.62
MLP	0.69	0.52	0.62
RL	0.73	0.60	0.66
XGB	0.70	0.55	0.62
CAT	0.71	0.60	0.66
LIGHT	0.67	0.57	0.63

Tabla 5.- Resultados temporada 19-20

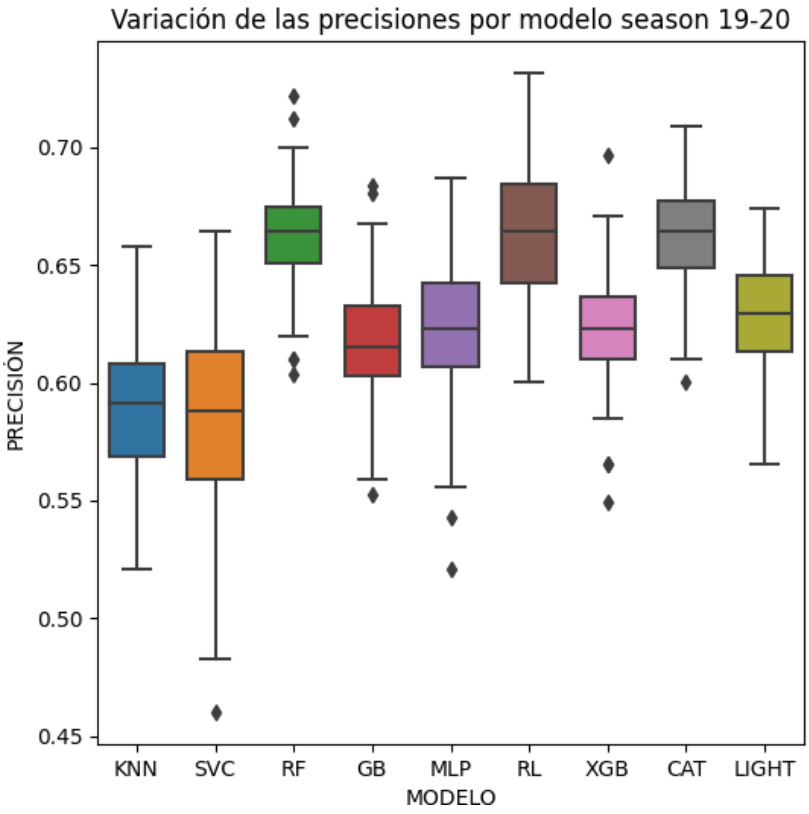


Figura 18.- Varianza temporada 19-20

En esta temporada se sigue demostrando la superioridad de los mismo tres modelos que la temporada pasada, con la mínima diferencia que el desempeño en promedio en esta temporada disminuyo de un 67% a 66%.

4.1.3 Temporada 20-21.

Para la temporada 20220-2021, se contó con 1143 datos (partidos), que de igual forma se segmento en 80-20 e se itero en 100 ocasiones por cada uno de los modelos

Modelo	Mejor desempeño (%)	Peor desempeño (%)	Promedio (%)
KNN	0.63	0.51	0.58
SVC	0.66	0.45	0.56
RF	0.75	0.61	0.66
GB	0.64	0.55	0.60
MLP	0.68	0.47	0.61
RL	0.70	0.58	0.65
XGB	0.67	0.53	0.60
CAT	0.73	0.62	0.65
LIGHT	0.66	0.55	0.61

Tabla 6.- Resultados temporada 20-21

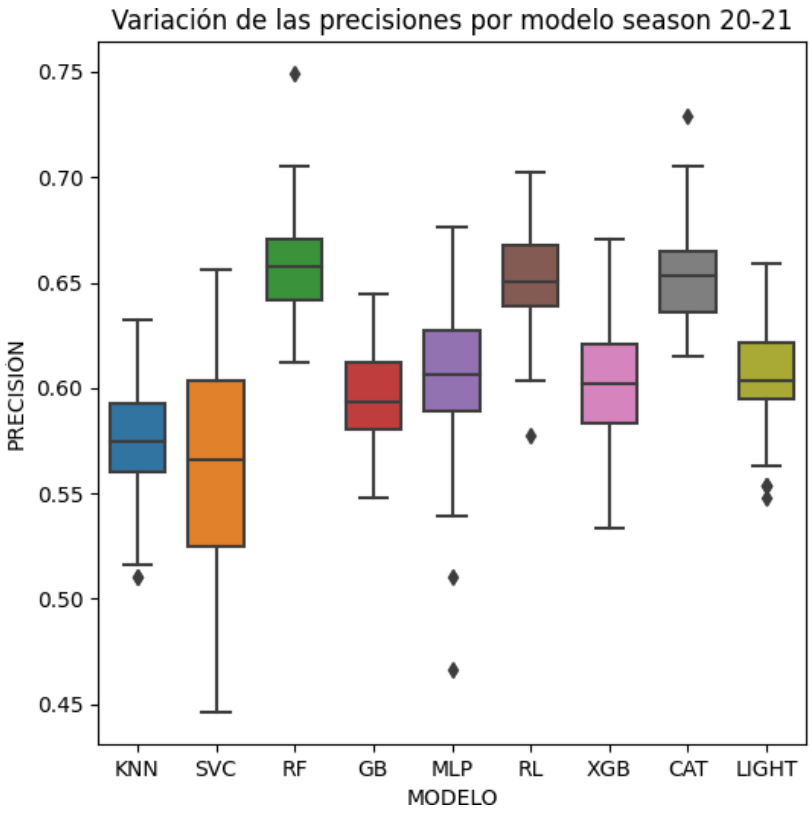


Figura 19.- Varianza temporada 20-21

Para esta temporada vemos un pequeño cambió, siendo ahora el RF el que mejor se desempeñó, pero sin haber un cambio significativo con respecto a las 2 temporadas anteriores.

4.2 Resultados en tiempo real.

4.2.1 Primer experimento (23-03-22/10-04-22.)

Este experimento comenzó el 23 de marzo al 10 de abril del 2022 (126 partidos), siendo los últimos juegos de la temporada regular de la temporada 2021-2022 de la NBA.

En esta ocasión solamente se registraron los resultados de la regresión logística, esto debido a que fue el modelo que mejor desempeño demostró en el experimento anterior.

Cabe destacar que debido a la naturaleza de este experimento y en favor de contar con una referencia contra la que comparar el desempeño del modelo, también se hizo registro del desempeño mostrado por la casa de apuestas

En la figura 17 vemos el desempeño del modelo en este lapso de tiempo en contraste con el realizado por la casa de apuestas.

Figura 20.- Comparativa de la casa de apuesta vs modelo

4.2.2 Segundo experimento (22-10-2022 / 26-11-2022.)

Debido a que el experimento anterior se empezó tarde en la temporada 2021-2022. Se decidió muestrear los inicios de la temporada 2022-2023, el experimento empezó el día 22 de octubre del presente año y se concluyó el 26 de noviembre del 2022.

Para la exposición del desempeño de los modelos en este experimento se decidió plasmar los resultados diarios de cada uno de ellos en un histograma, en donde cada una de las barras representa una jornada de juegos, el color naranja en las barras significa el número de juegos que fueron predichos de manera correcta. De la figura 18 a la 22 se representan los resultados de cada uno de los modelos.

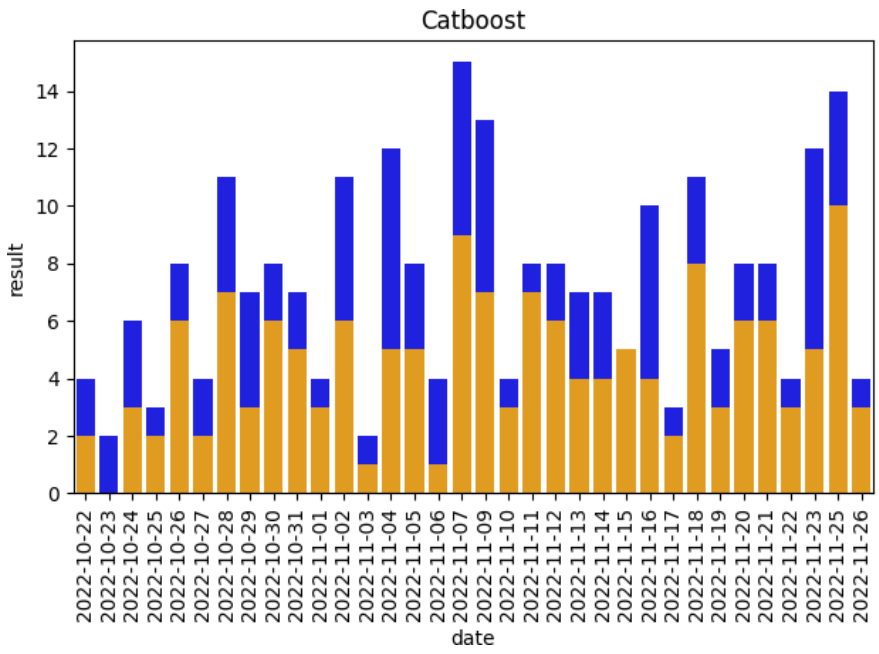


Figura 21.- Desempeño diario del modelo Catboost

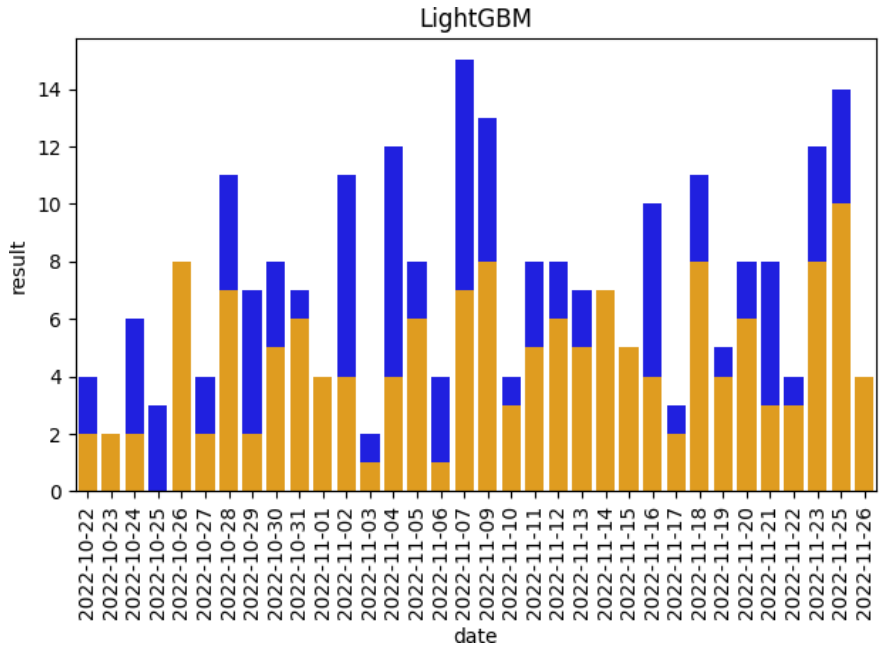


Figura 22.- Desempeño diario del modelo LightGBM

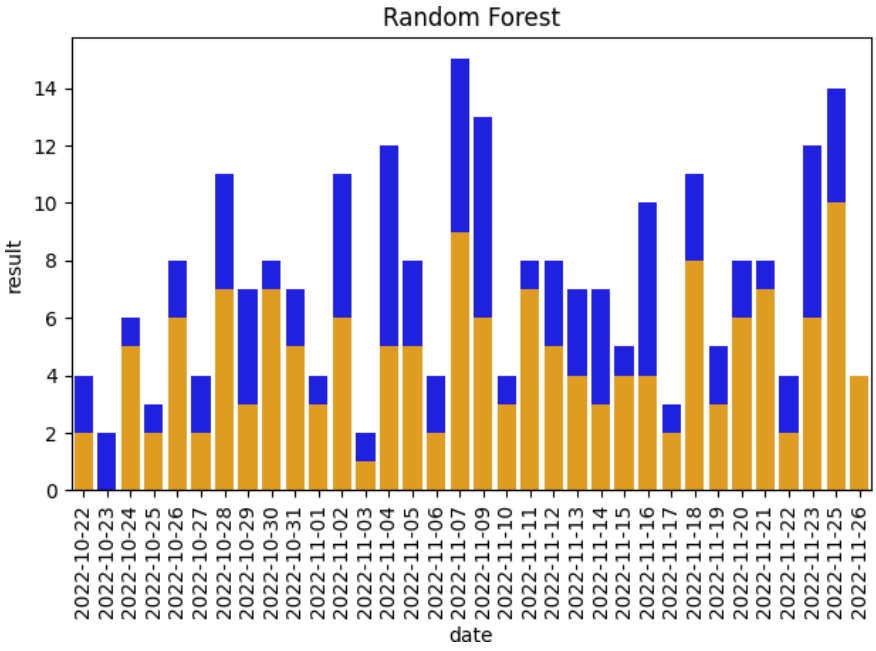


Figura 23.- Desempeño diario del modelo *Random Forest*

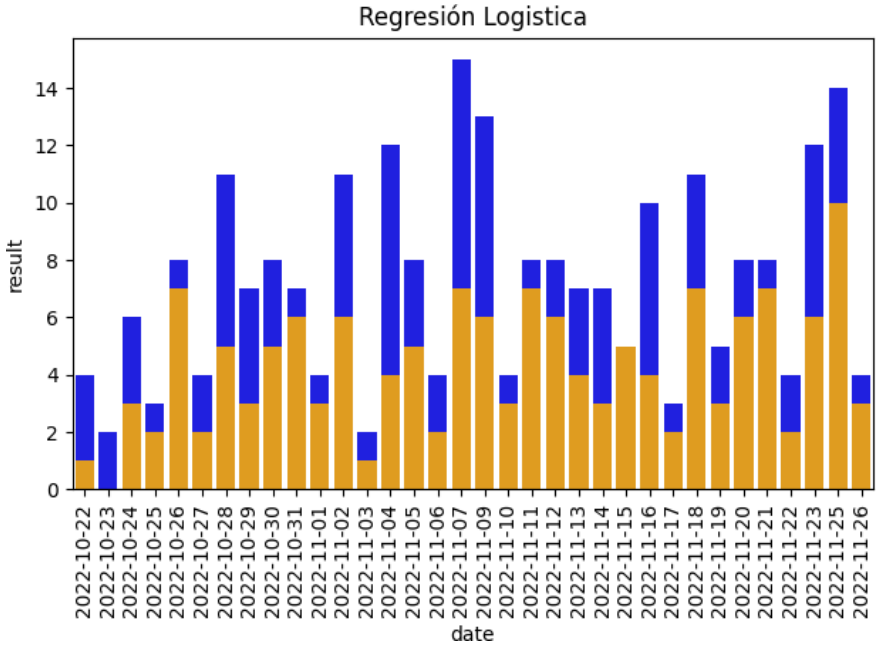


Figura 24.- Desempeño diario del modelo Regresión logística

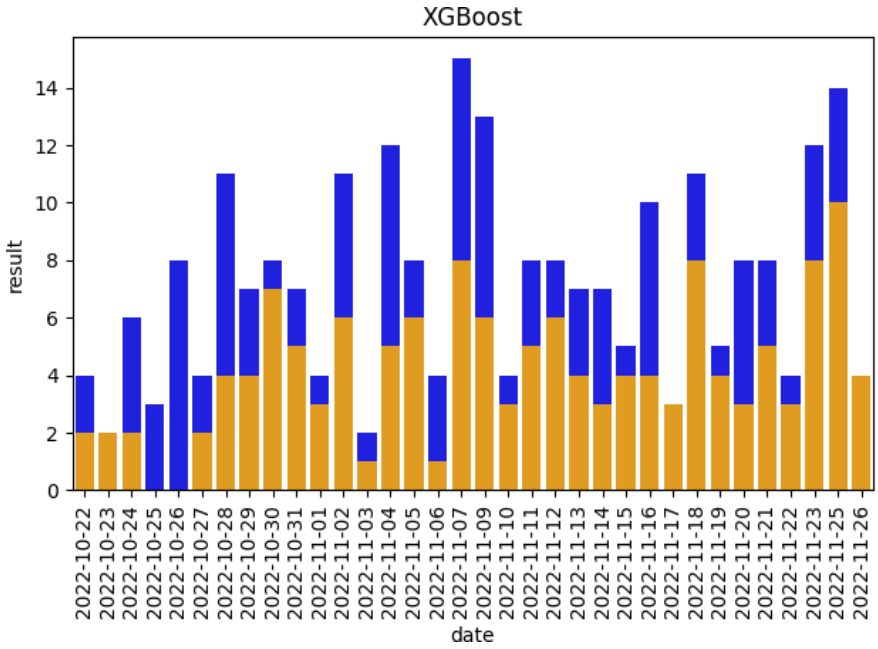


Figura 25.- Desempeño diario del modelo XGBoost

A su vez, en la tabla 7, muestra el desempeño general logrado por cada uno de los modelos a lo largo del experimento.

Modelo	Aciertos	Promedio
RL	146/247	59.10%
CAT	152/247	61.53%
RF	154/247	62.34%
XGB	141/247	57.08%
LIGHT	154/247	62.34%

Tabla 7.- Desempeño de los modelos en tiempo real

De estos resultados cabe destacar que ahora los 2 mejores modelos para las predicciones en tiempo real son el Random Forest y como sorpresa el LightGBM, ambos alcanzando un 62.34% de precisión.

Al ser inicio de temporada los modelos no cuentan con la misma cantidad de datos con las que entrenarse como lo fue en el experimento anterior, el cual era el último mes para finalizar la temporada. Debido a esto, las primeras "predicciones" de los modelos LightGBM y XGBoost, muestra contenida en las tablas 8 y 9, no eran más que el lanzamiento de una moneda, es decir daban 50% de probabilidad de ganar para ambos equipos.

Modelo	DATE	VISITANTE	PROB_0	LOCAL	PROB_1
LIGHT	22/10/2022	SAS	0.5	PHI	0.5
LIGHT	22/10/2022	DET	0.5	IND	0.5
LIGHT	22/10/2022	TOR	0.5	MIA	0.5
LIGHT	22/10/2022	LAC	0.5	SAC	0.5

Tabla 8.- Muestra de las primeras predicciones del modelo LightGBM

Modelo	DATE	VISITANTE	PROB_0	LOCAL	PROB_1
XGB	22/10/2022	SAS	0.5	PHI	0.5
XGB	22/10/2022	DET	0.5	IND	0.5
XGB	22/10/2022	TOR	0.5	MIA	0.5
XGB	22/10/2022	LAC	0.5	SAC	0.5

Tabla 9.- Muestra de las primeras predicciones de modelo XGBoost

Esto a comparación de los demás modelos, cuyas muestras se encuentran en la tabla 10. 11 y 12, los cuales a pesar de la poca información intentaban predecir.

MODELO	DATE	VISITANTE	PROB_0	LOCAL	PROB_1
RL	22/10/2022	SAS	0.02330504	PHI	0.97669496
RL	22/10/2022	DET	0.90749796	IND	0.09250204
RL	22/10/2022	TOR	0.18059768	MIA	0.81940232
RL	22/10/2022	LAC	0.16923422	SAC	0.83076578

Tabla 10.- Muestra de las primeras predicciones de Regresión logística

Modelo	DATE	VISITANTE	PROB_0	LOCAL	PROB_1
RF	22/10/2022	SAS	0.57	PHI	0.43
RF	22/10/2022	DET	0.58	IND	0.42
RF	22/10/2022	TOR	0.46	MIA	0.54
RF	22/10/2022	LAC	0.47	SAC	0.53

Tabla 11.- Muestra de las primeras predicciones de *Random Forest*

MODELO	DATE	VISITANTE	PROB_0	LOCAL	PROB_1
CAT	22/10/2022	SAS	0.55328521	PHI	0.44671479
CAT	22/10/2022	DET	0.52085995	IND	0.47914005
CAT	22/10/2022	TOR	0.48842976	MIA	0.51157024
CAT	22/10/2022	LAC	0.43374421	SAC	0.56625579

Tabla 12.- Muestra de las primeras predicciones de Catboost

4.3 Shap values.

Dado que la intención de este análisis era tanto el de ver cuál era la importancia dada por los modelos a las características como si la importancia de estas cambiaba a lo largo de la temporada, se analizaron los shap values dos fechas de la temporada 2021-2022, la primera siendo el 15 de noviembre del 2021 y la segunda el 9 de abril de 2022.

Para este análisis se entrenaron a los modelos con todos los juegos pasados a la fecha a analizar y los juegos que se jugaron en esa fecha fueron nuestra prueba.

4.3.1 15 de noviembre de 2021

Al escoger esta fecha se intenta representar los inicios de la temporada, siendo esta fecha el día número 24 de juegos de la temporada 2021-2022.

Los resultados se muestran en una gráfica de puntos, donde el color rojo en los puntos significa que la predicción se acerca más al valor 1 y el color azul significa que la predicción se acerca más al valor 0.

Para la correcta interpretación de dichas gráficas, se deben leer como si fuesen renglones, cada punto representa un juego y el significado de la distancia en el eje de las X significa la cantidad de dicha característica, mientras más a la derecha se encuentre el punto, más grande es su valor.

Por último, las características se ordenan de la de forma descendente, siendo la característica que se encuentra más arriba en la gráfica la que nuestro modelo más importancia le da y la que se encuentra al final la de menor importancia.

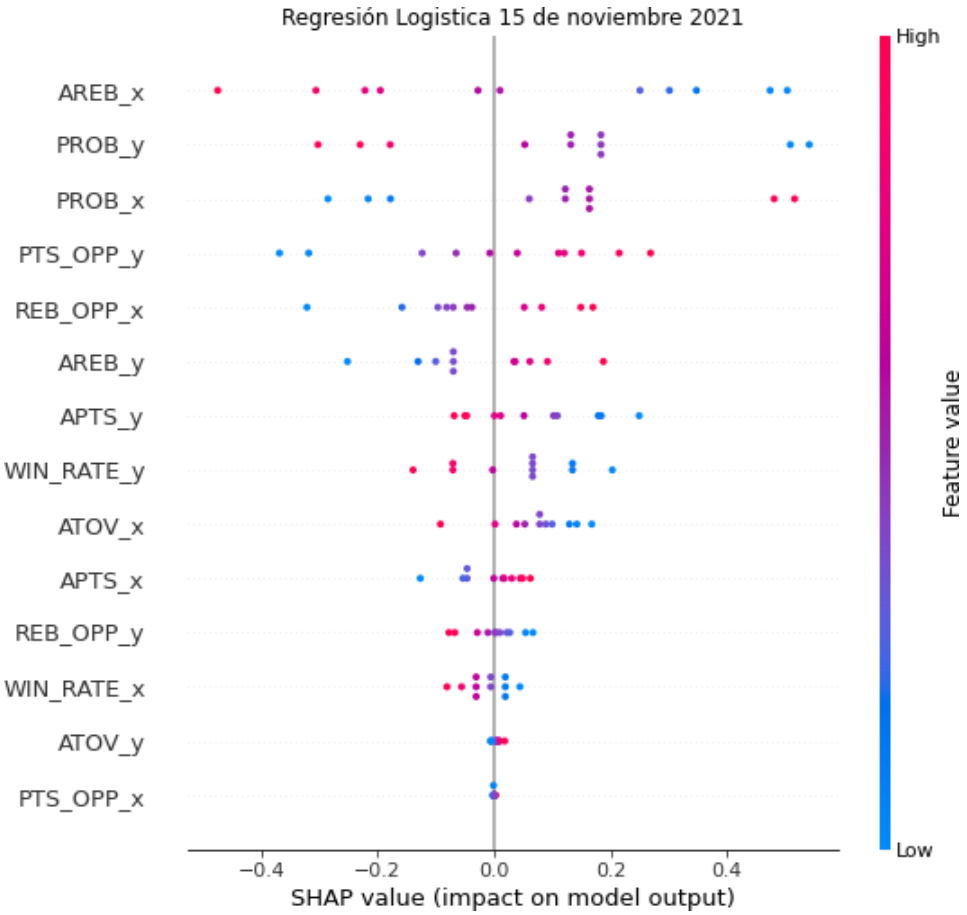


Figura 26.- Shap values regresión logística 15-11-2021

En la figura 26 se encuentran la representación de los Shap values correspondientes a la regresión logística, en la cual podemos ver que la característica que mayor importancia le da este modelo son los promedios de los rebotes totales hechos por el equipo local y los puntos hechos en promedio contra el equipo local la característica con menor importancia.

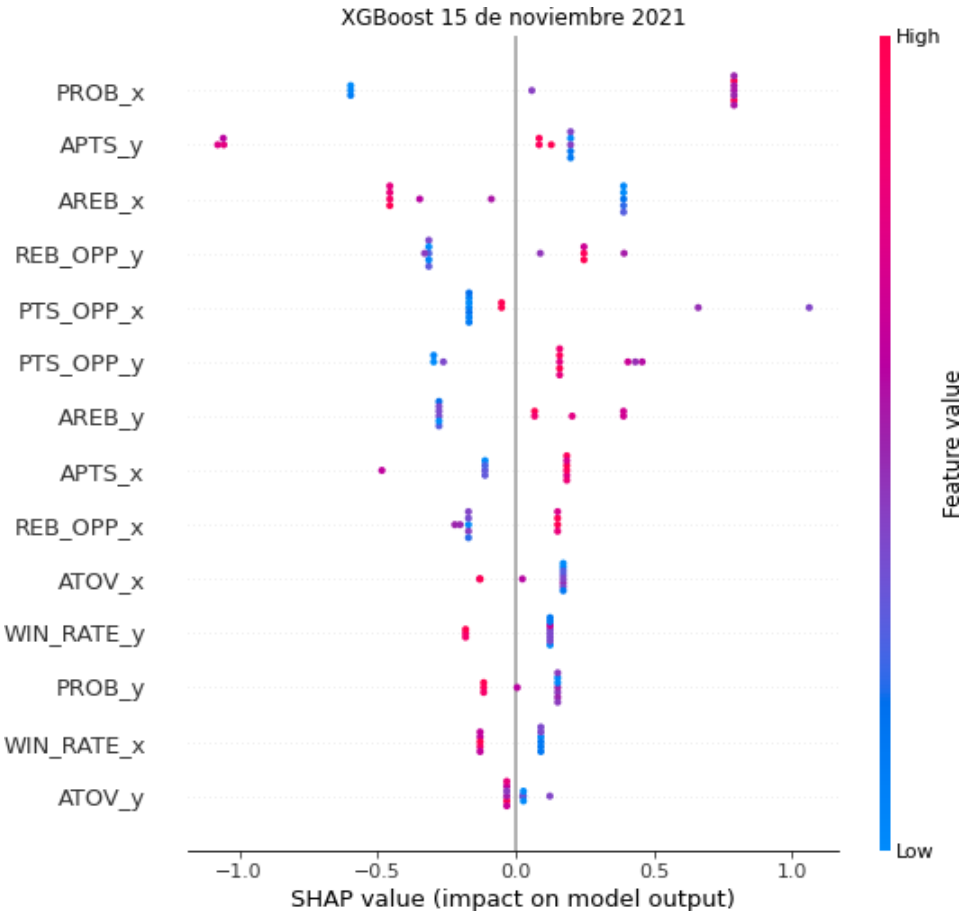


Figura 27.- Shap values XGBoost 15-11-2021

En el caso del modelo XGBoost, representado por la figura 27, la característica con mayor importancia es la probabilidad de la casa de apuestas de que el equipo local gané y la de menor importancia las pérdidas de balón del equipo visitante.

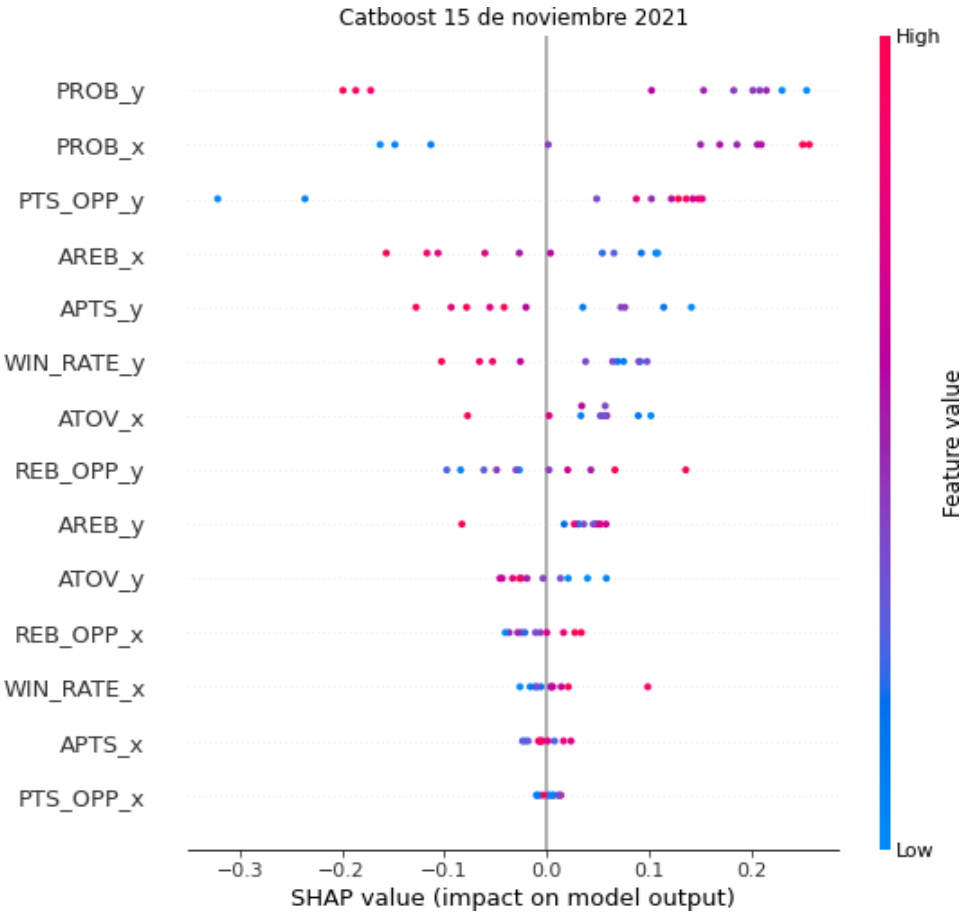


Figura 28.- Shap values Catboost 15-11-2021

En la figura 28, podemos ver que las dos características a la que Catboost más importancia da son a las probabilidades dadas por la casa de apuestas, mientras que la de menor importancia son los puntos hechos en promedio en contra del equipo local.

4.3.2 09 de marzo de 2022

La segunda fecha analizada fue el 9 de marzo de 2022, esta fecha representa la recta final de la temporada, en donde se espera que con la mayor cantidad de datos para entrenar la distribución de la importancia de las características sea distinta.

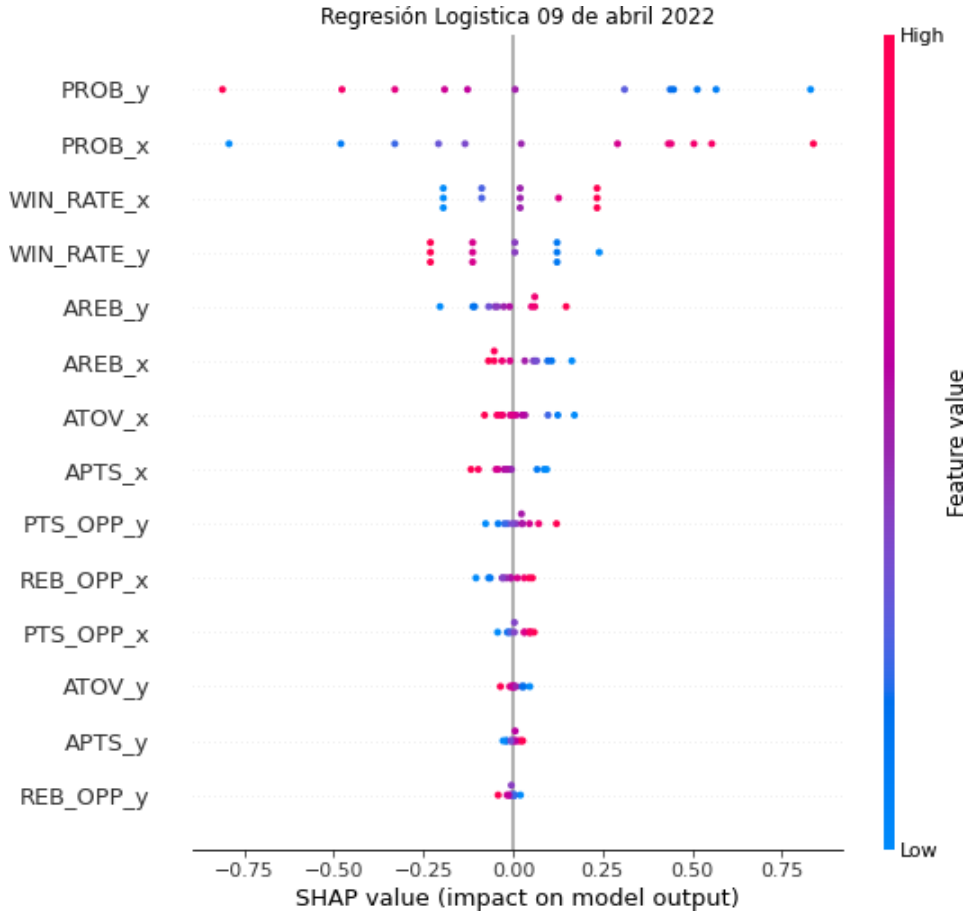


Figura 29.- Shap values regresión logística 09-03-2022

En esta ocasión podemos ver la figura 29 como ahora las dos características con mayor importancia para la regresión logística son las probabilidades de la casa de apuestas y la de menor importancia los rebotes promedios realizados en contra del equipo visitante.

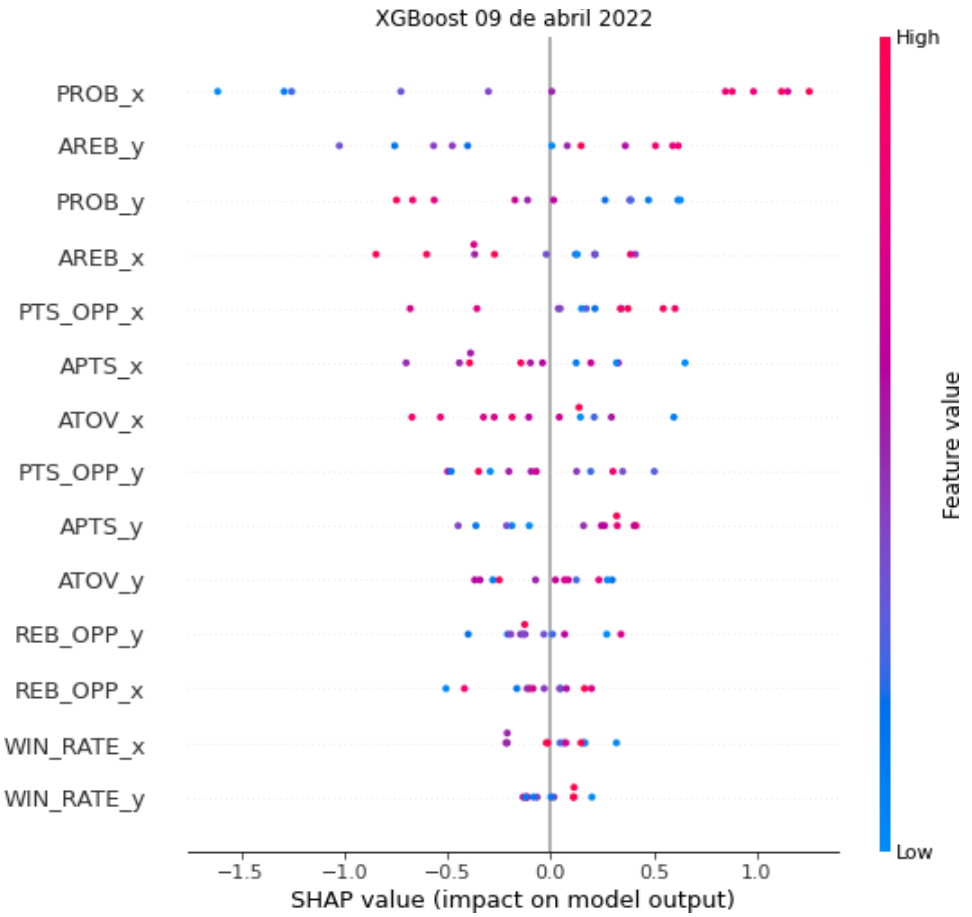


Figura 30.- Shap values XGBoost 09-03-2022

Como podemos ver en los shap values de esta fecha para el modelo XGBoost en la figura 30. Sigue siendo la probabilidad de la casa de apuesta la característica de mayor importancia, pero sorprende que la de menor importancia sea el *win rate* de ambos equipos.

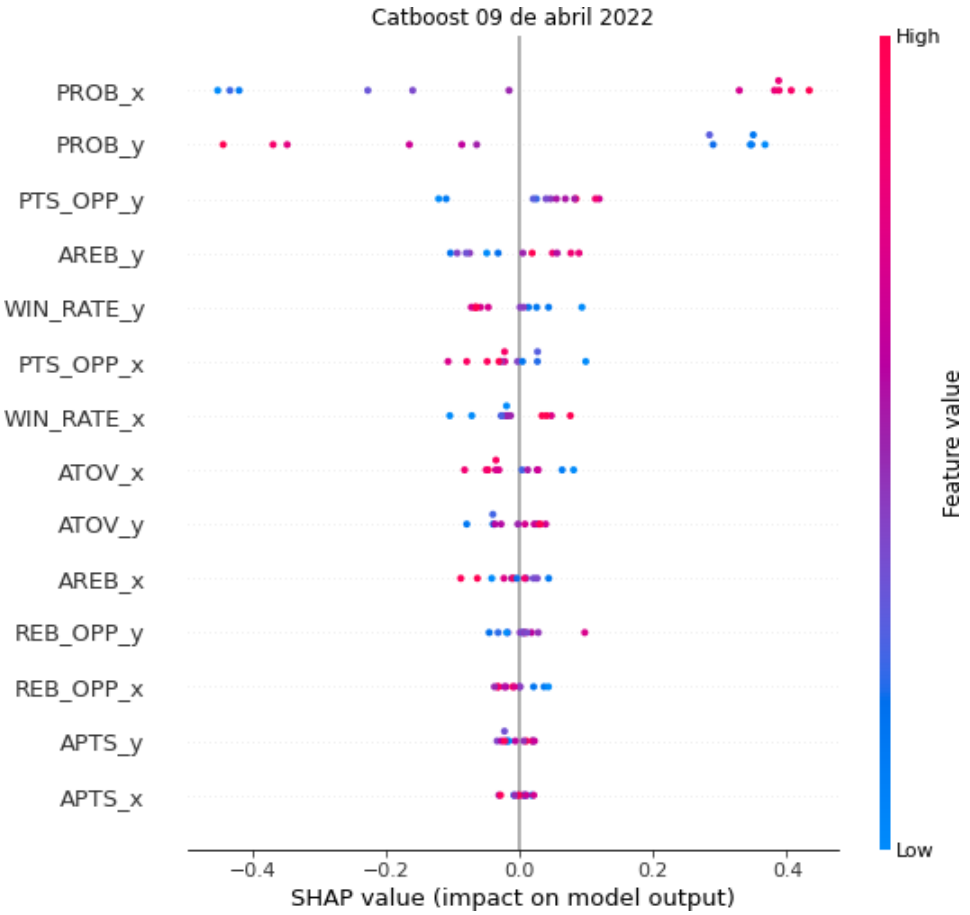


Figura 31.- Shap values catboost 09-03-2022

Los resultados del modelo Catboost (figura 31) las características más importantes no sufrieron cambios, pero es destacable que las características de menor importancias ahora sean los puntos promedios hechos por ambos equipos.

4.3.3 LightGBM y Random Forest

La documentación de la librería SHAP no deja muy en claro el motivo, pero, los modelos lightGBM y random forest no son compatibles con gráficas de puntos, en su defecto los puntos son sustituidos por barras.

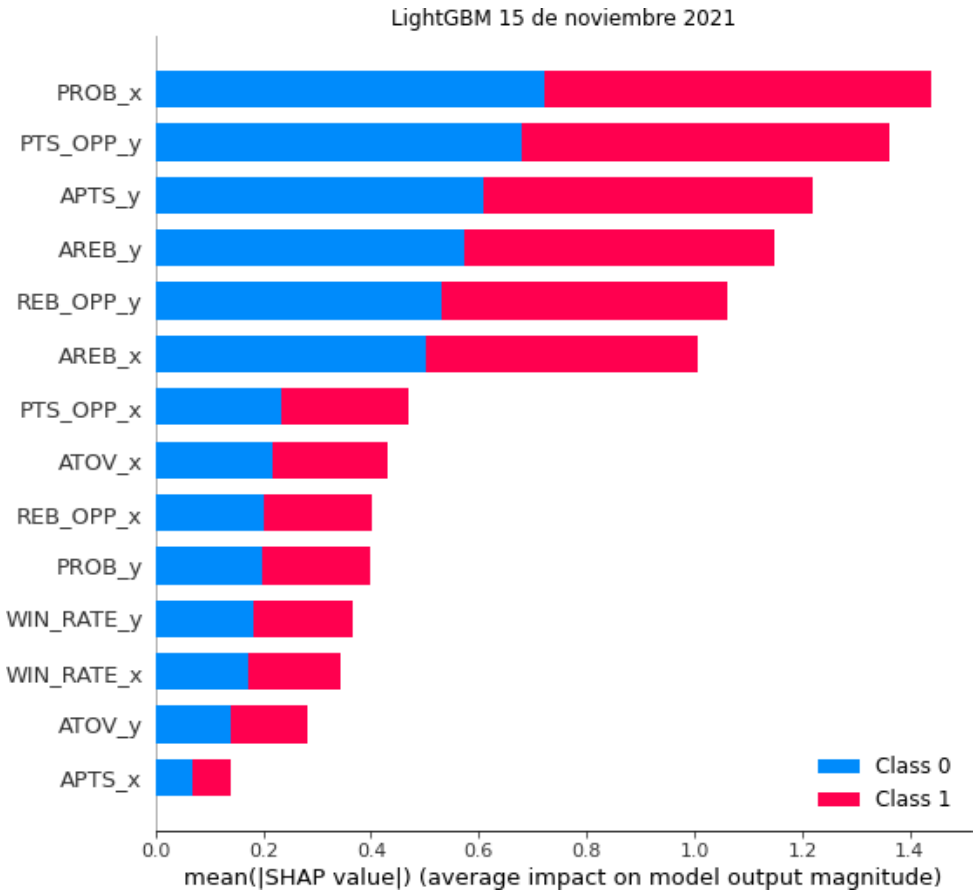


Figura 32.- Shap values lightGBM 15-11-2021

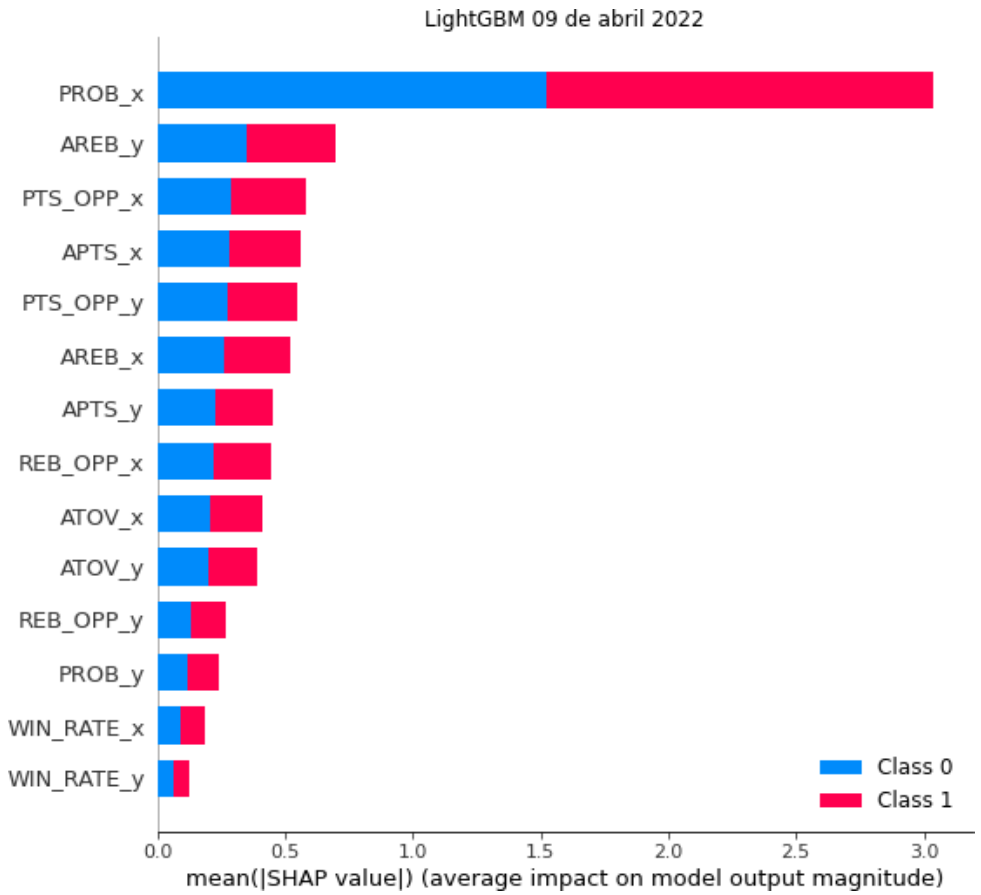


Figura 33.- Shap values lightGBM 09-03-2022

En las figuras 32 y 33 vemos los shap values del modelo lightGBM, en lo que se puede destacar que casi para el final de la temporada la decisión casi se basa por completo en lo que la probabilidad de ganar del equipo local le de la casa de apuestas.

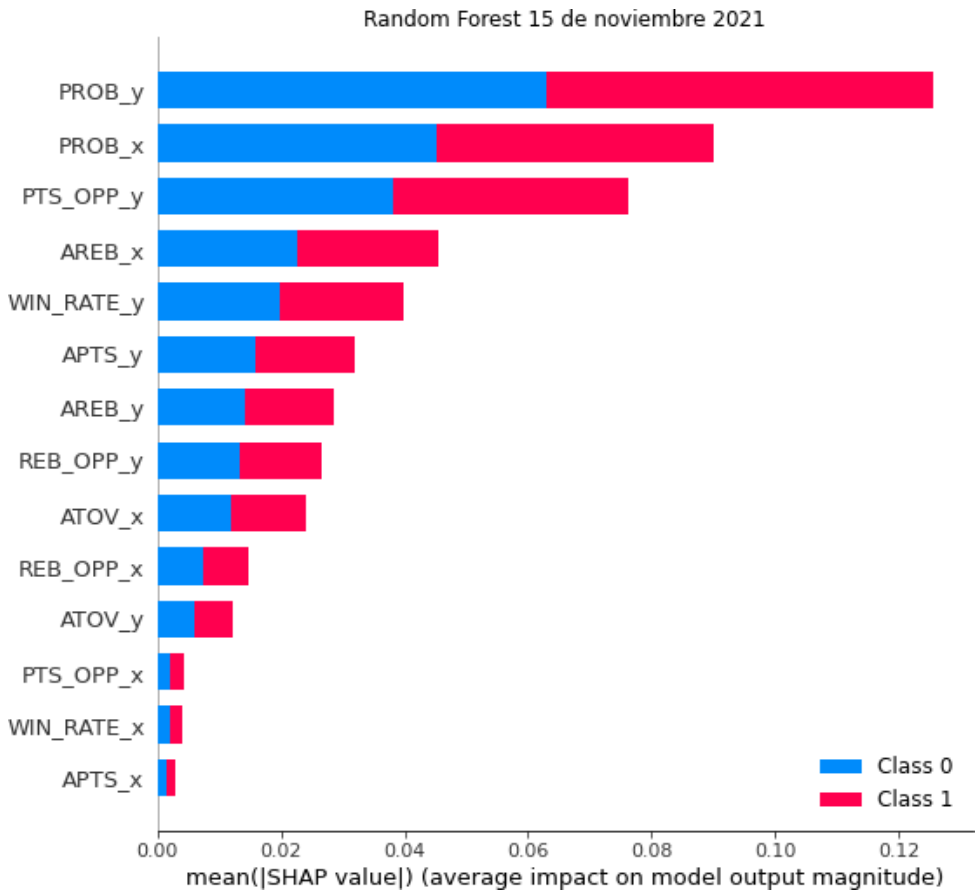


Figura 34.- Shap values random forest 15-11-2021

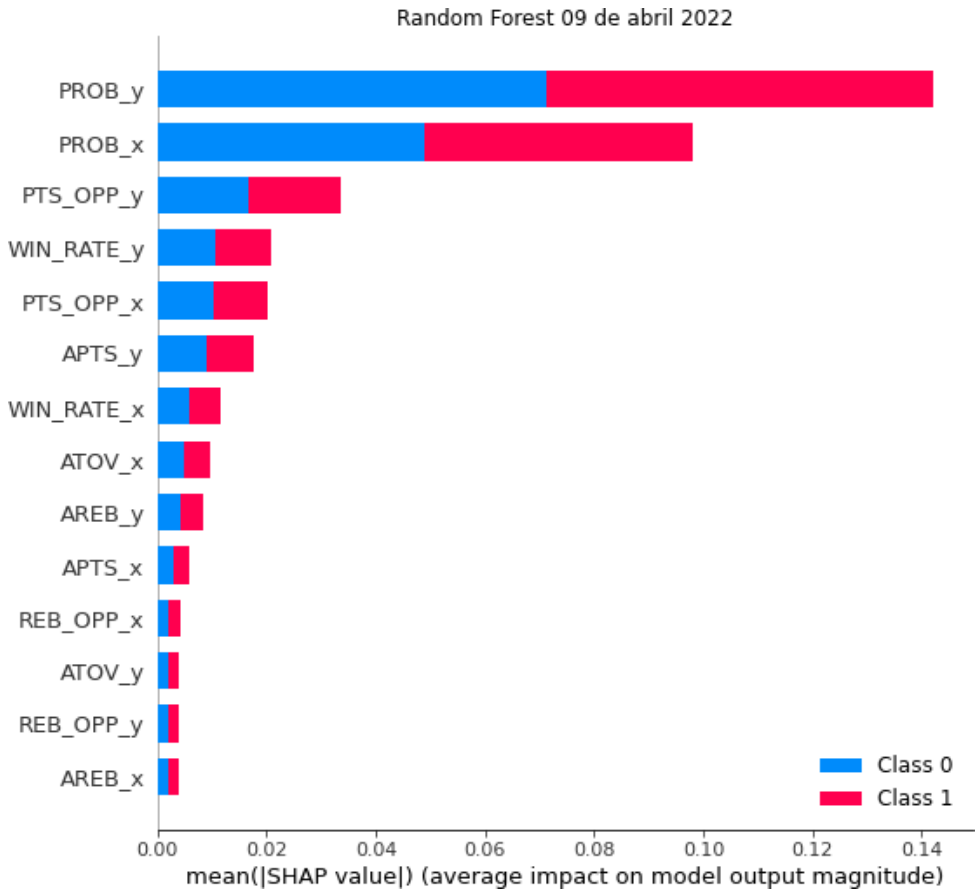


Figura 35.- Shap values random forest 09-03-2022

Por último, en las figuras 34 y 35 vemos los resultados del modelo random forest, en el cuál, al igual que en lightGBM, para el final de la temporada (aunque un poco más apoyado con otras características) el mayor peso de la decisión recae en las probabilidades de la casa de apuestas.

CAPÍTULO 5: CONCLUSIONES

A lo largo del desarrollo de esta investigación se exploró una forma de procesamiento de las estadísticas distintas a las planteadas a los trabajos relaciones, buscando una optimización en las predicciones con los distintos modelos comparados.

Los resultados obtenidos a través de la experimentación demuestran que dicha forma de procesar las estadísticas ha sido competitiva y hemos logrado alcanzar los objetivos plantados al inicio con creces, logrando por ocasiones alcanzar hasta el 75% de precisión. Demostrando que esta manera de abordar el problema es completamente viable.

Si bien, el problema se abordó principalmente de la misma forma que los trabajos mencionados, el apartado de las predicciones en tiempo real se asemeja más a la problemática real, en la cual también logramos demostrar a través de los experimentos que, si bien no es la manera definitiva, este procesamiento es completamente viable.

5.1 Discusiones.

A lo largo del desarrollo de esta investigación destacaron ciertos temas que son importantes a destacar los cuales estarán siendo mencionados en esta sección.

5.1.1 Estructuras del vector de características.

En la revisión de los antecedentes del problema se pueden apreciar las distintas formas y combinaciones que se han analizado para la resolución de este problema, algunos quedándose al margen de las estadísticas de los partidos. Pero con el trabajo realizado en esta investigación y comparándolo con otros, se puede asegurar que la introducción de variables externas, como la probabilidad de las casas de apuesta, los días descansados o los jugadores lesionados, suelen resultar en una mejor precisión en las predicciones. Esto sustentando de igual manera por el análisis realizado con los shap values.

A su vez otra cosa para resaltar en la elaboración de los vectores de características es que no siempre más es mejor. Bajo la lógica de que mientras más características mejor será el desempeño del modelo solemos pensar que obtendremos mejores predicciones, pero todo parece indicar que a veces meter un sin número de características puede resultar en un ruido en las predicciones y termine afectando al desempeño de los modelos.

Por último, es destacable que en muchas ocasiones es preferible hacer una característica combinando 2 o 3 características individuales que poner de forma cruda cada una de las características.

5.1.2 Predicciones en tiempo real.

A pesar de que las predicciones en tiempo real podrían ser de lo primero que se viene a la mente en esta problemática, la sorpresa es que no se encuentra muy investigada (al menos no en las investigaciones recurridas).

Si bien las predicciones con datos históricos pueden ser un buen indicativo de que desempeño podemos esperar de nuestros modelos, la verdad es que al intentar predecir en tiempo real suele obtener resultados bastante distintos debido a los propios problemas que esta acción conlleva, por ejemplo, el hecho de que existen cambios en las alineaciones de los equipos de una temporada a otra, puede repercutir en que el desempeño de un equipo en la temporada anterior ya no sea un indicativo de su desempeño en la temporada actual, lo que perjudicaría el desempeño de nuestro modelo.

5.1.3 Factores impredecibles.

Como en casi todos los deportes encontramos ciertos factores que nos es imposible predecir, los cuales van desde problemas personales en los jugadores, malos entendidos entre los miembros del equipo, una lesión que deje fuera a un jugador en medio del partido, etc. Todo esto significa que

debemos estar conscientes de que por más preciso que pueda ser nuestros modelos, es imposible lograr una precisión del 100%, por ende, es necesario poder proponer metas alcanzables.

5.2 Trabajos futuros.

Esta investigación puede servir de apoyo para la continuación de la problemática, a la cual se podría continuar investigando más en la importancia de las características seleccionadas, seguir explorando más modelos con sus respectivos hiperparametros y sobre todo seguir profundizando en las predicciones en tiempo real, los cuales vienen siendo de los apartados más atractivos.

A su vez, con el avance de las tecnologías y de los algoritmos en general, es probable que en algún futuro contemos que más herramientas que puedan ir mejorando los resultados establecidos en esta investigación.

Bibliografía

[1] D. de M. DEMOS, “Dan a conocer Impacto del Futbol en Economía Nacional,” La Jornada, 20-Nov-2019. [Online]. Disponible en: https://www.jornada.com.mx/2019/11/20/deportes/a11n2dep. [Rescatado: 11-Oct-2021].

[2] El Imparcial, “Los Deportes que más dinero generan,” El Imparcial, 21-Apr-2020. [Online]. Disponible en: https://www.elimparcial.es/noticia/212289/sociedad/los-deportes-que-mas-dinero-generan.html. [Rescatado: 11-Oct-2022].

[3] T. Horvat and J. Job, “The use of machine learning in sport outcome prediction: A Review,” WIREs Data Mining and Knowledge Discovery, vol. 10, no. 5, 2020.

[4] H. Li, “Analysis on the construction of sports match prediction model using neural network,” Soft Computing, vol. 24, no. 11, pp. 8343–8353, 2020.

[5] J. McCullagh, “Data mining in sport: A neural network approach.” International Journal of Sports Science and Engineering, vol. 1, no. 1, pp. 131–138, 2010.

[6] E. Zdravevski and A. Kulakov, “System for prediction of the winner in a sports game,” ICT Innovations 2009, pp. 55–63, 2010.

[7] C. Cao, “Sports Data Mining Technology Used in Basketball Outcome Prediction,” thesis, School of Computer Sciences en ARROW@TU Dublin., Dublin, 2012.

[8] B. Loeffelholz, E. Bednar, and K. W. Bauer, “Predicting NBA games using Neural Networks,” Journal of Quantitative Analysis in Sports, vol. 5, no. 1, 2009.

[9] R.A. Torres. “Prediction of NBA games base don Machine Learning Methods.” 2013

[10] J. Guzman. “Various Machine Learning Approaches to Predicting NBA Score Margins” 2016

[11] J. Lin, L. Short & V. Sundaresan “Predicting National Basketball Association Winners” 2014

[12] I. B. M. Staff, “Machine learning,” IBM, 2017. [Online]. Available: https://www.ibm.com/mx-es/analytics/machine-learning. [Accessed: 01-May-2022].

[13] Hosmer, D. W., Jovanovic, B., & Lemeshow, S. (1989). Best Subsets Logistic Regression. Biometrics, 45(4), 1265–1270. https://doi.org/10.2307/2531779

[14] C. Vega, G. Rodríguez & A. Montoya. “Metodología de evaluación del clima organizacional a través de un modelo de regresión logística para una universidad en Bogotá, Colombia”, Work environment evaluation methodology en Bogota, Colombia. 2011

[15] J. S. Armstrong, “Illusions in regression analysis,” International Journal of Forecasting, vol. 28, no. 3, pp. 689–694, 2012.

[16] Grove, W. M., & Meehl, P. E. (1996). Comparative efficiency of informal (subjective, impressionistic) and formal (mechanical, algorithmic) prediction procedures: The clinical–statistical controversy. Psychology, public policy, and law, 2(2), 293.

[17] M. Lahtela and P. (P. Kaplan, “What is boosting?,” Amazon, 1966. [Online]. Available: https://aws.amazon.com/es/what-is/boosting. [Accessed: 23-Aug-2021].

[18] J. M. Heras, “Random Forest (Bosque Aleatorio): Combinando árboles,” IArtificial.net, 18-Sep-2020. [Online]. Available: https://www.iartificial.net/random-forest-bosque-aleatorio. [Accessed: 23-Oct-2022].

[19] “XGBoost documentation,” XGBoost Documentation - xgboost 1.7.1 documentation, [Online]. Available: https://xgboost.readthedocs.io/en/stable/. [Accessed: 29-May-2021].

[20] L. GBM, “LIGHTGBM's documentation,” - LightGBM 3.3.2 documentation, 2010. [Online]. Available: https://lightgbm.readthedocs.io/. [Accessed: 29-May-2021].

[21] Catboost, CatBoost Available: https://catboost.ai/docs//. [Accessed: 16-May-2022].

[22] Shap, “Welcome to the shap documentation,” Welcome to the SHAP documentation - SHAP. [Online]. Available: https://shap.readthedocs.io/. [Accessed: 17-Apr-2022].

[23] L. S. Shapley, “17. A value for n-person games,” Contributions to the Theory of Games (AM-28), Volume II, pp. 307–318, 1953.

[24] F. Lopez, “Shap: Shapley additive explanations,” Medium, 11-Jul-2021. [Online]. Available: https://towardsdatascience.com/shap-shapley-additive-explanations-5a2a271ed9c3. [Accessed: 19-Sep-2021].

[25] NBA API. Swar, “NBA_API,” GitHub, 2017. [Online]. Available: https://github.com/swar/nba_api. [Accessed: 20-Jun-2020].

Jesús Ignacio Rodríguez Sibaja
Chihuahua/Chihuahua, México
Rodriguez.sibaja.ignacio@gmail.com
Científico de datos

Resumen

Maestro en Ingeniería en Computación con experiencia en Análisis de Datos, Machine Learning, Ciencia de Datos y habilidades de liderazgo. Buscando utilizar mi experiencia para agregar valor a la empresa y continuar desarrollándome profesionalmente.

HABILIDADES

Python | Power Bi | SQL | Microsoft Azure | Github | Análisis de Datos | Python | R | Java | Tableu Rapid Minner | Databricks | Machine Learning | Ciencia de datos | Ingles B2 | Microsoft Office Suite Liderazgo | Trabajo en equipo | Proactivo.

EDUCACIÓN

Universidad Autónoma de Chihuahua, Chihuahua/Chihuahua, México
Maestría en Ingeniería en Computación. Promedio 94.4 (Año de graduación 2023)
Honores/reconocimientos: Cuadro de Honor
Instituto Potosino de Investigación de investigación Científica y Tecnológica, San Luis potosí, México
Diplomado en Inteligencia Artificial Avanzada. (Año de Graduación 2023)
Universidad Tecnología de Puebla, Puebla/Puebla, México
Ingeniería en mecatrónica. Promedio 94 (Año de graduación 2020)
Honores/Reconocimientos: Cuadro de Honor 3 años consecutivos
Cursos Relevantes: Innovacción virtual, Github, Python, Analisís de datos Linkedin, Java

EXPERIENCIA

Automatización, Productividad y Calidad S.A de C.V., Puebla/Puebla, México
Ingeniero de proyectos (abril 2017 – febrero 2020)

- Capacitación de personal. Impartición cursos a clientes y personal de la empresa para el entendimiento y uso de los softwares desarrollados por la empresa y tecnologías nuevas.
- Coordinación de proyectos eléctricos. Se lograron realizar los análisis e implementar las correcciones necesarias en el tiempo establecido a la empresa. A su vez se logró coordinar exitosamente trabajos de campo de manera remota.

Lobbyroom S.A de C.V., Xalapa/Veracruz, México
Data Scientist (diciembre 2022 - actualmente)

- Procesamiento y análisis de datos en bases de datos de empresas privadas para la implementación de modelos de Machine Learning. Se logró producir predicciones de consumo de energía y ventas para empresas privadas, se implementaron dashboards para reportar dichos resultados.

EXPERIENCIA ADICIONAL

Análisis de datos deportivos – Proyecto personal, Chihuahua/Chihuahua, Mexíco (2021-2022)

- Extracción de datos, procesamiento, implementación de modelos de Machine Learning, predicción de ganadores, automatización de los procesos.

LOGROS

Universidad Autónoma de Chihuahua, Chihuahua/Chihuahua, México, 2022

- **Primer lugar en Hackathon Quantum Apps.** Por la implementación y desarrollo de una página web para el análisis y detección de anomalías en gases en tiempo real