Predicción de resultados en encuentros profesionales de tenis de campo

Daniel Vanegas Gómez

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Sebastián Rodríguez Colina, Magíster (MSc) en Ingeniería

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2023

Cita	(Soto-Valero, 2017)
Referencia **Estilo APA 7 (2020)**	Soto-Valero, C. (07 de 2017). *A Gaussian mixture clustering model for characterizing football players using. Obtenido de RICYDE. Revista Internacional de Ciencias del Deporte: https://bit.ly/3qijN3H* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte IV.

Tabla de contenido

Contenido

Lista de tablas

Lista de figuras

Resumen

Estimar el ganador de un partido en tenis de campo es una tarea que implica entender las condiciones bajo las cuales se dan cada uno de los encuentros y comprender las diferentes variables que pueden beneficiar o afectar a cada individuo. Variables como el entorno, el oponente y resultados en encuentros previos son indicadores que pueden ser analizados para determinar la probabilidad que se dé X ó Y resultado.

Contar con datos acerca de los diferentes partidos jugados además de las estadísticas que se dieron en los mismos, permitiría comprender las razones y/o circunstancias que favorecen o afectan los resultados de cualquier jugador. Por lo tanto, en la plataforma de Kaggle, se obtuvo un dataset que contiene el detalle de todos los partidos oficiales de la ATP (Asociación de tenistas profesionales) desde 1968 y, a partir de 1991, contiene además las estadísticas de los encuentros hasta el año 2019.

En el presente trabajo se presenta una propuesta mediante algoritmos de clasificación para determinar quién sería el ganador en un encuentro profesional de tenis de campo bajo diferentes condiciones.

Abstract

Estimating the winner of a tennis match is a task that involves understanding the conditions under which each match takes place and comprehending the different variables that can benefit or affect each individual. Variables such as the environment, the opponent, and previous match results are indicators that can be analyzed to determine the probability of achieving either an X or Y outcome.

Having data about the different matches played, along with the statistics that occurred during those matches, would allow us to understand the reasons and circumstances that favor or affect the results of any player. Therefore, on the Kaggle platform, a dataset was obtained that contains the details of all official ATP (Association of Tennis Professionals) matches since 1968, and starting from 1991, it also includes match statistics up until the year 2019.

In this present work, a proposal is presented using classification algorithms to determine who would be the winner in a professional tennis match under different conditions.

Introducción

La predicción de resultados en los encuentros de tenis de campo ha sido un tema de interés en el mundo del deporte y la estadística durante muchos años (Barnett, 2005). En este trabajo de monografía, se aborda la predicción de resultados en los partidos de la ATP, utilizando un dataset que contiene estadísticas de todos los partidos jugados desde 1991.

El objetivo de este trabajo es desarrollar un modelo de clasificación para predecir quién ganó o perdió un partido entre los jugadores clasificados dentro de los 100 primeros del mundo. Para ello, se utiliza un modelo de clasificación, que es un estadístico ampliamente utilizado para la predicción de resultados binarios (Ricardo Gil Rubio, 2022).

La idea principal detrás del modelo es analizar el conjunto de variables que podrían influir en el resultado de un partido, incluyendo la clasificación de los jugadores, su historial de partidos previos, su rendimiento en diferentes superficies y otros factores relevantes. Luego, se utiliza esta información para entrenar el modelo y realizar predicciones precisas sobre el resultado de los partidos.

El enfoque en este trabajo se centra en los partidos de la ATP, ya que el tenis de campó es uno de los deportes más populares y seguidos en todo el mundo (Descubre cuáles son los deportes más populares del mundo, 2022). Además, al utilizar un dataset con un gran número de partidos jugados, se puede realizar un análisis exhaustivo de los factores que influyen en los resultados de los partidos y evaluar la eficacia del modelo propuesto.

En resumen, este trabajo de monografía tiene como objetivo proporcionar una evaluación detallada del modelo de regresión logística para la predicción de resultados en los encuentros profesionales de tenis de campo y demostrar su eficacia en la predicción de quién ganó o perdió un partido

Planteamiento del problema

El objetivo principal de este trabajo es desarrollar un modelo de regresión logística para predecir los resultados en encuentros profesionales de tenis de campo. El tenis es un deporte ampliamente popular y altamente competitivo, y tener la capacidad de predecir los resultados de los partidos puede ser de gran interés para los fanáticos, los jugadores, los entrenadores y los analistas.

Para lograr esto, se utilizará un conjunto de datos exhaustivo que contiene estadísticas de partidos de tenis desde el año 1991. Estas estadísticas incluyen información relevante sobre los jugadores, como clasificaciones, edad, país de origen, así como datos específicos de los partidos, como el número de sets, juegos ganados, errores no forzados, entre otros. Estos datos servirán como datos de entrenamiento para el modelo de regresión logística.

Además, se identificarán aquellos patrones o variables que afectan en mayor medida el que un jugador gane o pierda un partido y cómo se correlaciona con las demás variables, especialmente las categóricas como el tipo de superficie, el nivel del torneo y la mano hábil del jugador.

Justificación

La predicción de resultados en tenis de campo es un tema de gran relevancia debido a sus diversas aplicaciones prácticas en distintos ámbitos. Uno de los campos en los que esta predicción resulta invaluable es el análisis deportivo. Un modelo de clasificación preciso y confiable puede proporcionar a los entrenadores una herramienta adicional para evaluar las fortalezas y debilidades de los jugadores, permitiéndoles tomar decisiones estratégicas más fundamentadas durante los entrenamientos y competiciones. Esto a su vez puede mejorar el rendimiento y los resultados del equipo o de los jugadores.

Por otro lado, las predicciones en tenis de campo también tienen un impacto en el ámbito de las apuestas deportivas. Tanto los apostadores profesionales como los aficionados buscan información precisa y confiable para tomar decisiones informadas al realizar sus apuestas. Un modelo de clasificación preciso podría proporcionarles una ventaja al predecir los resultados de los partidos y ayudarles a maximizar sus ganancias o minimizar sus pérdidas. Esto resulta especialmente relevante en un deporte tan impredecible como el tenis, donde las sorpresas y los cambios inesperados son comunes.

Además de su impacto en el análisis deportivo y las apuestas, este tipo de investigación también contribuye al campo del aprendizaje automático aplicado al deporte. La aplicación de técnicas de aprendizaje automático en el tenis de campo ofrece la oportunidad de descubrir patrones y tendencias ocultas en los datos, lo que puede conducir a un mejor entendimiento del juego y a la mejora de las estrategias y tácticas utilizadas por los jugadores y entrenadores. Los avances realizados en este campo no solo benefician al tenis, sino que también pueden sentar las bases para la aplicación de metodologías similares en otras disciplinas deportivas, ampliando así su impacto en el mundo del deporte en general.

En resumen, la predicción de resultados en tenis de campo tiene aplicaciones prácticas en el análisis deportivo, las apuestas y la toma de decisiones estratégicas. Un modelo de clasificación preciso puede proporcionar información valiosa a entrenadores, apostadores y aficionados, permitiéndoles tomar decisiones fundamentadas y mejorar su desempeño. Además, este tipo de investigación contribuye al campo del aprendizaje automático aplicado al deporte, ofreciendo metodologías que pueden ser utilizadas en otras disciplinas deportivas.

Objetivos

Objetivo general

Desarrollar un sistema de predicción precisa para determinar el ganador en encuentros profesionales de tenis de campo, utilizando como base las estadísticas históricas de los partidos de la Asociación de Tenistas Profesionales (ATP) desde 1991.

Objetivos específicos

- Recopilar y analizar los datos históricos de partidos oficiales de la ATP desde 1991, incluyendo estadísticas detalladas de los encuentros, como porcentaje de primeros servicios, errores no forzados, puntos ganados, entre otros.
- Preprocesar y limpiar los datos recopilados, asegurando la calidad y consistencia de la información, y realizando la transformación y normalización necesarias para prepararlos para el entrenamiento del modelo.
- Implementar un modelo de clasificación utilizando el algoritmo de regresión logística. Dividir los datos en conjuntos de entrenamiento y prueba, y ajustar los parámetros del modelo para maximizar su rendimiento y precisión en la predicción de resultados.
- Evaluar el modelo utilizando métricas de desempeño, como la precisión, el porcentaje de acierto y la matriz de confusión.
- Analizar las características y variables más influyentes en la predicción de resultados. Evaluar la importancia de cada variable en el modelo y su contribución al rendimiento predictivo.

Marco teórico

La aplicación de modelos estadísticos y predictivos en el ámbito deportivo es una técnica que ha venido en crecimiento durante las últimas décadas. Tener la capacidad de predecir o identificar los factores que favorecen una lesión o que benefician la obtención de un resultado ha sido un plus muy importante que cada vez repercute en la mejora de los logros obtenidos por los diferentes deportistas o equipos.

Actualmente la comunidad científica internacional ha mostrado un interés mayor en aplicar las ciencias de la computación en el deporte con el objetivo de combinar aspectos teóricos y prácticos que involucren las técnicas utilizadas en el área de la computación para generar mejoras o outputs reveladores acerca de la teoría del deporte y cómo generar mejoras en la práctica del deporte moderno (Link, 2009). Estudiar y relacionar la teoría y la práctica en el deporte genera muchos beneficios en el sentido de que las inteligencias artificiales tienen la capacidad de identificar hasta el más mínimo error en un movimiento o jugada que puede ser imperceptible para el ojo humano, lo cual permite la aplicación de entrenamientos o correctivos en pro de generar mejoras en los resultados y/o técnica del deportista involucrado.

(Soto-Valero, 2017) hace referencia a que la estadística tradicionalmente se ha utilizado para resolver los problemas que se refieren al manejo de los datos del deporte, principalmente, a la conclusiones o estadísticos importantes para tomar decisiones respecto de los resultados del estudio. Este tipo de técnicas es especialmente útil cuando se quiere determinar el método de entrenamiento óptimo entre varias opciones lo cual permite mejorar en el tiempo el rendimiento y los resultados de los deportistas. No obstante, no es un método que permita identificar los patrones que maximizan las posibilidades de victoria, en el caso de un partido de tenis de campo, únicamente selecciona el mejor plan deportivo entre varias opciones.

Por lo tanto, y teniendo en cuenta que para poder identificar los factores que influyen en ganar un juego se deben analizar cientos de miles de datos de partidos para identificar patrones o tendencias que favorecen salir victorioso de un encuentro, es importante contar con herramientas que sean capaz de procesar tal magnitud de información y aprender de ella. Por lo tanto, y según lo expuesto por (Mitchell, 1997), el aprendizaje automático es un campo que busca mejorar su funcionamiento con base en la experiencia, por lo cual sería una herramienta ideal para modelar

problemas de clasificación en donde se deben identificar características dentro de un mar de datos muy elevado como es el caso de predecir resultados en tenis de campo.

Una de las primeras aplicaciones del aprendizaje automático se dio en el contexto deportivo donde con métodos de regresión y clasificación se estimaron predicciones de resultados en competencias de dicha índole (Robert P. Schumaker, 2010). El objetivo de tales predicciones era conseguir información que fuera determinante para obtener una ventaja ya fuera competitiva o financiera respecto de los demás. Hoy en día las aplicaciones de apuestas deportivas hacen mucho uso de este tipo de métodos con el fin de estimar las probabilidades de que se de cierto suceso y a partir de modelos estadísticos determinar las cuotas a pagar.

Estimar los ganadores en el deporte depende de muchos factores y se debe tener en cuenta si son deportes individuales o en conjunto ya que el primero contempla más variables y factores lo cual lo hace más complejo. Para el caso de los deportes individuales, los análisis se basan en el desempeño de cada individuo y las estadísticas históricas asociadas al mismo, variables que se han utilizado históricamente en la selección de talentos del deporte, prevención de lesiones y evaluación de rendimiento (Subramanian Rama Iyer, 2009).

Metodología

- **Enfoque de investigación**: En este trabajo se utilizará un enfoque cuantitativo, ya que se busca desarrollar un modelo de clasificación y utilizar estadísticas históricas para predecir los resultados de los partidos de tenis de campo.
- **Recopilación de datos**: Se recopilarán datos históricos de partidos de tenis de campo de fuentes confiables, como kaggle. Se obtendrán datos relevantes, como estadísticas de juego, resultados finales, características de los jugadores y detalles de los partidos.
- **Preprocesamiento de datos**: Se realizará una limpieza de los datos para eliminar registros duplicados, valores atípicos y datos faltantes. Se aplicarán técnicas de transformación de variables, como el escalado y la normalización, para asegurar la calidad y la consistencia de los datos.
- **Diseño y entrenamiento del modelo de clasificación: S**e implementará un modelo de clasificación utilizando el algoritmo de regresión logística. Se ajustarán los parámetros del modelo mediante la búsqueda de hiperparámetros para maximizar su rendimiento y precisión.
- **Evaluación del modelo:** Se evaluará el modelo utilizando métricas de desempeño, como la accuracy, la curva ROC y la matriz de confusión.
- **Interpretación de resultados**: Se analizarán las características y variables más influyentes en la predicción de resultados y se evaluará la importancia de cada variable en el modelo y su contribución al rendimiento predictivo.

Resultados

Análisis Exploratorio de los Datos

El conjunto de datos utilizado es de libre acceso en la plataforma kaggle y se puede acceder al mismo aceptando los términos y condiciones a través del siguiente enlace: https://bit.ly/3N2WyCt. Además, el análisis exploratorio de los datos así como el modelado y despliegue se realiza en Jupyter Notebook el cual puede ser consultado en el siguiente repositorio de github: https://bit.ly/3NmB192.

Se tienen un total de 169.690 filas y 49 columnas, donde cada una de las filas corresponde a un partido jugado.

Descripción de variables categóricas

En total existen 10 variables categóricas relacionadas con la nacionalidad, superficie, mano hábil del jugador, entre otras. Este tipo de características son importantes para determinar el ganador de un partido ya que hay condiciones que favorecen a ciertos jugadores sobre otros. A continuación, se detallan algunos ejemplos.

- Jugar partidos en alturas superiores a los 2.000 metros sobre el nivel del mar puede beneficiar mayormente a aquellos jugadores acostumbrados a entrenar bajo estas condiciones ya que sus cuerpos están adaptados a la menor cantidad de oxígeno presente en el aire lo que hace que su desempeño físico no se afecte en comparación con jugadores que provienen de zonas cálidas.
- Puesto que la gran mayoría de tenistas son diestros, se está menos habituado a los efectos, velocidades y golpes provenientes de jugadores zurdos por lo que tener un buen nivel en tenis de campo y ser zurdo supone una pequeña ventaja competitiva frente a los demás.
- Existen superficies que favorecen a ciertos estilos de juego. Por ejemplo, la hierba y el cemento favorecen a aquellos que juegan de manera agresiva y veloz y buscan ganar los puntos con pocos intercambios de golpes ya que la pelota no bota mucho y cuando pega en el suelo se acelera. Por el contrario, el polvo de ladrillo favorece a los jugadores que son “pasabolas” ya que la bola rebota mucho y cuando pega en el suelo se frena,

otorgando mayor tiempo de reacción.

Entry

Se refiere a la manera en que el jugador hizo parte del cuadro principal del torneo.

WC - Wildcard

Q - Qualifier

LL - Lucky loser

PR - Protected ranking

SE - Special Exempt

ALT - Alternate player

NaN - Entered directly

Hand

Indica la mano hábil del tenista.

R- Right

L - Left

U - Unknown

Id

Número de identificación único para cada jugador contenida en el dataset.

IOC

Consta de tres letras que hacen alusión al país de origen del jugador.

Name

Nombre completo del jugador

Round

Ronda del torneo en la cual se disputa el partido.

R128- Ronda de 128

R64 - Ronda de 64

R32 - Ronda de 32

R16- Ronda de 16

QF - Cuartos de Final

SF - Semifinal

F - Final

Surface

Superficie en el cual el partido se jugó.

Grass - Hierba

Hard - Pista Dura

Clay - Polvo de ladrillo

Carpet - Carpeta

Tourney_id

Número de identificación único para cada torneo.

Tourney_level

Categoría del torneo.

G - Grand Slam

M - Masters 1000s

A- Other tour-level events

C - Challengers

S - Satellites/ITFs

F - Tour finals and other season-ending events

Tourney_name

Nombre del torneo

Descripción de variables numéricas

En total existen 19 variables numéricas que en su mayoría corresponden a las estadísticas del partido jugado mientras que las demás hacen referencia a generalidades del partido.

best_of

El máximo número de sets jugados (3 ó 5)

draw_size

Tamaño del cuadro principal

1_stin

Puntos jugados con el primer servicio

1stWon

Puntos ganados con el primer servicio

2ndWon

Puntos ganados con el segundo servicio

SvGms

Numero de games jugados al servicio.

ace

Número de aces en el partido

bpFaced

Puntos de quiebre enfrentados

bpSaved

Puntos de quiebre salvados

df

Dobles faltas

svpt

Porcentaje de servicio

age

Edad del jugador

ht

Altura del jugador

rank

Ranking del jugador

seed

La siembra del jugador en el torneo

match num

Número de partido en un determinado torneo

minutes

Duración del partido en minutos

score

Resultado final del partido

tourney_date

Fecha de inicio del torneo

***Nota**: Una variable puede aparecer dos veces en el dataset con la diferencia que una presenta las estadísticas del ganador mientras que la segunda es del perdedor, por ejemplo:

w_1stIn → Hace referencia a los puntos jugados con el primer servicio de quien ganó el partido (winner)

l_1stIn→ Hace referencia a los puntos jugados con el primer servicio de quien perdió el partido (loser)

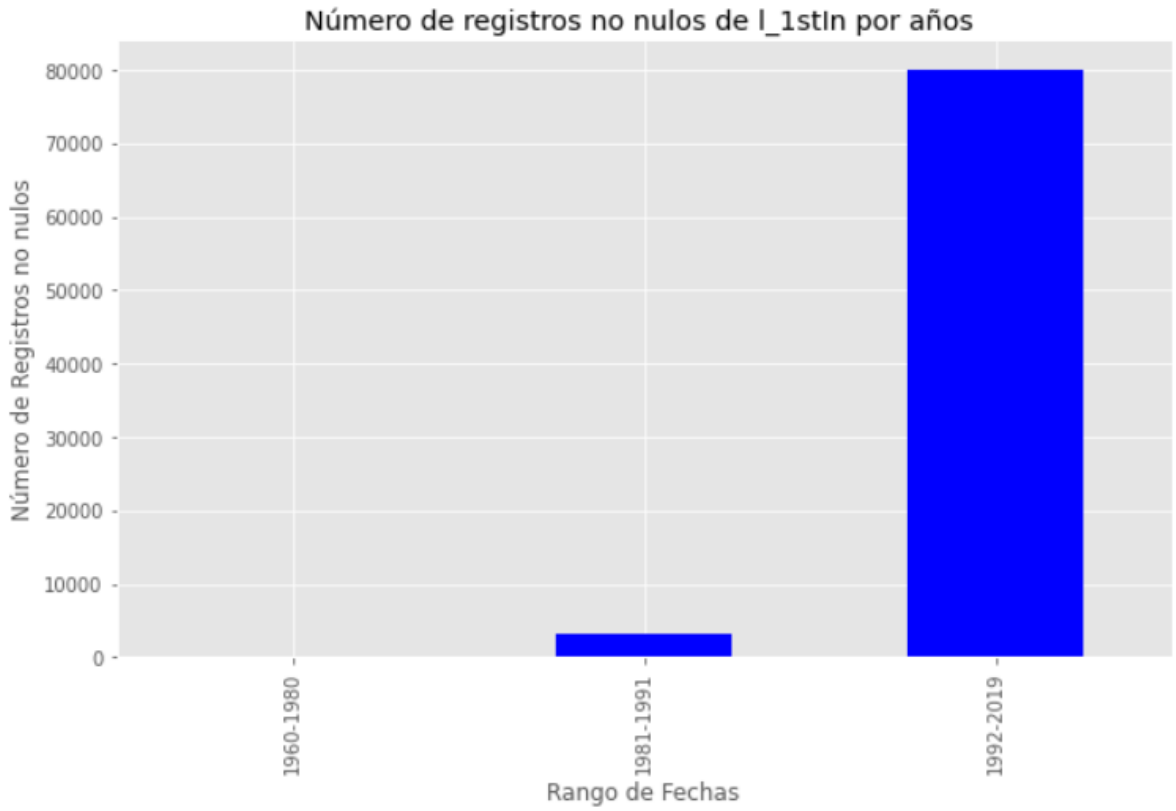
Limpieza de datos y preparación

El código utilizado para realizar el análisis exploratorio de datos (EDA) se encuentra disponible en el siguiente repositorio de github : https://bit.ly/3Nseh7y

Valores nulos o faltantes

Al realizar la validación en el dataset de los registros nulos se encuentra que, en todas las variables menos en 13, existen registros nulos o vacíos por lo que se profundiza en estos para verificar las razones de ello.

Figura 1 *Registros no nulos por años*



Para la variable l_1stIn que corresponde a los puntos jugados con el primer servicio del jugador perdedor se encuentra que no existen casi registros antes del año 1992 lo cual explica la cantidad elevada de registros nulos o vacíos que existen en la base de datos. Por lo tanto, y teniendo en cuenta que las estadísticas de los partidos son variables importantes para determinar quién gana o no un partido, se toma la decisión de sólo tomar los registros a partir del año 1992 para garantizar así que los datos que se vayan a pasar por el modelo tengan consistencia y puedan ser comparados frente a los demás.

Asimismo, teniendo en cuenta que en el dataset existen datos de todos los jugadores que han estado en el circuito profesional de tenis independiente del ranking y que existe una marcada diferencia en cuanto a la calidad de los jugadores que están en los primeros 100 del mundo frente a los demás, se decide tener únicamente en cuenta los partidos que se hayan dado entre jugadores ubicados en dicho top. De igual forma, teniendo presente que estos jugadores se entrenan para dar sus mejores resultados en los principales torneos del tenis como lo son los de categoría Grand Slam

(G), Masters 1000 (M) y otros eventos del circuito de tenis profesional (A), se decide tener en cuenta sólo los resultados obtenidos en dichos torneos, de esta manera se garantiza que los registros que se tienen en cuenta son equiparables en cuanto a la calidad de los jugadores que indicaría su ubicación en el ranking además de tomar en cuenta los torneos más importantes del circuito.

Por lo tanto, después de excluir los registros previos al año 1992 y teniendo en cuenta partidos entre jugadores del top 100 en las 3 categorías de torneo más importantes, se obtiene un nuevo dataset con 74.918 registros, donde no se evidencian registros duplicados y se procede a realizar nuevamente la validación de datos nulos o faltantes.

Tabla 1 *Registros nulos por cada variable del dataset*

Variable	Registros nulos
Draw_size	74.918
Loser_entry	59.889
Loser_seed	55.740
Winner_entry	65.694
Winner_seed	40.782

Se encuentra que la mayor cantidad de registros nulos se dan en 3 variables: “draw_size”, “entry” y “seed”. Las primeras dos no corresponden a estadísticas propias del desempeño de los jugadores en el partido y se considera que son datos que no aportarían valor al modelo de clasificación, puesto que el “draw_size” sólo indica con cuantos jugadores inició el torneo y “entry” hace referencia a si el jugador se inscribió al torneo de manera voluntaria o si recibió una invitación especial. Por otro lado, la variable “seed” que se refiere a la siembra que tenía el jugador en el torneo, es una característica que podría ayudar a predecir el ganador de un partido teniendo en cuenta que normalmente en un encuentro gana el jugador que tiene mejor siembra, no obstante, se evidencia que casi en el 75% de los registros no se cuenta con los datos de la siembra del torneo, lo cual haría inviable considerar la misma como parte del modelo de clasificación por falta de información. Sin embargo, la siembra de los jugadores en cualquier torneo se define según la

posición del ranking, por lo tanto, la variable “rank” contiene implícitamente la información de la variable “seed”.

Por lo anterior, se toma la decisión de excluir las variables del dataset. Se identifica adicionalmente que existen 1176 registros que no tienen ningún dato asociado a las estadísticas del partido, únicamente cuenta con la información de los jugadores y del torneo por lo que se decide igualmente excluirlos de la base.

Asimismo, en las variables “ht” y “minutes”, se evidencian algunos valores nulos adicionales que se consideran que no son representativos ya que contemplan menos del 3% del total de los datos. Además, realizar una imputación de datos en variables como “ht” afectaría la calidad de la información al modificar su valor real ya que la altura de los jugadores no es un dato que cambie con el tiempo, es por ello que se decide igualmente eliminar la información de la base de datos.

Tabla 2 *Registros nulos por variable después de tratamiento de datos*

Variable	Registros nulos
Loser_ht	74.918
Winner_ht	59.889
Minutes	55.740

Análisis de Correlación

Se determina la correlación que existe entre las variables presentes en el set de datos según la siguiente figura.

Figura 2 *Correlación entre variables*

	best_of	l_1stIn	l_1stWon	l_2ndWon	l_SvGms	l_ace	l_bpFaced	l_bpSaved	l_df	l_svpt
best_of	1.000000	0.493422	0.456740	0.347808	0.540738	0.201380	0.353632	0.259291	0.198558	0.528119
l_1stIn	0.493422	1.000000	0.949216	0.561328	0.875278	0.395198	0.458985	0.489329	0.212537	0.926174
l_1stWon	0.456740	0.949216	1.000000	0.606614	0.897551	0.545975	0.309717	0.399345	0.267395	0.914417
l_2ndWon	0.347808	0.561328	0.606614	1.000000	0.767702	0.421901	0.283316	0.358942	0.329401	0.783754
l_SvGms	0.540738	0.875278	0.897551	0.767702	1.000000	0.492069	0.364028	0.353475	0.317764	0.934847
l_ace	0.201380	0.395198	0.545975	0.421901	0.492069	1.000000	-0.017515	0.096414	0.251016	0.457096
l_bpFaced	0.353632	0.458985	0.309717	0.283316	0.364028	-0.017515	1.000000	0.921744	0.248180	0.519459
l_bpSaved	0.259291	0.489329	0.399345	0.358942	0.353475	0.096414	0.921744	1.000000	0.226784	0.541147
l_df	0.198558	0.212537	0.267395	0.329401	0.317764	0.251016	0.248180	0.226784	1.000000	0.367304
l_svpt	0.528119	0.926174	0.914417	0.783754	0.934847	0.457096	0.519459	0.541147	0.367304	1.000000

Nota: Muestra de correlación entre las primeras 10 variables de dataset.

Luego de realizar el análisis de correlación se evidencian que existen variables que están relacionadas con otras en más de un 80%, es decir, que si una de ellas aumenta o disminuye en cierta medida, la otra lo hará de la misma manera, al menos, en un 80% del cambio de la variación de la primera variable.

En la gráfica anterior se observa, para el jugador perdedor, que la correlación entre los primeros servicios logrados (l_1st_in) y los primeros servicios ganados (l_1st_won) es de 94,92%, lo cual se puede explicar en el hecho de que a mayor cantidad de primeros servicios logrados en un partido, mayor será la posibilidad de ganar puntos con el primer servicio, máxime, cuando normalmente el primer servicio es el que se realiza con mayor velocidad y fuerza y suele ser más complicado de responder para el oponente.

Teniendo claro lo anterior, se toma la decisión de eliminar aquellas variables que presenten una correlación superior al 80% ya que la variable que permanece en el dataset explicaría en un porcentaje muy alto la que fue excluida; esto es útil porque disminuye complejidad al modelo de predicción que se realizará más adelante. Además, y de cara al objetivo del presente trabajo, que es determinar quién sería el ganador de un partido de tenis, se encuentra que existen en el dataset variables que no aportan valor para tal fin, como lo son: “loser_ioc”, “tourney_date”, “loser_id”, “winner_id”, “match_num”, “winner_rank_points”, “loser_rank_points”, “best_of”, “tourney_id”, “tourney_name”, “winner_ioc”, “round”, por tanto, se toma la decisión de excluir las variables del mismo.

Figura 3 *Correlación entre variables después de excluir aquellas superiores al 80%*

	l_2ndWon	l_ace	l_bpFaced	l_df	loser_age	loser_ht	loser_rank	w_2ndWon	w_ace	w_bpFaced
l_2ndWon	1.000000	0.421901	0.283316	0.329401	-0.002515	0.040532	-0.089204	0.621661	0.335772	0.388628
l_ace	0.421901	1.000000	-0.017515	0.251016	0.035632	0.358938	-0.090208	0.393138	0.309574	0.112809
l_bpFaced	0.283316	-0.017515	1.000000	0.248180	-0.025340	-0.097727	-0.020312	0.195777	-0.053145	0.370939
l_df	0.329401	0.251016	0.248180	1.000000	-0.036373	0.099101	-0.007692	0.256323	0.111835	0.197196
loser_age	-0.002515	0.035632	-0.025340	-0.036373	1.000000	-0.009886	-0.067216	-0.004475	0.080159	-0.028172
loser_ht	0.040532	0.358938	-0.097727	0.099101	-0.009886	1.000000	-0.050299	0.054586	0.021877	-0.061213
loser_rank	-0.089204	-0.090208	-0.020312	-0.007692	-0.067216	-0.050299	1.000000	-0.082645	-0.064609	-0.059901
w_2ndWon	0.621661	0.393138	0.195777	0.256323	-0.004475	0.054586	-0.082645	1.000000	0.308496	0.431749
w_ace	0.335772	0.309574	-0.053145	0.111835	0.080159	0.021877	-0.064609	0.308496	1.000000	0.012674
w_bpFaced	0.388628	0.112809	0.370939	0.197196	-0.028172	-0.061213	-0.059901	0.431749	0.012674	1.000000
w_df	0.351613	0.186734	0.167565	0.236422	-0.013066	-0.002758	-0.009459	0.397162	0.246763	0.374526
winner_age	0.019902	0.071881	-0.027554	-0.009050	0.135663	0.046849	-0.017792	0.000308	0.040052	-0.018139
winner_ht	0.055192	0.036614	-0.121295	-0.018662	0.053576	0.029610	-0.040816	0.019061	0.403653	-0.101347
winner_rank	-0.021867	-0.026735	-0.043497	0.036443	-0.025242	-0.033292	0.133884	-0.029400	-0.076971	0.041786

Nota: Muestra de correlación entre las primeras 10 variables de dataset.

El dataset final contiene 66.715 filas y 21 columnas, de las que 14 son numéricas mientras que las restantes son categóricas.

Variable a predecir

La base de datos no cuenta con una única variable que indique quien fue el ganador del partido, específicamente, hay dos columnas donde se especifica el nombre de quién ganó y quién perdió, es decir, existen las variables “loser_name” y “winner_name”. Esto es un inconveniente de cara al modelo de predicción que se realizará ya que para el mismo se necesita una única variable a predecir pero como se explicó, existen dos.

Con el fin de solucionar el tema de la variable a predecir para la creación del modelo, se decide tomar como referencia la columna “loser name”, que será sobre la cual se determinará si se ganó o se perdió el partido. Además, se creará una nueva variable “Resultado” que podrá tomar dos valores, “G” ó “P”, haciendo referencia a los dos posibles resultados que se pueden obtener. Finalmente, se duplicarán los registros del dataset y se invertirán las estadísticas y nombres de los jugadores para así contar con partidos ganados y perdidos para modelar. En las siguientes imágenes se explica de mejor manera la solución planteada.

Figura 4 *Representación dataset original*

loser_name	Estadísticas del partido del loser		Winer_name	Estadísticas del partido del winner	
Pepito Perez	5	10	Juan Cortez	10	15
Luis Ruíz	4	10	Román Torres	8	20

Nota: Representación de la base de datos actual

La anterior es una representación corta de la base de datos que se tiene en la cual existen dos columnas destinadas a indicar el nombre del jugador que ganó y el que perdió el partido. En el dataset original existen varias variables que representan las estadísticas del partido para cada jugador, para el presente ejemplo se resumen en dos.

Figura 5 *Representación del set de datos adicionando columna de resultado*

loser_name	Estadísticas del partido del loser		Winer_name	Estadísticas del partido del winner		Resultado
Pepito Perez	5	10	Juan Cortez	10	15	P
Luis Ruíz	4	10	Román Torres	8	20	P

Se crea una nueva variable llamada “Resultado” la cual podrá tomar únicamente dos valores, “G” ó “P”, y la cual toma como referencia en este caso a la variable “loser_name”, por lo que en todos los casos será “P”.

Figura 6 *Duplicación y reacomodamiento de datos*

loser_name	Estadísticas del partido del loser		Winer_name	Estadísticas del partido del winner		Resultado
Pepito Perez	5	10	Juan Cortez	10	15	P
Luis Ruíz	4	10	Román Torres	8	20	P
Juan Cortez	10	15	Pepito Perez	5	10	G
Román Torres	8	20	Luis Ruíz	4	10	G

En el tercer paso, se duplican todos los registros de la base de datos, pero con la diferencia que las variables de los nombres de los jugadores y sus estadísticas se invierten, es decir, se replican en el lado opuesto al que estaban. En el ejemplo, se evidencia que la base de datos se duplicó de 2 a 4 filas y las descripciones de los jugadores junto con sus estadísticas se replicaron en el lado opuesto, para el caso que está resaltado en negro, pasaron de “winner_name” a “loser_name” y

como esta última variable es la que se toma como base para la columna “Resultado”, se agregan a esos nuevos registros el valor de “G”.

Figura 7 *Representación de ejemplo después del tratamiento a los datos*

Estadísticas del partido del loser		Estadísticas del partido del winner		Resultado
5	10	10	15	P
4	10	8	20	P
10	15	5	10	G
8	20	4	10	G

En el cuarto y último paso, se eliminan las variables de los nombres de los jugadores ya que las características de cada jugador ya están en el dataset y el objetivo es que el modelo funcione para otros duelos que pasen a futuro, incluso con jugadores que hoy no están en la base de datos.

Con los cambios que se realizaron en la base de datos ya se garantiza que se tiene una variable a predecir (Resultado) y el modelo se puede entrenar con las diferentes estadísticas asociadas a cada uno de los jugadores en cada partido. También, y debido a la transformación realizada, en el nuevo dataset en total existe la misma cantidad de partidos ganados y perdidos.

Figura 8 *Frecuencia de posibles resultados*

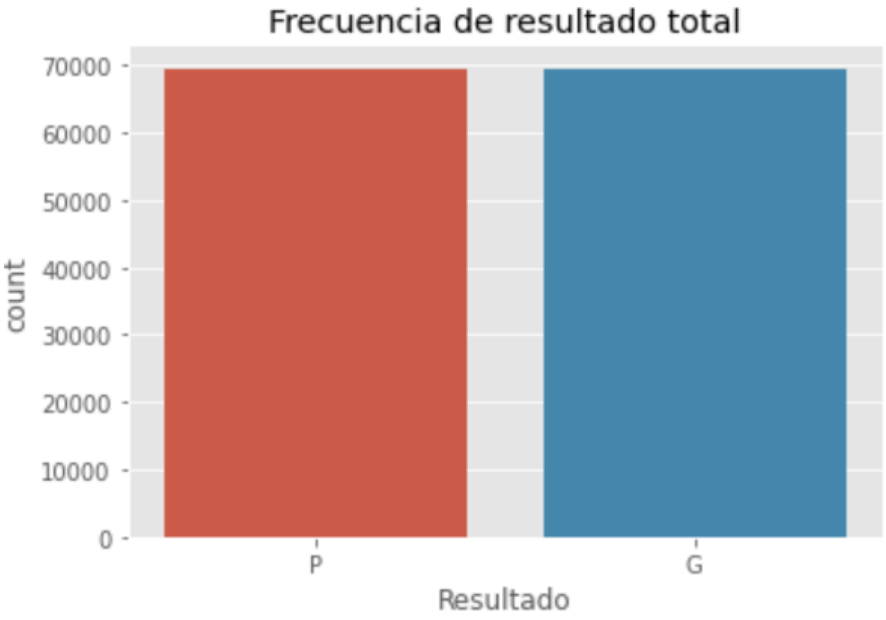


Imagen: Representa la totalidad de partidos ganados y perdidos en el dataset completo.

En la figura anterior se evidencia que cada posible resultado tiene la misma cantidad de registros que corresponde a 69.546 para cada uno.

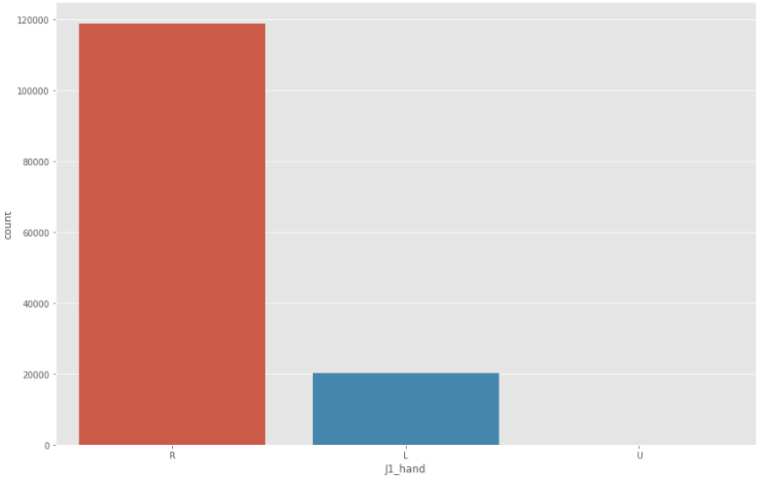
Distribución de las variables categóricas

Antes de transformar las variables categóricas en dummies, se verifican los posibles valores que cada una de estas puede tomar con el fin de determinar la cantidad de columnas adicionales que tendría el dataset.

J1_hand

Mediante un gráfico de barras se evidencian los posibles valores que la variable J1_hand puede tomar. Se encuentra una descripción atípica la cual corresponde a “U”, ya que se esperaría que la variable sólo tome dos valores, “R” o “L”. Al analizar los registros en los cuales está dicha descripción se encuentra que son 29 líneas que presentan tal condición y se determina que corresponde a registros en los cuales no se tiene conocimiento de la mano hábil del jugador (Unknown), por lo tanto, y con el fin de no afectar el modelo de clasificación ingresando una variable de la cual no se tiene certeza, y que crearía una columna dummy adicional que no aportaría valor al resultado, se decide eliminar los registros de la base de datos.

Figura 9 *Frecuencia de J1_hand*

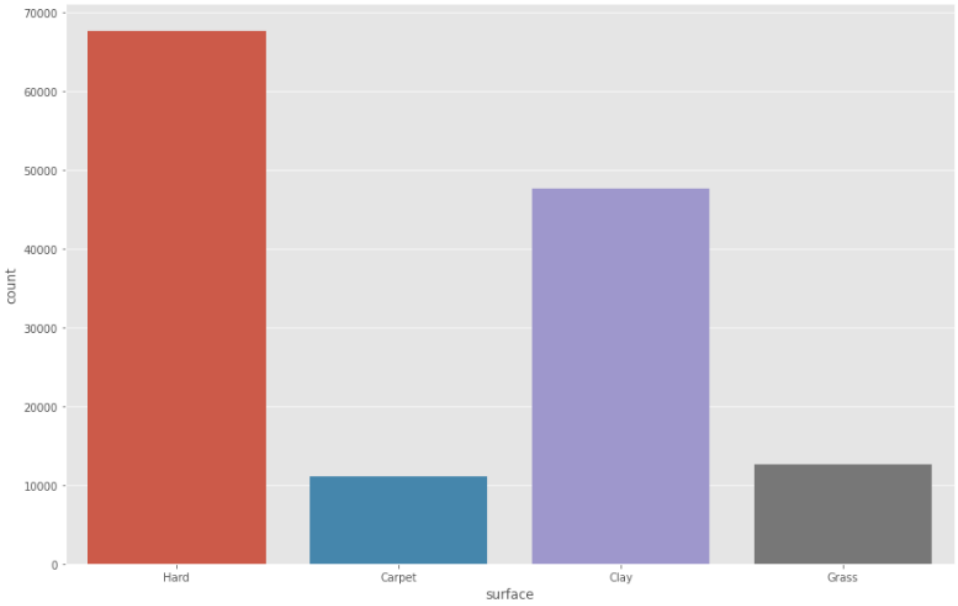


J2_hand

La distribución y diferentes posibles valores que puede tomar “J2_hand” son exactamente los mismos que “J1_hand” debido al paso en el cual se duplicó el dataset. Nuevamente se eliminan los 29 registros con la descripción “U”.

Surface

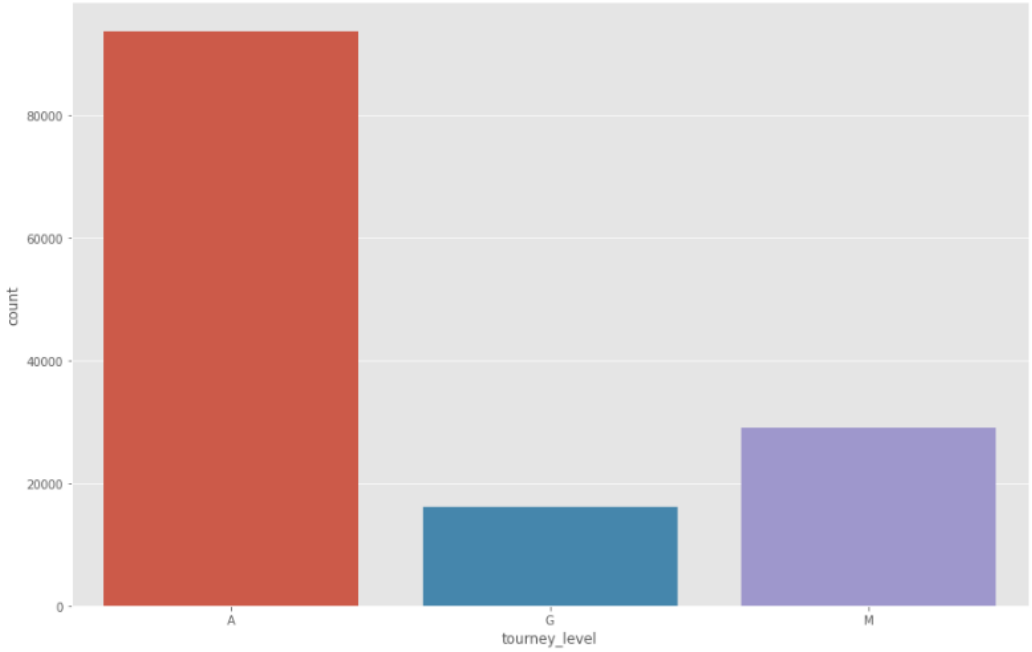
Figura 10 *Frecuencia de surface*



En la variable “surface” se evidencia que puede tomar 4 posibles resultados correspondientes a pista dura (Hard), cesped (Grass), polvo de ladrillo (Clay) y carpeta (Carpet). Esta última corresponde a una superficie de goma con unas características muy similares en velocidad y apariencia a las canchas de cemento.

Tourney level

Figura 11 *Frecuencia de Tourney level*



La variable del nivel de torneo puede tomar 3 diferentes valores que son aquellos con los que se decidió entrenar el modelo de predicción únicamente; G (Grand Slam), M (Masters 1000) y A (Cualquier otro torneo del circuito de tenis profesional). Se evidencia que los valores de "G" y "M" son menos comunes en el dataset lo cual tiene sentido teniendo presente que en el año sólo existen 4 torneos de categoría Grand Slam y 9 torneos Masters 1000 mientras que del resto de torneos del circuito ATP existen 52.

Modelamiento

Se decide realizar el modelo con una regresión logística ya que es una técnica comúnmente utilizada para la clasificación de datos y, en particular, para predecir resultados en diferentes deportes, incluyendo el tenis. La regresión logística es una técnica de aprendizaje supervisado que permite modelar la relación entre una variable de resultado binaria (en este caso, ganar o perder un partido) y una o más variables explicativas (por ejemplo, la clasificación de los jugadores, la superficie de la cancha, el historial de enfrentamientos, etc.).

En el caso específico de predecir el ganador en encuentros profesionales de tenis, la regresión logística puede ser especialmente útil debido a varias razones:

- Gran cantidad de datos disponibles: Se cuenta con una base de datos muy extensa y completa con información que puede ser utilizada para entrenar un modelo de regresión logística y mejorar la precisión de las predicciones.

- Variables explicativas significativas: Hay varias variables que han demostrado ser significativas en la predicción de los resultados de los partidos de tenis, como la clasificación de los jugadores, la edad, la superficie de la cancha y el historial de enfrentamientos previos. Estas variables pueden ser incluidas en el modelo para mejorar la precisión de las predicciones.
- Capacidad de interpretar los resultados: Este tipo de regresión permite una interpretación de los resultados de manera clara y sencilla, además se pueden realizar pruebas para determinar la significancia estadística de las variables explicativas del modelo.
- Escalabilidad: Es una técnica escalable y puede manejar grandes conjuntos de datos lo cual es muy útil teniendo en cuenta que el dataset cuenta con más de 130.000 registros.

Por lo tanto, para el presente trabajo se realizarán dos iteraciones con un modelo de regresión logística para clasificar datos. En la primera iteración, se ajustará un modelo de regresión logística utilizando todos los atributos del conjunto de datos. El objetivo de esta iteración es evaluar el rendimiento del modelo utilizando todos los datos disponibles. En la segunda iteración, se aplicará PCA (análisis de componentes principales) al conjunto de datos antes de entrenar el modelo de regresión logística. Si bien la utilización de PCA puede proporcionar beneficios en términos de reducción de dimensionalidad y mejorar el rendimiento del modelo de regresión logística, es importante tener en cuenta que la interpretación de los resultados puede verse afectada ya que los atributos iniciales se transforman en componentes principales que son combinaciones lineales de los atributos originales por lo que se debe analizar si las componentes presentan algún grado de

interpretabilidad o no. En ambas iteraciones se utilizará GridSearch para encontrar los mejores parámetros del modelo.

Criterio de aceptación

Actualmente existen diversos modelos de predicción de resultados en tenis de campo que se usan principalmente en aplicaciones de apuestas deportivas y, aunque no existe un valor conocido de acierto mínimo de dichos modelos, en general se espera que tengan una precisión lo suficientemente alta como para superar al azar y a las cuotas de las casas de apuestas, por lo que el accuracy esperado es de, al menos, el 70%. Por lo tanto, el criterio de aceptación para el modelo que se está desarrollando será obtener un accuracy mínimo de 70%.

Además, se revisará la correlación existente entre el ganador y su posición en el ranking, ya que se puede presumir que el jugador mejor ubicado debería ganar el partido, no obstante, es un detalle que se revisará con atención ya que hay jugadores que son especializados en cierto tipo de superficies y torneos, por lo que la posición en el ranking no estaría necesariamente acorde con el ganador del encuentro.

One-hot encoding

Una vez se ha organizado la base de datos, se procede a crear las variables dummies para las características categóricas. Se obtiene una base de datos con 139.034 registros y 22 columnas, que se consideran que son razonables y se esperaría que un modelo de clasificación procesara dichos datos de manera eficiente. Se realiza un único cambio y es en la descripción de la variable a predecir que quedó con el nombre de “Resultado_P” después de la transformación, por lo que se cambia el nombre a “Resultado” mientras que los valores de 1 y 0 en dicha columna se reemplazan por “P” y “G”, respectivamente, para permitir una mejor identificación de la variable que se quiere predecir.

División del conjunto de datos y escalamiento

Se realiza la división del conjunto de datos en los grupos de entrenamiento y de testeo con una proporción de 80-20 con los siguientes parámetros:

- X - Datos de entrada los cuales no contienen la variable a predecir (Resultado)
- y.values.reshape(-1,1) - Los datos objetivo (Resultado), remodelados en un vector de columna utilizando el método reshape() de un array NumPy, con -1 indicando que el tamaño de esa dimensión debe ser inferido a partir de los datos.
- train_size=0.8 - La proporción de los datos a utilizar para el entrenamiento, configurada en el 80% de los datos.
- random_state=1234: El estado aleatorio para generar la división de entrenamiento/prueba. Al fijar el mismo estado aleatorio, se obtiene la misma división cada vez que se ejecute el código.
- shuffle=True: si se deben mezclar aleatoriamente los datos antes de dividirlos en conjuntos de entrenamiento/prueba. Si se establece en False, los datos se dividirán en el mismo orden en que aparecen en la matriz original.

Figura 12 *Codificación de conjunto de datos de test y prueba*

```
X_train, X_test, y_train, y_test = train_test_split(
                                       X,
                                       y.values.reshape(-1,1),
                                       train_size   = 0.8,
                                       random_state = 1234,
                                       shuffle      = True
                                   )
```

Primera iteración

Se definen los parámetros y valores para ajustar el modelo con GridSearch y se crea el modelo de regresión logística con los mejores parámetros

Figura 13 *Codificación del pipeline primera iteración*

```
# Crear un objeto Pipeline que incluya el escalado y regresión logística
pipe = Pipeline([
    ('scaler', MinMaxScaler()),
    ('logistic', LogisticRegression(solver='liblinear'))
])

# Definir los parámetros que deseas ajustar y los valores que deseas probar
param_grid = {
    'logistic__C': [0.01, 0.1, 1, 10, 100],
    'logistic__penalty': ['l1', 'l2']
}

# Crear un objeto GridSearchCV y ajustarlo en los datos de entrenamiento
grid = GridSearchCV(pipe, param_grid, cv=5, scoring='accuracy')
grid.fit(X_train_scaled, y_train)

# Obtener el mejor modelo encontrado
best_model = grid.best_estimator_

# Predecir en los datos de prueba usando el mejor modelo encontrado
y_pred = best_model.predict(X_test_scaled)
```

Segunda iteración

Se definen los parámetros y valores para ajustar el modelo con GridSearch y PCA y se crea el modelo de regresión logística con los mejores parámetros

Figura 14 *Codificación del pipeline segunda iteración*

```
# Crear un objeto Pipeline que incluya el escalado, PCA y regresión logística
pipe = Pipeline([
    ('scaler', MinMaxScaler()),
    ('pca', PCA(n_components=2)),
    ('logistic', LogisticRegression(solver='liblinear'))
])

# Definir los parámetros que deseas ajustar y los valores que deseas probar
param_grid = {
    'pca__n_components': [2, 3, 4, 5, 10, 15],
    'logistic__C': [0.01, 0.1, 1, 10, 100],
    'logistic__penalty': ['l1', 'l2']
}

# Crear un objeto GridSearchCV y ajustarlo en los datos de entrenamiento
grid = GridSearchCV(pipe, param_grid, cv=5, scoring='accuracy')
grid.fit(X_train_scaled, y_train)

# Obtener el mejor modelo encontrado
best_model1 = grid.best_estimator_

# Predecir en los datos de prueba usando el mejor modelo encontrado
y_pred = best_model1.predict(X_test_scaled)
```

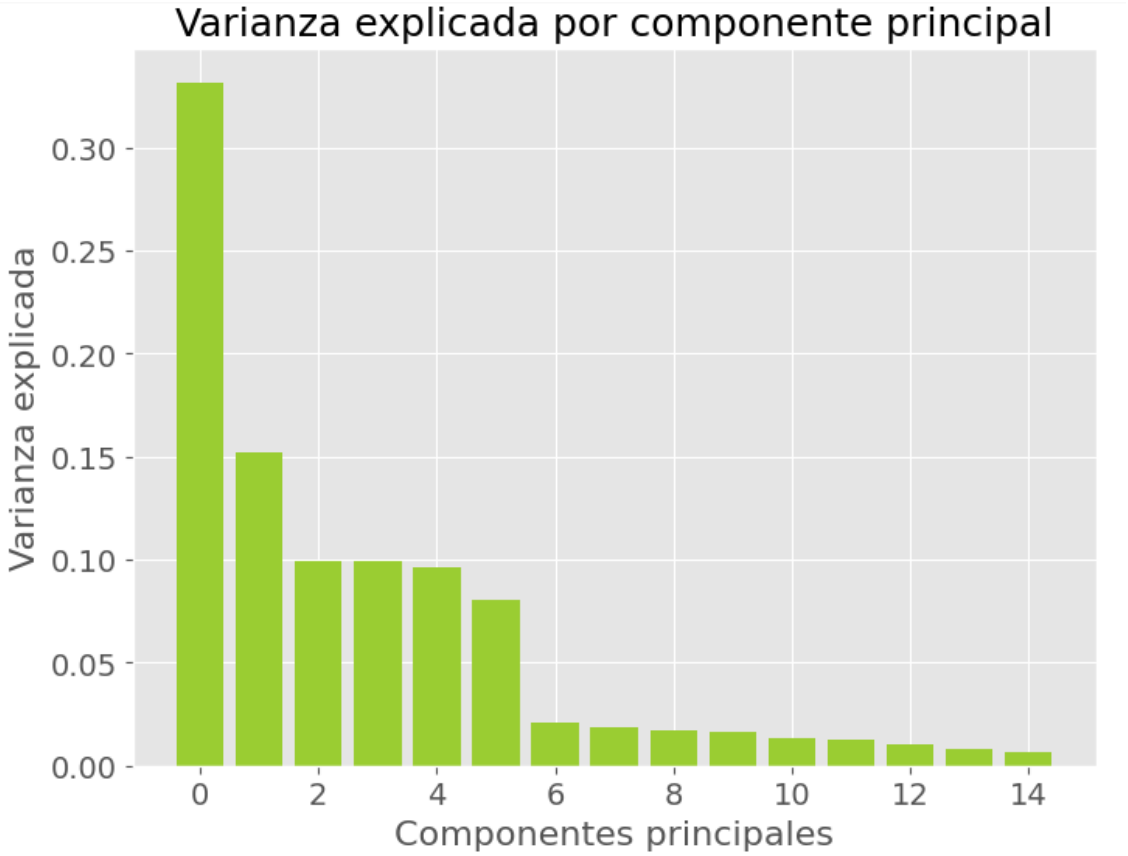
Evaluación

Tabla 3 *Resultados iteraciones del modelo*

Métrica	Iteración 1	Iteración 2
Accuracy	2.1	11.8
Curva ROC	3.5	6.9

Se evidencia que la primera iteración tuvo unos indicadores de desempeño superiores a los obtenidos en la segunda que se explican principalmente por la pérdida de información al implementar el método PCA, ya que el número de componentes utilizados en el modelo fueron 15 de un total de 22, lo cual indica que la estructura de datos es compleja y se requieren de múltiples componentes para explicar la varianza de los datos lo cual se observa en la siguiente gráfica.

Figura 15 *Varianza explicada por cada componente con PCA*



Se observa que la componente que mayor varianza explica lo hace en menos de un 35%, lo cual sugiere que la estructura de los datos es compleja y que se necesitan múltiples componentes para explicar la varianza de los datos.

Despliegue

Con el fin de disponibilizar el modelo para que cualquier usuario pudiera utilizarlo y obtener las predicciones con base en los datos que ingresen, se decidió desplegar el modelo con FastAPI ya que es una aplicación que presenta un alto rendimiento para aplicaciones web en Python, además puede manejar grandes volúmenes de información y solicitudes por lo que es una excelente opción para tener respuestas rápidas en tiempo real.

Guardar el modelo.

Una vez de determinó el mejor modelo para predecir resultados, que en este caso fue la regresión logística sin PCA, se guarda el modelo en una representación serializada del mismo en un archivo joblib, donde se incluye además el escalador utilizado en el entrenamiento para así garantizar que los datos nuevos que vayan a pasar por el modelo tengan el mismo tratamiento que aquellos con los cuales se entrenó.

Configuración de la aplicación

En el archivo main.py, se lleva a cabo la inicialización y configuración de la aplicación FastAPI para el despliegue del modelo de regresión logística. Este paso permite preparar la aplicación antes de su ejecución y garantizar su correcto funcionamiento. Durante la inicialización, se carga el modelo previamente entrenado utilizando la biblioteca joblib. El modelo cargado se almacena en una variable para su posterior uso en la inferencia y se definen las rutas de la API que permitirán la interacción con el modelo, en particular, se establece una ruta POST que acepta los datos del partido de tenis y devuelve la predicción del resultado (“P” ó “G”).

Estructura de datos y validaciones

Teniendo en cuenta que el modelo en este punto no tiene restricciones y puede recibir datos incoherentes y aún así generar una predicción, se realiza una validación de los campos de entrada donde se acota el tipo de dato que recibe el modelo así como el rango de valores que puede tomar, de esta manera se minimiza la posibilidad de obtener predicciones con datos que no tengan sentido. El tipo de datos y los valores que puede recibir cada variable se presenta en la tabla **Tabla 4** Estructura de datos de entrada del modelo.

Tabla 4 *Estructura de datos de entrada del modelo*

Variable	Tipo de Dato	Valores que puede recibir
J1_2ndWon	Integer	0 a 60
J1_ace	Integer	0 a 30
J1_bpFaced	Integer	0 a 30
J1_df	Integer	0 a 20
J1_age	Integer	15 a 40
J1_ht	Integer	150 a 220
J1_rank	Integer	1 a 100
J2_2ndWon	Integer	0 a 60
J2_ace	Integer	0 a 30
J2_bpFaced	Integer	0 a 30
J2_df	Integer	0 a 20
J2_age	Integer	15 a 40
J2_ht	Integer	150 a 220
J2_rank	Integer	1 a 100
J1_hand_R	Integer	0 ó 1
Surface	String	C, G, H ó P
Tourney_level	String	G, M ó A
J2_hand_R	Integer	0 ó 1

El código se generó de manera tal que sólo se deba ingresar una superficie y un tipo de torneo para generar la predicción ya que internamente está parametrizado para que asigne automáticamente el valor de 1 a la característica que corresponda con el valor ingresado por el usuario mientras que se asigna 0 en los demás casos, esto permite minimizar la complejidad del modelo al momento de ingresar los datos. También, se determinaron rangos para las variables numéricas que tuvieran un mayor sentido de cara al modelo predictivo de la siguiente manera:

2nd_Won: Se define el valor máximo de puntos ganados con el segundo servicio en 60 para tener un margen de holgura amplío ya que en un partido normalmente no se juegan más de 40 puntos con el segundo saque.

Ace: Se establece en 30 el límite de saque directos ya que normalmente el valor no supera los 20. Aunque hay un registro de un partido donde se dieron mas de 100 aces de cada uno de los dos jugadores se evidencia que es un hecho atípico ya que el match tuvo una duración de más de 11 horas y se jugó en tres días **(POZA, s.f.)**.

Bp_faced: 30 es el valor máximo asignado. Un partido normalmente no tiene más de 15 o 20 opciones de quiebre.

Df: Las dobles faltas se establecen en 20 cómo máximo siendo esta una variable que en muchos casos queda en cero al final de los encuentros.

Age: Teniendo en cuenta que los casos de personas menores de 15 o mayores de 40 años que estén en el top 100 del ranking de tenis es muy escaso, se definen estos valores cómo límites.

Ht: La altura se define entre 1,50 y 2,20 que son los rangos donde ha habido jugadores en el top 100.

Rank: Sólo puede tomar valores entre 1 y 100 ya que son con los cuales se entrenó el modelo.

Hand_R: Sólo puede tomar el valor de 1 si el jugador es diestro o 0 si es zurdo.

Surface: Puede tomar el valor de una de las 4 posibles superficies: Clay (“C”), Grass (“G”), Hard (“Hard”) y Carpet (“P”).

Tourney_level: Se refiere al tipo de torneo y para el cual sólo pueden haber 3 posibilidades: Grand Slam (“G”), Masters 1000 (“M”) y otros eventos del circuito ATP (“A”).

Así pues, se garantiza que el modelo genere predicciones basado en valores coherentes y en línea con lo que ha sido usual hasta el momento del entrenamiento del mismo. Por otro lado, para los casos en los que el usuario ingrese valores por fuera del rango indicado, la aplicación

generará un error que no le permitirá conocer el valor de la predicción hasta que no realice el ajuste en las variables correspondientes.

Conclusiones

Se pudo determinar cuáles con las variables que explican con un accuracy del 84% quién ganaría un partido de tenis profesional teniendo los datos históricos de las estadísticas del match. Es importante tener en cuenta que sólo se consideraron partidos entre jugadores dentro del top 100 desde 1992 que es el año desde el cual se tiene registro de las estadísticas de los encuentros.

Se evidenció que la eliminación de variables altamente correlacionadas (>80%) no afectó en gran medida la información de la base de datos y pudo ser útil para mejorar la precisión del modelo. La correlación entre variables puede crear problemas en el modelo, ya que puede introducir sesgos y aumentar la varianza de este.

Es posible que el desempeño del modelo se hubiera dado debido a los jugadores que se tuvieron en cuenta para entrenarlo y evaluarlo, pues son los tenistas que presentan una mayor regularidad en el circuito y extrapolar el modelo a jugadores fuera del top 100 podría no tener un resultado favorable debido a que el desempeño de los jugadores con un ranking más bajo es más irregular.

En el modelo se tuvieron en cuenta variables categóricas como lo fueron la superficie, la mano hábil de cada jugador y el tipo de torneo. Tales variables se consideran importantes ya que pueden ser determinantes en el resultado de un partido, brindando ventajas competitivas a algunos jugadores en ciertos terrenos o por el contrario afectar su desempeño en otros.

Aunque el modelo de predicción desarrollado puede proporcionar una buena aproximación de los resultados de los partidos de tenis de campo, es importante recordar que el tenis es un deporte complejo y que factores externos, como lesiones o cambios en las condiciones climáticas, pueden afectar el resultado de un partido. Por lo tanto, los modelos de predicción no siempre serán precisos al 100%.

Recomendaciones

Para futuras investigaciones relacionadas con la predicción de resultados en el tenis de campo se recomienda identificar las variables que diferencian a los jugadores que hacen parte del top 100 respecto de los que no. Es altamente probable que un modelo que funcione en uno de los dos grupos no funcione en el otro y es importante tenerlo en cuenta puesto que un sólo modelo predictivo puede no tener un buen desempeño en todos los casos.

Aunque en este estudio se utilizaron variables específicas de los partidos de tenis de campo, sería interesante explorar la inclusión de nuevas variables que puedan tener un impacto significativo en la predicción de resultados. Esto podría incluir variables relacionadas con el estado físico de los jugadores, condiciones climáticas o características del entorno de juego.

También, la presente investigación se baPara futuras investigaciones relacionadas con la predicción de resultados en el tenis de campo se recomienda identificar las variables que diferencian a los jugadores que hacen parte del top 100 respecto de los que no. Es altamente probable que un modelo que funcione en uno de los dos grupos no funcione en el otro y es importante tenerlo en cuenta puesto que un sólo modelo predictivo puede no tener un buen desempeño en todos los casos.

Aunque en este estudio se utilizaron variables específicas de los partidos de tenis de campo, sería interesante explorar la inclusión de nuevas variables que puedan tener un impacto significativo en la predicción de resultados. Esto podría incluir variables relacionadas con el estado físico de los jugadores, condiciones climáticas o características del entorno de juego.

También, la presente investigación se basó en datos recopilados desde el año 1991 hasta el 2019. Sería beneficioso actualizar el conjunto de datos de entrenamiento y validación utilizando información más reciente. Esto permitiría evaluar la capacidad predictiva del modelo en el contexto actual del tenis de campo y garantizar su aplicabilidad en el presente.

Además de la regresión logística, existen otros modelos de aprendizaje automático que podrían ser aplicados al análisis de resultados en el tenis de campo. Sería interesante realizar comparaciones entre diferentes modelos, como árboles de decisión, redes neuronales o métodos de

clasificación, para determinar cuál de ellos ofrece el mejor rendimiento en términos de precisión predictiva.

Finalmente, teniendo en cuenta que la regresión logística puede ser implementada en otros deportes individuales, sería valioso analizar la posibilidad de extrapolar el modelo a otros deportes de características similares al tenis de campo como lo son el tenis de mesa, el squash o el bádminton, donde existen diferencias marcadas en cuanto a la superficie, la técnica y el estilo de juego pero permitiría evaluar la efectividad del modelo en diferentes contextos deportivos y ampliar su aplicabilidad.

Referencias

Barnett, T. B. (2005). *DEVELOPING A TENNIS MODEL THAT REFLECTS OUTCOMES OF TENNIS MATCHES.* Retrieved 05 01, 2023, from https://bit.ly/3MN8PLc

Cmdsport. (2022, 06 08). Retrieved 05 01, 2023, from https://bit.ly/45zd7OK

Link, D. &. (2009). *Sports informatics - Historical roots, interdisciplinarity and future developments.* (I. J. sport, Ed.) Retrieved 05 02, 2023, from https://bit.ly/43xN9t3

Mitchell, T. M. (1997, 01 03). *Machine Learning*. Book News, Inc. Retrieved 05 03, 2023, from https://bit.ly/3OHOIQT

POZA, E. A. (n.d.). *La Web del tenis*. Retrieved from http://bit.ly/43yfcsT

Ricardo Gil Rubio, E. A. (2022, 03). *Universidad Santo Tomás.* Retrieved 05 03, 2023, from https://bit.ly/3oAfuQz

Robert P. Schumaker, O. K. (2010). *Semanticscholar*. Retrieved 05 03, 2023, from https://bit.ly/437Xz2K

Soto-Valero, C. (2017, 07). *A Gaussian mixture clustering model for characterizing football players using*. Retrieved from RICYDE. Revista Internacional de Ciencias del Deporte: https://bit.ly/3qijN3H

Subramanian Rama Iyer, R. S. (2009, 04). Sciencedirect. ELSEVIER. Retrieved 05 03, 2023, from https://bit.ly/3IIuBOM