Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales

Investigación sobre la aplicación de técnicas de filtrado colaborativo para la recomendación de jugadores a equipos de fútbol

Tesis de Licenciatura en Ciencias de Datos

Pablo Groisman

Director: Dr. Andrés Farall

Codirector: Ing. Manuel Duran

Buenos Aires, 2025

APLICACIÓN DE TÉCNICAS DE FILTRADO COLABORATIVO PARA LA RECOMENDACIÓN DE JUGADORES A EQUIPOS DE FÚTBOL

En el fútbol, la elección adecuada de futbolistas por parte de los clubes es una tarea difícil y a la vez fundamental para el desempeño de un equipo. Tradicionalmente, este trabajo lo realizaban los equipos de scouts, o los mismos dirigentes y cuerpo técnico. Hoy en día, existen múltiples herramientas para evaluar el desempeño de un jugador y la mayoría de ellas tiene en cuenta estadísticas clásicas como goles o asistencias o algunas más rebuscadas como el XG (expected goals).

Inspirada en el éxito de los métodos de filtrado colaborativo en los sistemas de recomendación de películas o productos a usuarios, esta tesis explora la adaptación de los trabajos de los ganadores del Gran Premio de Netflix, ajustando sus modelos y técnicas algorítmicas al ámbito del fútbol. Los sistemas creados mediante estos métodos permiten generar recomendaciones de jugadores a equipos basándose únicamente en el historial de rendimiento pasado, sin depender de estadísticas de juego convencionales. Este enfoque tiene el potencial de descubrir patrones latentes de compatibilidad entre jugadores y equipos, ofreciendo una herramienta innovadora para el análisis. La utilidad de estos modelos radica en su capacidad para combinar un rendimiento predictivo prometedor con una relativa interpretabilidad de sus componentes (como sesgos y factores latentes), buscando en este caso, no reemplazar, sino complementar los métodos de scouting y análisis estadístico ya existentes, enriqueciendo así el proceso de toma de decisiones en la selección de futbolistas.

Se desarrollan los modelos adaptados de [15, 13, 14], así como las técnicas de blending mencionadas junto a otras que se proponen en este trabajo. Se evalúan los métodos en métricas clásicas en el ámbito (RMSE, MAE, PMAE o MAPE), logrando superar a los benchmarks básicos, y se proponen otras dos métricas de ordenamiento para evaluar la utilidad de las recomendaciones (Spearman, Kendall). Luego, se estudian las "Top-5" recomendaciones de modelos particulares a través de un análisis que nos deja como corolario que un error predictivo más bajo no siempre se traduce en recomendaciones más útiles. Por último, se comentan las conclusiones del trabajo junto a lineas de continuación de la investigación y una breve motivación para cada una.

Palabras claves: Filtrado Colaborativo, Factorización Matricial, SVD, TIMESVD++, Modelos de Vecindad, Modelos de Factores Latentes, Dinámicas Temporales, Blending, Sistemas de Recomendación, Recomendacion de Jugadores de Fútbol, Gran Premio de Netflix.

FOOTBALL PLAYER RECOMMENDATION USING COLLABORATIVE FILTERING TECHNIQUES

In football, the appropriate selection of players for clubs to purchase is a difficult yet fundamental task for a team's performance. Traditionally, this work was carried out by scouting teams, or by club directors and coaching staff themselves. Nowadays, multiple tools exist to evaluate a player's performance, most of which consider classic statistics such as goals or assists, or more elaborate ones like xG (expected goals). Inspired by the success of collaborative filtering methods in recommendation systems for movies or products to users, this thesis aims to explore the adaptation of the works by the Netflix Grand Prize winners, adjusting their models and algorithmic techniques to the football domain. Systems created through these methods allow for the generation of player recommendations to teams based solely on past performance history, without relying on conventional game statistics. This approach has the potential to uncover latent compatibility patterns between players and teams, offering an innovative tool for analysis. The utility of these models lies in their ability to combine promising predictive performance with a relative interpretability of their components (such as biases and latent factors), seeking not to replace, but to complement existing scouting and statistical analysis methods, thereby enriching the decision-making process in footballer selection. The adapted models from [15, 13, 14] are developed, as well as the aforementioned blending techniques along with other proposals in this work. The methods are evaluated using classic metrics in the field (RMSE, MAE, PMAE), with better results than the basic benchmarks, and two additional ranking metrics are proposed to assess the utility of the recommendations (Spearman, Kendall). Subsequently, the "Top-5" recommendations from particular models are studied through an analysis that shows as a corollary that a lower predictive error does not always translate into more useful recommendations. Finally, the conclusions of the work are discussed along with future lines of research and a brief motivation for each.

Keywords: Collaborative Filtering, Matrix Factorization, SVD, TimeSVD++, Neighborhood Models, Latent Factor Models, Temporal Dynamics, Blending, Recommender Systems, Football Player Recommendation, Netflix Grand Prize.

AGRADECIMIENTOS

Quiero agradecer en primer lugar a la Universidad de Buenos Aires y en especial a todos los profesores de la Facultad de Ciencias Exactas y Naturales por brindarme la posibilidad de aprender de docentes de altísima calidad y prestigio, que enseñan su enorme conocimiento con muchísima pasión.

A Andy y Manu, mis directores, por mostrarme este proyecto, por acompañarme y aconsejarme durante el desarrollo de la investigación y por estar pendientes para ayudarme en el proceso.

También quiero agradecer a mi familia, por apoyarme y bancarme a lo largo de toda la carrera, por motivarme a siempre seguir estudiando y por acompañarme en cada paso de mis estudios.

A mis amigos de la facultad, de los que no solamente aprendí mucho, sino que también me ayudaron a divertirme y motivarme en el estudio. En particular me gustaría agradecer a Teo, por mostrarme esta carrera y motivarme a estudiarla juntos. Sin vos este camino no hubiera sido tan lindo.

A mis amigos y personas cercanas fuera de la facultad, por su apoyo constante, por los espacios de estudio compartido, por estar presentes acompañándome durante el recorrido y por ser curiosos y buscar entender y participar de lo que fui estudiando a lo largo de estos años.

Índice general

1.. Introducción 1
1.1. Motivación del estudio 1
1.2. Métodos utilizados en el fútbol 2
1.3. El desafío del Netflix Prize y su legado 5
1.4. Estructura del resto de la tesis 6

2.. Marco teórico y antecedentes 7
2.1. Definición de sistemas de recomendación de filtrado colaborativo y clasificación 7
2.2. Predictores base 8
2.3. Incorporación de dinámicas temporales a sesgos y parámetros 9
2.4. Fundamentos matemáticos relevantes 10
2.5. Algoritmos de aprendizaje 12
2.6. Métodos de blending de modelos 13

3.. Datos y Preprocesamiento 15
3.1. Filtrado de observaciones y densificación 15
3.2. Conjunto final de entrenamiento y test 18

4.. Modelos Implementados 21
4.1. Predictores base 22
4.1.1. Modelo base SIN sesgos temporales 22
4.1.2. Modelo base CON sesgos temporales (por temporada) 22
4.1.3. Modelo base CON sesgos temporales (por Bins) 23
4.1.4. Modelo base con más variables (adaptado de TimeSVD++) 23
4.2. Modelos de Vecindario 24
4.2.1. Item-based Similarity model 24
4.2.2. Modelo de Vecindario con Pesos Entrenados 25
4.3. Modelos de Factores Latentes 25
4.3.1. SVD Simple 26
4.3.2. SVD Básico 26
4.3.3. SVD con Biases Temporales 27
4.4. Modelos que Incorporan Ambos Métodos (Factorización y Vecindario) . . . 27
4.4.1. SVD++ 27
4.4.2. TimeSVD++ 28

4.4.3. TimeSVD++ Simplificado 28
4.5. Blending 28
4.6. Hiperparámetros 29

5.. Evaluación y Resultados 31
5.1. Métricas de evaluación 31
5.2. Benchmarks 33
5.3. Análisis de hiperparámetros y dimensión del embedding 34
5.4. Resultados de Modelos y Benchmarks 36
5.5. Resultados de Blending 38
5.6. Análisis de Recomendaciones Top-N 40
5.6.1. Modelo simple SVD 41
5.6.2. Modelo SVD Básico versión 2 42
5.6.3. Modelo de Vecindario Basado en Similaridad 43
5.6.4. Modelo de Vecindario con Pesos Globales(`pred_item_vector_model`) 44
5.6.5. Modelo TimeSVD++ Simplificado(`pred_timeSVDpp_simplified`) . . 45

6.. Análisis de Resultados y Discusión 47
6.1. Análisis de resultados con métricas clásicas (RMSE, MAE, PMAE) 47
6.1.1. Análisis general 47
6.1.2. Impacto de agregar sesgos temporales 48
6.1.3. Análisis de modelos de vecindario vs modelos de factores latentes . . 49
6.1.4. Comparación de modelos de blending 49
6.1.5. Interpretación de resultados en el contexto del fútbol 50
6.2. Comparación y análisis de recomendación de jugadores 50
6.2.1. Modelo simple SVD o BasicSVD 0 50
6.2.2. Modelo basicSVD versión 2 51
6.2.3. Modelo TimeSVD++_simplified 51
6.2.4. Modelo pred_similaridad 51
6.2.5. Modelo pred_item_vector_model 52
6.3. Análisis de la Distribución de Parámetros y Componentes de Predicción . . 52
6.3.1. Distribución de los ratings predichos de los modelos 53
6.3.2. Descomposición de la Predicción para un Jugador Específico 55
6.4. Análisis de resultados con métricas de ordenamiento 56
6.5. Comparación con métodos tradicionales 57
6.6. Limitaciones, sesgos y su impacto 58

7.. Conclusiones y futuros pasos 61
7.1. Conclusiones finales 61
7.2. Aportes metodológicos y prácticos de la tesis 62

7.3. Futuros pasos: Posibles líneas de continuación de la investigación y breve motivación de cada una 63

1. INTRODUCCIÓN

1.1. Motivación del estudio

En el dinámico y competitivo mundo del fútbol profesional, la identificación y selección de jugadores adecuados es un desafío constante para el éxito de los equipos. Tradicionalmente, este proceso fue realizado mediante ojeadores, el análisis de estadísticas de rendimiento convencionales y la intuición de los cuerpos técnicos y dirigentes. Si bien estos métodos han demostrado su valía, el creciente volumen de datos disponibles y los avances en el campo de la ciencia de datos ofrecen nuevas oportunidades para complementar y potenciar la toma de decisiones en la captación de jugadores con características adecuadas para un equipo.

Los sistemas de recomendación son herramientas tecnológicas de gran valor e impacto. Su presencia es fundamental en plataformas de comercio electrónico como Amazon, que personaliza la oferta de productos [16], y en servicios de entretenimiento como Netflix, cuyo desafío público impulsó significativamente el campo [5]. Estos sistemas no solo enriquecen la experiencia del usuario al facilitar el descubrimiento de ítems de interés, sino que también generan un considerable valor de negocio al incrementar la interacción y la conversión del cliente. Esto también aplica para los clubes de fútbol, donde contratar jugadores con bajo valor de mercado, adecuados para el equipo, puede generar un eventual crecimiento del jugador, aumentando su valor y brindando al club crecimiento económico.

Esta tesis surge de la motivación de explorar y evaluar cómo técnicas avanzadas de modelado provenientes del campo del filtrado colaborativo en los sistemas de recomendación, pueden ser adaptadas y aplicadas para generar recomendaciones de jugadores a equipos de fútbol. La aplicación de técnicas de filtrado colaborativo a este dominio presenta una perspectiva innovadora. Mientras que la mayoría de los métodos se enfocan en atributos explícitos y medibles (algunos más fáciles de medir, como la cantidad de goles marcados y otros más difíciles, como la participación de un jugador en los goles de su equipo), el filtrado colaborativo tiene el potencial de descubrir patrones sutiles y relaciones implícitas de los equipos y jugadores únicamente a través de los datos históricos de desempeño. En esencia, se busca responder a la pregunta: ¿qué jugadores, basándose en su trayectoria y desempeño pasado, podrían encajar y rendir bien en un determinado equipo, considerando también las características de dicho equipo a partir de los futbolistas que han pasado por éste?

El potencial de implementar sistemas de recomendación al ámbito del fútbol es altamente atractivo debido a la posibilidad de ofrecer a los clubes instrumentos basados en datos, complementando la experiencia humana para la toma de decisiones en el mercado

de fichajes. En particular, en este trabajo se busca analizar y estudiar el enfoque a través del desarrollo de un sistema que utiliza una técnica distinta a las tradicionales en el área.

La inspiración principal del trabajo viene del éxito de los algoritmos de los ganadores del Gran Premio de Netflix, que lograron modelar gustos de usuarios para recomendar películas a través de filtrado colaborativo. En ese caso superaron en error a cualquier otra técnica desarrollada, mostrando una interpretabilidad destacable, donde pudieron modelar y representar en un mismo espacio vectorial a usuarios y películas, logrando comprender el significado de las dimensiones de ese espacio vectorial.

La idea de este trabajo es investigar si esa misma lógica puede aplicarse para encontrar relaciones entre un jugador y un equipo de fútbol, brindando una herramienta potente tanto en recomendación como en explicabilidad. Si bien pueden ser peores en predicción que otros modelos como redes neuronales profundas, la utilidad de estos métodos no solo radica en las recomendaciones, sino también en la interpretación de las mismas, ya que a partir del modelado realizado, se puede analizar las valores de los parámetros y estudiar su significado de manera más sencilla y rápida.

Esta investigación busca tender un puente entre la precisión algorítmica y la complejidad del mundo del fútbol, con la esperanza de ofrecer herramientas que ayuden a tomar decisiones más informadas, sin pretender reemplazar la indispensable experiencia humana, sino complementándola.

De esta forma, de algún modo, existe un desafío doble: por un lado, la adaptación técnica de modelos diseñados para preferencias subjetivas (gustos de películas) a un dominio donde el "rating" (más adelante explicaremos de donde sale) refleja un rendimiento un poco más objetivo, aunque aún influenciado por múltiples factores contextuales. Por otro lado, la exploración de la capacidad de estos modelos para capturar las complejas interacciones entre el estilo de un jugador, la filosofía de un equipo, la influencia de una liga o una temporada particular, en un entorno tan cambiante y complejo como el fútbol, para generar recomendaciones verdaderamente valiosas y útiles.

Vale la pena comentar que la investigación se realiza dentro del marco de un convenio de la Facultad de Ciencias Exactas y Naturales con el club Racing de Santander, de España. Por lo que se destaca la participación, tanto del club como de la Universidad de Buenos Aires, en el apoyo al estudio.

1.2. Métodos utilizados en el fútbol

La identificación y recomendación de jugadores para fichajes profesionales se fue transformando drásticamente en las últimas décadas. Históricamente, la selección de futbolistas para un equipo se basaba en la intuición de ojeadores y entrenadores, con metodologías poco científicas. Hoy en día, ante la enorme competencia deportiva y financiera, los clubes profesionales complementan la experiencia humana con análisis de datos avanzados para

tomar decisiones de contratación más informadas.

Hace unos años, los clubes utilizaban modelos estadísticos sencillos para evaluar jugadores. Un enfoque clásico es el método plus/minus ajustado [20], originado en otros deportes y adaptado al fútbol, que mide la influencia de un jugador en el marcador comparando la tasa de goles anotados y recibidos por su equipo cuando el jugador está en el campo versus cuando no lo está. Este tipo de análisis mediante regresión proporciona una medida del impacto neto del jugador en el rendimiento colectivo dentro del equipo al que pertenece. Sin embargo, mientras estos modelos ofrecen métricas para evaluar a un jugador en un equipo, no proporcionan una manera de predecir cómo le irá a esos jugadores en un nuevo equipo. Aún así, estas métricas sentaron las bases para evaluar objetivamente el rendimiento, más allá de los goles o asistencias simples. También dieron origen a indicadores ahora comunes en scouting, como el expected goals (xG), entre otros.

En paralelo a los enfoques puramente estadísticos, se han utilizado métodos de decisión multicriterio [4] para estructurar la evaluación de fichajes. Estos métodos abordan la selección de jugadores como un problema de decisión que involucra numerosos criterios (técnicos, físicos, tácticos e incluso financieros) utilizando diferentes herramientas de optimización, aprendizaje automático y toma de decisiones. Estos métodos con enfoques heurísticos permiten explorar miles de combinaciones de jugadores para armar el equipo ideal según métricas definidas por el club.

Luego, otro método que se comenzó a utilizar es el clustering, para agrupar jugadores según su estilo de juego o características, con el fin de encontrar perfiles similares [2]. En la búsqueda de refuerzos, es común que el cuerpo técnico diga: “Queremos un jugador del estilo de X”. Tradicionalmente esto implicaba horas de vídeo buscando a alguien que “se parezca” al referente deseado. El clustering automatiza y objetiviza esa búsqueda: a partir de datos de rendimiento (estadísticas de pases, velocidad, recuperaciones, tiros, etc.), algoritmos no supervisados agrupan jugadores con patrones similares, revelando equivalentes estadísticos al jugador prototipo buscado. Estas agrupaciones permiten a un club filtrar qué futbolistas podrían encajar bien en su modelo de juego: si un equipo necesita un mediocampista creativo de posesión, el análisis de clúster puede sugerir nombres de ligas remotas con ese perfil de juego. De este modo, el clustering surge como una de las primeras herramientas en permitir recomendar jugadores a equipos a través de la similitud de jugadores mediante técnicas de machine learning no supervisado, convirtiéndose en una herramienta valiosa para reemplazar o complementar jugadores en el equipo manteniendo la filosofía táctica.

Con el crecimiento del registro de datos en el deporte y el crecimiento de la ciencia de datos, tomaron protagonismo los métodos de aprendizaje automático que aprenden patrones complejos a partir de grandes conjuntos de datos de jugadores. En lugar de predefinir un modelo estadístico simple, el machine learning permite descubrir relaciones no evidentes entre las características de un futbolista y su éxito futuro o adecuación

a cierto nivel competitivo. Dentro de este marco existen muchos trabajos distintos que ofrecen perspectivas interesantes. Como ejemplo representativo se encuentra el estudio de Ćwikliński et al. (2021) [26] donde presentaron un sistema para predecir la "transferencia exitosa" de un jugador a un equipo, probando algoritmos como Random Forest, Naive Bayes y AdaBoost sobre datos históricos reales.

En la práctica, los modelos supervisados de machine learning permiten combinar decenas de variables de un jugador (estadísticas de juego, indicadores físicos, edad, experiencia, etc.) y generar predicciones o calificaciones. Muchos clubes y empresas utilizan estas técnicas donde el output puede ser un ranking de candidatos sugeridos que maximicen cierta métrica (ej. expectativa de gol + asistencias) o una advertencia sobre jugadores con alto riesgo de no adaptarse. Estos modelos ofrecen un buen trade-off entre buenos resultados e interpretabilidad. En resumen, el machine learning supervisado aporta objetividad y capacidad predictiva, transformando el scouting en un proceso más científico. Estudios como los citados validan su utilidad y han allanado el camino para una adopción creciente de estos modelos en el fútbol profesional.

Más adelante aparecieron técnicas que aprovecharon el crecimiento del aprendizaje profundo. Este lleva las capacidades del aprendizaje automático un paso más allá, empleando redes neuronales de muchas capas capaces de detectar patrones altamente complejos en datos masivos. En el contexto de recomendación de jugadores, el deep learning se ha aplicado sobre todo cuando la información disponible es muy rica o no estructurada, como secuencias espacio-temporales de partidos, posicionamiento de jugadores o incluso videos. Un área de avance ha sido el uso de redes neuronales recurrentes y convolucionales para analizar datos de seguimiento (tracking data) de los jugadores durante los partidos, aprendiendo representaciones de estilo de juego táctico. También surgen investigaciones que usan redes neuronales gráficas (GNN) para modelar al equipo como un grafo (jugadores como nodos, interacciones como aristas), lo que permite evaluar cómo encajaría un jugador nuevo en la red de pases y movimientos de un conjunto. Aunque muchas de estas aplicaciones de deep learning todavía están en fases experimentales, apuntan a capturar la dimensión táctica y contextual del rendimiento, algo crucial para la compatibilidad jugador–equipo que también se busca capturar a través de los métodos de filtrado colaborativo.

La adopción de estos métodos no se quedó solo en la teoría, sino que encontró terreno fértil en clubes profesionales y empresas especializadas en scouting. Desde mediados de la década de 2010, numerosos equipos invirtieron en departamentos de análisis de datos para apoyo en fichajes. Un referente es el Brentford FC en Inglaterra [3], mencionado justamente por su filosofía "Moneyball". Brentford emplea modelos analíticos para guiar la búsqueda de jugadores. Por ejemplo, desarrollaron un sistema para comparar la calidad de ligas distintas y así encontrar jugadores destacados en campeonatos de menor perfil –y por lo tanto, más baratos– que puedan rendir en niveles superiores.

En conclusión, la recomendación de jugadores hoy en día es un proceso híbrido y

sofisticado que ya se puso en práctica profesional. Además, los clubes muestran que los métodos no representan una herramienta única, sino que la complementariedad entre las técnicas estadísticas, modelos avanzados y los scouts humanos es lo que genera valor. La ciencia de datos aplicada al fútbol sigue evolucionando rápidamente, pero ya ha probado ser una ventaja competitiva, convirtiendo el arte del scouting en una disciplina más precisa sin quitarle la mirada humana y la pasión que hace único a este deporte.

1.3. El desafío del Netflix Prize y su legado

Antes de sumergirnos en los detalles técnicos, es crucial entender el contexto y la magnitud del Netflix Prize. Lanzado en octubre de 2006, Netflix ofreció un premio de 1 millón de dólares al primer equipo que pudiera mejorar en un 10 % la precisión de su algoritmo de recomendación de películas existente, Cinematch, medido por el Error Cuadrático Medio (RMSE) en un conjunto de datos de prueba oculto.

Este desafío no solo capturó la imaginación de miles de investigadores y entusiastas de la ciencia de datos en todo el mundo (más de 40,000 equipos de 186 países descargaron los datos), sino que también tuvo un impacto profundo y duradero en el campo de los sistemas de recomendación por varias razones:

El Netflix Prize no solo impulsó la búsqueda de algoritmos más precisos, sino que también tuvo un impacto estructural en la investigación de sistemas de recomendación. La disponibilidad de un dataset masivo democratizó el acceso a problemas realistas, mientras que la competencia misma estimuló una significativa innovación en técnicas como la factorización matricial y el blending. Además, consolidó el RMSE como métrica estándar y fomentó un ambiente de colaboración y apertura que aceleró el progreso colectivo en el campo.

El equipo BellKor (Robert Bell, Yehuda Koren y Chris Volinsky de AT&T Labs) fue uno de los pioneros y dominadores de la competencia, ganando los Progress Prizes de 2007 y 2008. Las contribuciones fundamentales, base de esta tesis, incluyeron un sofisticado modelado de los efectos base (biases), el desarrollo y popularización de la factorización matricial para capturar interacciones latentes, y la crucial incorporación de dinámicas temporales para reflejar la evolución de preferencias y popularidad. Además, demostraron la esencialidad del blending de modelos diversos para alcanzar la máxima precisión predictiva.

Finalmente, en 2009, el equipo "BellKor's Pragmatic Chaos", una fusión de BellKor con los equipos "BigChaosz "Pragmatic Theory", ganó el Gran Premio al superar el umbral del 10 %. Su solución final era un complejo ensamble de cientos de modelos, donde las técnicas que exploraremos en esta tesis jugaban un papel central [24].

El legado del Netflix Prize es innegable: no sólo impulsó la tecnología de recomendación, sino que también demostró el poder de los datos abiertos y la competencia colaborativa

para resolver problemas complejos. La motivación de esta tesis es, en parte, tomar ese legado de innovación y aplicarlo a un nuevo y apasionante desafío en el mundo del fútbol.

1.4. Estructura del resto de la tesis

La presente tesis se organiza de la siguiente manera:

- **Capítulo 2: Marco Teórico y Antecedentes:** Se establecen los fundamentos conceptuales y matemáticos del filtrado colaborativo, se describen los principales enfoques (basados en vecindario y factorización matricial), los métodos de aprendizaje y se introduce la importancia de las dinámicas temporales y el blending.
- **Capítulo 3: Datos y Preprocesamiento:** Se describen en detalle las fuentes de datos y su estructura, las decisiones tomadas en el filtrado y limpieza, la definición de las variables clave (usuario, ítem, rating, tiempo) y la creación de los conjuntos de entrenamiento y test.
- **Capítulo 4: Modelos Implementados:** Se detalla la implementación de cada uno de los métodos de filtrado colaborativo desarrollados, desde los predictores base hasta los modelos más complejos que integran factorización, vecindario y dinámicas temporales, así como las técnicas de blending.
- **Capítulo 5: Evaluación y Resultados:** Se presentan las métricas utilizadas para evaluar el rendimiento de los modelos, se describen los benchmarks empleados para la comparación y se muestran los resultados obtenidos por cada modelo y por las técnicas de blending. Además, se presentan recomendaciones reales de algunos modelos representativos.
- **Capítulo 6: Análisis de resultados y discusión:** Se analizan los resultados obtenidos, comparando e interpretando los modelos implementados y conectando los aspectos técnicos con el dominio de aplicación. También se presentan las limitaciones y sesgos del estudio.
- **Capítulo 7: Conclusiones y Trabajos Futuros:** Se recapitularán los aportes principales de la tesis, se resumirán las conclusiones más relevantes y se plantearán posibles líneas de investigación futura.

2. MARCO TEÓRICO Y ANTECEDENTES

Este capítulo tiene como objetivo establecer los fundamentos conceptuales y matemáticos del filtrado colaborativo (también lo llamaremos CF por sus siglas en inglés, Collaborative Filtering), proporcionando el contexto necesario para comprender los modelos implementados en esta tesis. Se revisarán los principales enfoques de CF, la incorporación de dinámicas temporales y el concepto de blending, contextualizando el influyente trabajo de Yehuda Koren, Robert Bell y Chris Volinsky en estos sistemas y en el marco del Netflix Prize, que sirve como principal inspiración para este estudio.

2.1. Definición de sistemas de recomendación de filtrado colaborativo y clasificación

Un sistema de recomendación es un tipo de sistema de filtrado de información que busca predecir la "calificación" o "preferencia" que un usuario daría a un ítem [21] con el que no ha interactuado previamente (o sobre el cual no tiene una evaluación formada). Estos sistemas se han vuelto cruciales en entornos con mucha carga de información, ayudando a los usuarios a descubrir ítems de su interés.

En particular, los métodos de filtrado colaborativo crean sistemas de recomendación que construyen un modelo a partir del comportamiento pasado de un usuario (ítems previamente consumidos o calificados) y de las decisiones tomadas por otros usuarios ("colaborativos"). El CF se basa en la suposición de que si una persona A tiene la misma opinión que una persona B sobre un tema, es más probable que A tenga la misma opinión que B sobre un tema diferente que sobre una persona elegida al azar.

A su vez, el filtrado colaborativo puede subdividirse principalmente en dos enfoques [23]:

- **Métodos Basados en Memoria (o de Vecindario):** Estos métodos operan sobre la totalidad de la base de datos de interacciones usuario-ítem para realizar predicciones. Se basan en el cálculo de relaciones entre usuarios o entre ítems.
  - **CF Basado en Usuarios (User-based CF):** Identifica usuarios ("vecinos") con historiales de calificación similares al usuario activo. Las predicciones para el usuario activo se generan promediando (generalmente de forma ponderada por la similaridad) los ratings dados por estos vecinos a los ítems no calificados por el usuario activo.
  - **CF Basado en Ítems (Item-based CF):** En lugar de encontrar usuarios vecinos, este enfoque calcula las similaridades entre ítems, basándose en cómo

han sido calificados por los mismos usuarios. Para predecir el rating de un usuario para un ítem, se consideran los ratings que ese usuario ha dado a ítems similares. Los modelos basados en ítems suelen ser más escalables y, en muchos casos, más precisos que los basados en usuarios, especialmente cuando el número de usuarios es mucho mayor que el número de ítems [22, 16].

- **Métodos Basados en Modelos de Factores Latentes:** Estos métodos utilizan los ratings observados para entrenar un modelo paramétrico que luego se utiliza para realizar predicciones. Una de las familias más exitosas de modelos basados en modelos es la **factorización matricial**.

  - **Factorización Matricial (MF):** La idea central es mapear tanto a usuarios como a ítems a un espacio latente conjunto de dimensionalidad f (donde f es típicamente mucho menor que el número de usuarios o ítems). Cada usuario j es representado por un vector $P_j \in \mathbb{R}^f$ y cada ítem e por un vector $Q_e \in \mathbb{R}^f$. La interacción entre el usuario y el ítem, y por lo tanto el rating predicho $\hat{R}_{je}$, se modela como el producto interno de sus vectores latentes: $\hat{R}_{je} \approx Q_e^T P_j$. Los vectores de factores latentes se aprenden minimizando un error de reconstrucción regularizado sobre los ratings conocidos [15]. La popularidad de estos métodos se disparó con el Netflix Prize, donde demostraron ser muy efectivos [10, 19]. En la Sección 2.4 se detalla más sobre la fundamentación teórica de estos métodos.

2.2. Predictores base

Una observación fundamental en los datos de recomendación es que gran parte de la varianza en los ratings se debe a efectos que son independientes de las interacciones específicas entre usuarios e ítems. Estos efectos se conocen como **sesgos** (o *biases*).

- **Media Global (μ):** Es el rating promedio sobre todas las calificaciones en el conjunto de datos. Sirve como el punto de partida más básico para cualquier predicción.

- **Sesgo del Ítem (α_e):** Representa la desviación promedio del ítem e con respecto a la media global. Un α_e positivo indica que el ítem e tiende a recibir ratings más altos que el promedio, y viceversa. Esto puede deberse, por ejemplo, a la popularidad del ítem. En el caso de Netflix, Titanic tendría un sesgo alto, o en caso del fútbol, el Real Madrid.

- **Sesgo del Usuario (β_j):** Representa la desviación promedio del usuario j con respecto a la media global, después de considerar el sesgo del ítem. Un β_j positivo indica que el usuario j tiende a dar ratings más altos que el promedio, incluso a ítems de calidad media, y viceversa. En el caso de Netflix se lo asocia a un usuario que tiende a ser menos crítico. En cambio en el fútbol, esto podría reflejar si un

jugador tiende a tener rendimientos consistentemente por encima o por debajo de lo esperado.

La predicción base b_{je} para el rating del usuario j al ítem e se calcula como: $b_{je} = \mu + \beta_j + \alpha_e$.

Modelar estos sesgos explícitamente es crucial por varias razones [15]: Al capturar estas tendencias globales, quitan ruido a la señal del rating y los modelos más complejos (como la factorización matricial o los modelos de vecindario) pueden enfocarse en modelar las interacciones residuales, que son las verdaderamente personalizadas. Los sesgos también son fáciles de entender y pueden proporcionar interpretaciones directas del modelo (e.g., "este equipo es consistentemente bien valorado", "este jugador tiende a rendir por debajo de la media"). Además, para usuarios o ítems con pocos ratings, los sesgos pueden ser la principal fuente de información para la predicción.

Los sesgos se aprenden minimizando el error cuadrático medio regularizado sobre los ratings conocidos.

2.3. Incorporación de dinámicas temporales a sesgos y parámetros

Las preferencias de los usuarios y las características de los ítems no son estáticas; evolucionan con el tiempo. En el fútbol, esto es evidente: el rendimiento de un jugador fluctúa a lo largo de su carrera (picos de forma, declive por edad, adaptación a nuevas ligas), y el nivel o estilo de un equipo cambia entre temporadas (nuevos entrenadores, fichajes, cambios generacionales). El trabajo de Koren (2009) [14] y la solución BellKor [15] destacaron la importancia crítica de modelar e incorporar estas dinámicas temporales.

- **Sesgos Temporales:**
  - **Sesgo del Ítem Dependiente del Tiempo ($\alpha_e(t)$):** La popularidad o calidad percibida de un ítem puede cambiar. En el fútbol, el nivel de un equipo puede variar significativamente entre temporadas debido a cambios de entrenador, cambios grandes en la plantilla, cambios dirigenciales, etc. Esto se puede modelar permitiendo que el sesgo del ítem α_e varíe con el tiempo. Veremos más adelante que hay varias formas de dividir el tiempo. Una forma es hacerlo en "bins"(e.g., temporadas o grupos de temporadas) y aprender un sesgo $\alpha_{e,bin(t)}$ para cada bin. Otra forma es modelar una función más suave. Koren (2009) [14] propone modelar $\alpha_e(t)$ dividiendo la línea temporal en "bins" y aprendiendo un sesgo constante $\alpha_{e,Bin(t)}$ para cada bin. Así, el sesgo total del ítem en el tiempo t es $\alpha_e(t) = \alpha_e + \alpha_{e,Bin(t)}$, donde α_e es el sesgo estacionario del ítem.
  - **Sesgo del Usuario Dependiente del Tiempo ($\beta_j(t)$):** Los usuarios exhiben cambios temporales en sus tendencias de calificación, ya sea porque descubrieron un nuevo género que les gustó o porque han cambiado algo en su vida que les produjo un cambio en las películas elegidas. Los jugadores también exhiben

cambios temporales en sus tendencias de rendimiento. Esto puede deberse a lesiones, el estilo de juego del jugador que va cambiando o simplemente que va entrenando, mejorando y ganando experiencia. En el caso de Netflix se exploran varias parametrizaciones [14] :

- Una función lineal simple: $\beta_j(t) = \beta_j + \delta_j \cdot dev_j(t)$. El término $dev_j(t) = \text{sign}(t - t_j) \cdot |t - t_j|^\gamma$ modela una desviación gradual del sesgo base del usuario β_j a lo largo del tiempo. t_j es la temporada media de actividad del jugador y γ (usaron 0,4) controla la no linealidad de la función.
- β_{jt}: Para capturar fluctuaciones muy locales y transitorias en el comportamiento del usuario que no siguen una tendencia suave. En el Netflix Prize, se observó que los ratings de un mismo usuario en un mismo día tienden a agruparse, lo que β_{jt} ayuda a modelar.
- $c_j(t)$: Refleja que los usuarios pueden tener diferentes percepciones de la escala de calificación, y esta percepción puede cambiar. La predicción del sesgo del ítem se ajusta por $\alpha_e(t) \cdot c_j(t)$. Similar a $\beta_j(t)$, $c_j(t)$ puede tener un componente estacionario c_j y uno diario c_{jt}.

Koren, Bell y Volinsky [15] también incorporaron un término de sesgo del ítem que depende de la frecuencia con la que el usuario calificó en el día del rating t_{je}, denotado $\alpha_{e,f_{je}}$. La idea es que los ratings dados en días de alta actividad de calificación (muchos ratings) pueden tener características diferentes. En el contexto de esta tesis, dado que generalmente se tiene un rating por jugador-equipo-temporada, este tipo de sesgo de frecuencia diaria no es directamente aplicable y no será implementado.

- **Parámetros de Factores Latentes Dependientes del Tiempo ($P_j(t)$):** No solo los sesgos, sino también las preferencias latentes de un usuario pueden cambiar. El "perfil"de un jugador (representado por su vector de factores latentes P_j) puede evolucionar. Por ejemplo, a medida que un jugador crece en edad y experiencia, puede desarrollar nuevas habilidades, perder habilidades o cambiar su rol en los equipos. Esto se modela permitiendo que P_j sea una función del tiempo, $P_j(t)$, utilizando parametrizaciones similares a las de los sesgos de usuario temporales [15].

Así como mencionamos que los sesgos benefician las predicciones del modelo, la incorporación de estas dinámicas temporales permite a los modelos adaptarse a los cambios y capturar tendencias, lo que también resulta en algunos casos en predicciones más precisas.

2.4. Fundamentos matemáticos relevantes

- **SVD como método de factores latentes y Descomposición Matricial:** Como se mencionó anteriormente, la idea es aproximar la matriz de ratings R (de usuarios

por ítems) como el producto de dos matrices de factores latentes de menor rango, P (usuarios por factores) y Q (ítems por factores): $R \approx PQ^T$. La tarea es encontrar P y Q. Matemáticamente, esto está relacionado con la Descomposición en Valores Singulares (SVD) de una matriz. Si tuviéramos la matriz completa de ratings R (jugadores x equipos), SVD la descompone como $R = U\Sigma V^T$, donde U $(m \times m)$ y V $(n \times n)$ son matrices ortogonales cuyas columnas son los vectores singulares izquierdos y derechos, respectivamente, y Σ $(m \times n)$ es una matriz diagonal con los valores singulares $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ en su diagonal (donde r es el rango de R). La mejor aproximación de R de rango f (en el sentido de mínimos cuadrados de Frobenius) es $R_f = U_f \Sigma_f V_f^T$, donde U_f son las primeras f columnas de U, V_f las primeras f columnas de V, y Σ_f la submatriz diagonal superior izquierda $f \times f$ de Σ. En nuestro caso, P podría interpretarse como $U_f \Sigma_f^{1/2}$ y Q como $V_f \Sigma_f^{1/2}$ (o variantes). Las f dimensiones corresponden a las direcciones de mayor varianza en los datos, capturando así las "tendencias" o "arquetipos" más importantes. El producto escalar $Q_e^T P_j$ mide la alineación entre el perfil latente del jugador y el perfil latente del equipo en este espacio de f dimensiones. En el contexto de factorización de matrices para CF, no calculamos SVD directamente debido a los valores faltantes en la matriz. En su lugar, "buscamos" directamente las matrices de factores P $(m \times f)$ y Q $(n \times f)$ tales que $R \approx PQ^T$. Se puede pensar en P_j (fila j de P) como el vector de coordenadas del jugador j en el espacio latente de f dimensiones, y Q_e (fila e de Q) como las coordenadas del equipo e en ese mismo espacio. Los f ejes de este espacio representan las "características latentes" más importantes que explican la varianza en los ratings. Las dimensiones con valores singulares más altos en una SVD teórica de la matriz completa serían las más "importantes". Al elegir una f pequeña, estamos forzando al modelo a capturar las tendencias más generales y robustas, evitando el ruido. Finalmente, la interacción entre dos vectores (ya sea un jugador y un equipo o ambos jugadores o equipos) están dadas por el producto interno entre ellos.

- **Función de Pérdida:** Dado que la matriz R es muy dispersa (muchos ratings faltantes), no se puede aplicar SVD directamente. En su lugar, se aprenden P y Q minimizando el error cuadrático medio regularizado sobre los ratings conocidos R. Si llamamos $\mathcal{K}$ al conjunto de pares (j, e) cuyo rating conocemos, entonces obtenemos la siguiente función de pérdida: $\min_{P,Q} \sum_{(j,e)\in\mathcal{K}} (R_{je} - Q_e^T P_j)^2 + \lambda(||P_j||^2 + ||Q_e||^2)$ Además, esta es una función diferenciable que facilita la optimización.

- **Regularización:** El término de regularización $\lambda(||P_j||^2 + ||Q_e||^2)$ (regularización L2) en la función de pérdida es crucial para prevenir el sobreajuste en modelos con muchos parámetros (como los factores latentes al elegir dimensión alta o los sesgos para cada usuario/ítem), penalizando magnitudes grandes en los parámetros. Esto "suaviza" el modelo, forzando a que los factores y sesgos solo tomen valores grandes

si hay una fuerte evidencia en los datos. La idea detrás de ello es producir modelos más simples y con mejor capacidad de generalización. La regularización más común es la L2 (o de Tikhonov), que penaliza la suma de los cuadrados de los parámetros: $\lambda \cdot (\sum ||P_j||^2 + \sum ||Q_e||^2 + \sum \beta_j^2 + \sum \alpha_e^2)$. El hiperparámetro λ controla la fuerza de la penalización.

Una justificación teórica interesante de la regularización L2 proviene del contexto Bayesiano. Minimizar el error cuadrático con regularización L2 es equivalente a encontrar el máximo a posteriori (MAP) bajo la suposición de que los errores de rating son Gaussianos y los parámetros $(P_j, Q_e, \beta_j, \alpha_e)$ provienen de una distribución a priori Gaussiana con media cero que refleja la creencia de que la mayoría de los factores y sesgos deberían ser pequeños. Esto favorece soluciones más "simples" o con valores más pequeños, mejorando la generalización [7].

2.5. Algoritmos de aprendizaje

Para aprender los parámetros de los modelos de factorización matricial (y los sesgos), se utilizan principalmente dos algoritmos de optimización:

- **Descenso de Gradiente Estocástico (SGD):**
  El Descenso de Gradiente Estocástico es un método de optimización iterativo ampliamente utilizado para entrenar modelos de aprendizaje automático, especialmente aquellos con grandes cantidades de datos y/o funciones de pérdida no convexas, como es el caso de la Factorización Matricial aplicada en las técnicas de CF. Su popularidad se debe a una combinación de eficiencia computacional y buen rendimiento empírico. En lugar de calcular el gradiente exacto de la función de pérdida sobre todo el conjunto de entrenamiento en cada iteración, SGD estima el gradiente utilizando una única muestra de entrenamiento seleccionada aleatoriamente, o un pequeño subconjunto de muestras (un "mini-batch"). Los parámetros del modelo se actualizan dando un pequeño paso en la dirección opuesta a este gradiente estimado. Matemáticamente, si $L(\theta)$ es la función de pérdida con parámetros θ, y $L_k(\theta)$ es la pérdida para la k-ésima muestra, la actualización es: $\theta \leftarrow \theta - \gamma \cdot \nabla L_k(\theta)$, donde γ es la tasa de aprendizaje. La naturaleza "estocástica"de las actualizaciones (debido al uso de una sola muestra o un mini-batch) introduce ruido en el proceso de optimización. Paradójicamente, este ruido puede ser beneficioso, ya que ayuda al algoritmo a escapar de mínimos locales subóptimos y a encontrar regiones más amplias del espacio de parámetros que generalizan mejor. El entrenamiento se realiza típicamente durante varias épocas, donde una época consiste en un pase completo a través de todas las muestras del conjunto de entrenamiento. El tamaño de batch (número de muestras en un mini-batch) es un hiperparámetro importante; valores comunes oscilan entre decenas y algunos cientos (en nuestro caso 128). Tamaños de batch mayores reducen

la varianza de la estimación del gradiente pero aumentan el costo computacional por actualización. La simplicidad de implementación y la eficiencia en grandes datasets han hecho de SGD el método de elección para muchos problemas de aprendizaje automático, incluyendo su uso extensivo en el Netflix Prize.

- **Mínimos Cuadrados Alternados (ALS - Alternating Least Squares):** Aunque la función de pérdida de la factorización matricial no es convexa conjuntamente en P y Q, sí es convexa si se fija una de las matrices y se optimiza la otra. ALS explota esto de la siguiente manera: primero inicializa Q (e.g., aleatoriamente). Luego, fija Q y resuelve para P minimizando la función de pérdida regularizada. Esto es un problema de mínimos cuadrados estándar para cada fila de P (cada P_j). Además, fija P y resuelve para Q de manera análoga. Repite los pasos 2 y 3 hasta la convergencia.

  ALS puede ser más estable que SGD y es paralelizable. Es particularmente útil para sistemas con feedback implícito donde SGD es menos eficiente [12]. Sin embargo, en este caso no contamos con feedback implícito por lo que veremos que ambos métodos obtienen resultados similares. En general se optó por usar SGD por simplicidad.

Ambos métodos fueron utilizados por el equipo ganador del Netflix Prize, con SGD siendo a menudo preferido para los modelos de feedback explícito por su simpleza y velocidad en datasets grandes.

2.6. Métodos de blending de modelos

Ningún modelo individual es perfecto; cada uno tiene sus propias fortalezas y debilidades y tiende a cometer diferentes tipos de errores. El **blending**[1] es una técnica de ensamble que busca combinar las predicciones de múltiples modelos diversos para producir una predicción final que sea más precisa y robusta que la de cualquiera de los modelos componentes [25].

La motivación es que si diferentes modelos capturan diferentes aspectos de los datos o cometen errores no correlacionados, un meta-modelo puede aprender a ponderar sus predicciones de manera óptima debido a que diferentes modelos cometen errores en diferentes instancias [25, 8, 9]. En el contexto del Netflix Prize, el blending fue absolutamente esencial para alcanzar los niveles más altos de precisión [5, 24].

El proceso involucra entrenar una variedad de modelos base (e.g., diferentes parametrizaciones de SVD, modelos de vecindario, etc.) y generar predicciones "out-of-sample"de estos modelos base sobre un conjunto de validación (evitando que se generen sobre los

[1] Aunque se usa el término genérico "blending", el término más preciso y técnico sería "stacked generalization"

mismos datos con los que se entrenaron los modelos base). Luego, se utilizan estas predicciones como características de entrada para entrenar un meta-modelo (o blender). El meta-modelo aprende los pesos óptimos para combinar las predicciones de los modelos base. Para hacer una predicción final sobre datos nuevos (test set), se obtienen las predicciones de todos los modelos base y se introducen en el meta-modelo entrenado.

Los meta-modelos pueden ser desde una regresión lineal hasta algoritmos más complejos como Gradient Boosted Decision Trees (GBDT), Redes Neuronales o Support Vector Machines. El equipo ganador del premio de Netflix utilizó GBDT como blender. En este trabajo se estudió ese mismo modelo, así como también el caso de la regresión lineal y el método de Stacking de modelos.

3. DATOS Y PREPROCESAMIENTO

La calidad y adecuación de los datos son fundamentales para el éxito de cualquier sistema de recomendación. En esta sección, se detallan las fuentes de datos utilizadas, el proceso de selección y las etapas de preprocesamiento llevadas a cabo para construir los conjuntos de entrenamiento y evaluación.

Para esta tesis se trabajó con un conjunto de datos que, si bien no es el más extenso disponible en el dominio del fútbol profesional, fue seleccionado por su calidad y confiabilidad percibida en los ratings que ofrece, priorizando la precisión sobre la cantidad de observaciones y asumiendo que ratings más confiables conducirán a modelos de recomendación más robustos. Los ratings son justamente el target o variable a predecir, es decir los valores de la matriz R de jugadores por equipos que tenemos y los faltantes. Recordemos que estos se componen del promedio de rendimiento de los jugadores en sus partidos por temporada.

El desempeño de los jugadores en cada partido es un valor que no sabemos exactamente como fue calculado. Sin embargo, este conjunto de datos fue obtenido de la plataforma BeSoccer, una fuente que el club Racing de Santander considera confiable.

Este dataset incluye alrededor de 180.000 registros de jugadores actualmente activos, con la siguiente información: identificadores únicos, nombre, club, temporada, rating promedio, estadísticas de rendimiento y edad. Para todos los jugadores presentes, contiene todos los clubes por los que jugó en su carrera, además de su selección nacional, en caso de haber jugado en la misma. La densidad en este caso de la matriz R de jugadores por equipos es de **0,16 %** aproximadamente (cerca de 16 mil jugadores y 7 mil equipos).

3.1. Filtrado de observaciones y densificación

El preprocesamiento incluyó una etapa de agregación temporal y estandarización. Dado que un jugador puede participar en múltiples temporadas con el mismo club, se decidió unificar estas participaciones, colapsando la información a nivel jugador-equipo. Esto se hizo para simplificar el problema a una única matriz R de jugadores por equipos utilizando la temporada como dato ya que de lo contrario tendríamos una R por temporada. Para lograr esto, se sumaron las estadísticas aditivas (partidos jugados, goles, asistencias, minutos jugados), y se promediaron los ratings y los minutos por partido. En particular, se optó por no ponderar los ratings por cantidad de partidos jugados sino simplemente por temporada, ya que para este estudio el desempeño por temporada es lo que queremos predecir y por lo tanto es más importante que el desempeño por partido. Además, se homogeneizó la representación de las temporadas, quedándonos únicamente con el segundo año indicado (por ejemplo, "2024/25"se representa como "2025") para simplificar

el tratamiento temporal y asegurar consistencia en los modelos con dinámica temporal.

Una vez estandarizados los datos, se procedió a un filtrado por completitud y relevancia. Se eliminaron las observaciones que no contaban con valores válidos en el año o en el rating (incluyendo aquellos con rating igual a cero). Adicionalmente, se excluyeron aquellas instancias donde el jugador no alcanzaba los **400 minutos jugados** en la temporada con ese equipo. Este umbral, que se aproxima a la participación en cinco partidos completos, se estableció con el fin de asegurar la calidad y representatividad de los datos utilizados. Esta decisión se fundamenta en la necesidad de contar con una muestra de rendimiento individual lo suficientemente amplia como para que las métricas calculadas no estén excesivamente dominadas por la variabilidad inherente a un número muy reducido de partidos. Se busca así un equilibrio entre mantener un número significativo de jugadores en el análisis y garantizar que las estadísticas de rendimiento sean representativas del desempeño real del jugador. Estudios previos en el análisis del rendimiento en fútbol, como los de Mendez-Domenech et al. (2024) [18] y Mara-Empinotti et al. (2024) [17], han adoptado umbrales de participación similares, aunque con variaciones, para asegurar la robustez de sus análisis. La elección de 400 minutos en esta tesis representa un valor intermedio entre estos enfoques. Tras este filtrado, La densidad de la matriz R de jugadores por equipos es de **0,25 %** aproximadamente (65 mil registros provenientes de cerca de 4500 jugadores y 5800 equipos).

La capacidad de estos métodos para detectar patrones útiles es directamente proporcional a la cantidad de interacciones observadas de cada sujeto del modelo. Por este motivo, y con el objetivo primordial de maximizar la efectividad de los modelos, se buscó densificar la matriz jugador-equipo. Para hacerlo, se restringió el análisis a jugadores que hubieran participado en al menos cinco equipos (incluídas selecciones nacionales) distintos a lo largo de su carrera registrada en el dataset, y, subsecuentemente, a equipos que hubieran contado con el paso de al menos cinco jugadores distintos que cumplieran el criterio anterior. Si bien esta operación implica un inevitable trade-off, con una potencial pérdida de información sobre jugadores con trayectorias menos variadas o equipos con menor rotación, además de la pérdida de datos, permite trabajar sobre una submatriz de interacciones con una conectividad significativamente mayor y una menor dispersión. Esto es crucial para que los modelos basados en interacciones, como los de factorización matricial y vecindario, puedan aprender patrones de manera más efectiva [1].

Los efectos de este proceso de filtrado y densificación pueden visualizarse en los gráficos que muestran la distribución de la cantidad de equipos por jugador (Figuras 3.1 y 3.2) y la cantidad de jugadores por equipo (Figuras 3.3 y 3.4), antes y después de aplicar estos criterios. Esta etapa fue fundamental para garantizar que los métodos de factorización y vecindario pudieran operar sobre un grafo de interacciones suficientemente interconectado.

Tras la aplicación de los procesos de estandarización, filtrado y densificación sobre el dataset BeSoccer, el conjunto final de trabajo quedó conformado por aproximadamente

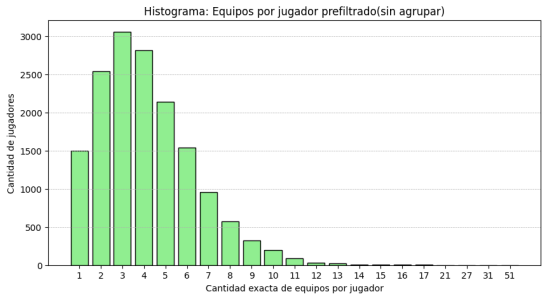

Fig. 3.1: Distribución de cantidad de equipos por jugador (Dataset BeSoccer) ANTES del filtrado.

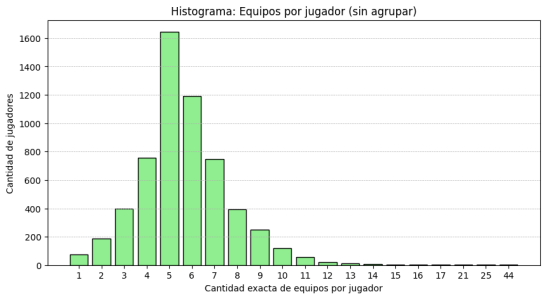

Fig. 3.2: Distribución de cantidad de equipos por jugador (Dataset BeSoccer) DESPUÉS del filtrado.

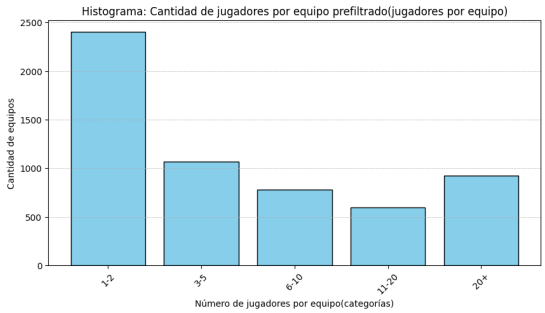

Fig. 3.3: Distribución de cantidad de jugadores por equipo (Dataset BeSoccer) ANTES del filtrado.

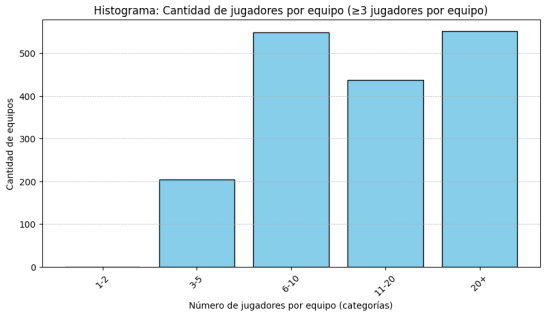

Fig. 3.4: Distribución de cantidad de jugadores por equipo (Dataset BeSoccer) DESPUÉS del filtrado.

33.000 registros. Estos corresponden a cerca de 5.800 jugadores únicos y 1.700 equipos únicos. Con estas cifras, la densidad de la matriz jugador-equipo resultante es de aproximadamente un **0,33 %**. Si bien este es un valor bajo, indicativo de la baja cantidad de datos disponible y la alta dispersión inherente al dominio, es comparable a la densidad encontrada en otros contextos de sistemas de recomendación a gran escala, como el famoso dataset del Netflix Prize, donde la matriz de interacciones usuario-película tenía una densidad cercana al 1 % [5].

3.2. Conjunto final de entrenamiento y test

Para la evaluación final del rendimiento de los modelos, se procedió a dividir este conjunto de datos procesado. Se separó un conjunto de test compuesto por poco más de **3.000 registros**, correspondientes a las interacciones de la última temporada disponible en el dataset (considerada como 2025 para los fines de este estudio). La selección de estas instancias se realizó de manera aleatoria dentro de dicha temporada. Crucialmente, esta partición se realizó *antes* de iniciar cualquier proceso de entrenamiento de los modelos y se mantuvo estrictamente la cronología, asegurando que el conjunto de test representara datos temporalmente posteriores al conjunto de entrenamiento. Esta práctica es estándar para evitar el "data leakeage" o fuga de información del futuro hacia el pasado, lo que invalidaría la evaluación de la capacidad predictiva real de los modelos [11].

Adicionalmente, se realizó un ajuste final al conjunto de test: se excluyeron aquellos jugadores (menos de 50 casos) que, tras la partición, no tenían ninguna observación remanente en el conjunto de entrenamiento. Esta medida es necesaria ya que los modelos de filtrado colaborativo aquí estudiados no pueden, por su naturaleza, realizar predicciones para jugadores o equipos completamente nuevos sobre los que no tienen ninguna información previa (un problema conocido como "cold start"para usuarios/ítems nuevos). De esta manera, se definieron de forma final los conjuntos de entrenamiento y test, con los datos ya filtrados y listos para la implementación y evaluación de los modelos de recomendación.

Las variables fundamentales que utilizan los modelos de recomendación se definieron de la siguiente manera, manteniendo la notación establecida:

- **Usuario (j):** Corresponde al ID único del jugador.
- **Ítem (e):** Corresponde club en el que jugó.
- **Rating (R_{je}):** Representa la calificación promedio de rendimiento del jugador j en el equipo e.
- **Tiempo (t):** Corresponde al año final de la temporada (e.g., 2025), utilizado para incorporar y modelar las dinámicas temporales.

Con estas definiciones, y siguiendo la notación de la literatura [15, 6], R_{je} denota el rating conocido del jugador j en el equipo e, mientras que $\hat{R}_{je}$ (o $\hat{R}_{je}(t)$ si depende del tiempo) se refiere a la predicción generada por el modelo. $\mathcal{K}$ será el conjunto de tuplas (j, e, t) observadas en el entrenamiento. $\mathcal{R}(j)$ denotará el conjunto de equipos por los que ha pasado el jugador j, y $\mathcal{R}(e)$ el conjunto de jugadores que han formado parte del equipo e.

La preparación y el preprocesamiento cuidadoso de estos datos constituyen la base indispensable sobre la cual se edificaron y evaluaron los modelos de recomendación desarrollados en esta tesis.

4. MODELOS IMPLEMENTADOS

Para comenzar esta sección, vale la pena aclarar que el código de los algoritmos utilizados fue implementado de forma personalizada en Python, utilizando la librería PyTorch para el entrenamiento de los modelos mediante descenso de gradiente estocástico. La decisión de desarrollar implementaciones propias, en lugar de utilizar librerías existentes como Surprise (Python) o RecommenderLab (R), se tomó con el objetivo de obtener mayor flexibilidad en la adaptación de los modelos y en la experimentación con variantes específicas, evitando las limitaciones o dependencias de versiones de librerías preexistentes. Además, el nivel de detalle proporcionado en los trabajos [15, 6] facilita la replicación de sus métodos. El código fuente se encuentra disponible en *(https://github.com/pablogroisman/Adaptacin-de-t-cnicas-de-filtrado-colaborativo-para-el-dominio-del-f-tbol)*.

PyTorch es una librería de Python para computación numérica que utiliza tensores (arrays multidimensionales) como su estructura de datos fundamental. Ofrece un robusto soporte para operaciones en GPU, lo que acelera significativamente el entrenamiento de modelos de aprendizaje profundo. Los DataLoaders de PyTorch son utilidades que facilitan la carga eficiente de datos en batches durante el entrenamiento, manejando aspectos como el muestreo, la paralelización y la reproducción de los datos.

La metodología seguida para la implementación de los métodos fue incremental, similar a la descrita en los artículos [15, 6] . Se comenzó con los modelos más simples y se fueron incorporando progresivamente variables y técnicas de filtrado colaborativo para llegar a los modelos más complejos. Este enfoque permite analizar el aporte incremental de cada componente tanto en la performance predictiva como en la naturaleza de las recomendaciones generadas. Se buscó cubrir los modelos propuestos por los ganadores del Gran Premio de Netflix que se consideraron más adaptables al dominio del fútbol, y se propusieron adaptaciones y modelos adicionales cuando se creyó pertinente.

Finalmente, se seleccionó un conjunto de modelos, buscando no solo los mejores predictores individuales sino también una diversidad de técnicas, para conformar el ensamble final mediante blending. Para este último paso, las predicciones de los modelos individuales sobre el conjunto de test se almacenaron para ser utilizadas como features de un nuevo modelo de aprendizaje.

Todas las variables latentes (embeddings) y sesgos se inicializaron siguiendo una distribución normal con media 0 y una desviación estándar menor a 1 y variable según el caso (e.g., 0.01 o $1/\sqrt{\text{dimensión\_embedding}}$). Se utilizó el optimizador Adam con un learning rate inicial de 0.01 (obtenido como óptimo mediante cross validation), y se aplicó regularización L2 a todas las variables entrenables para prevenir el sobreajuste. Aunque Koren y Bell proponen una actualización de parámetros específica en su descenso por gradiente,

esta técnica fue probada y utilizada en algunos modelos pero finalmente se optó por la implementación estándar de Adam en PyTorch por su eficiencia y resultados comparables en pruebas preliminares.

A continuación, se detalla la implementación de cada uno de los modelos.

4.1. Predictores base

Como vimos anteriormente en la Sección 2.2, los predictores base, también conocidos como modelos de sesgos, capturan las tendencias generales en los datos que no dependen de las interacciones específicas entre jugadores y equipos. Estos sesgos representan, por ejemplo, la popularidad o calidad intrínseca promedio de un equipo, independientemente de un jugador particular, y la tendencia general de un jugador a recibir ratings altos o bajos, independientemente del equipo. Modelar estos efectos es crucial, ya que explican una porción significativa de la varianza en los ratings y permiten que los modelos más complejos se enfoquen en capturar las interacciones residuales más sutiles [15]. En todos los casos, la función de pérdida minimizada es el error cuadrático medio (MSE) regularizado de los sesgos. Además, la predicción $\hat{R}_{je}$ se construye sumando la media global μ y los sesgos aprendidos.

4.1.1. Modelo base SIN sesgos temporales

Este es el predictor más fundamental. Predice el rating $\hat{R}_{je}$ como la suma de la media global de los ratings μ, el sesgo del jugador β_j (tendencia del jugador j a tener ratings altos/bajos) y el sesgo del equipo α_e (tendencia del equipo e a recibir ratings altos/bajos de los jugadores que pasan por él):

$$\hat{R}_{je} = \mu + \beta_j + \alpha_e$$

Aunque no utiliza la técnica de filtrado colaborativo como tal, este modelo es fundamental porque los sesgos β_j y α_e son componentes esenciales de modelos más avanzados que sí la utilizan. Para este modelo se utilizó una regularización L2 de 0.01 para ambos vectores de sesgos (β_j y α_e) y se entrenó durante 10 épocas. Este modelo fue incluido en el blending final como `pred_baseline_sin_temp`.

4.1.2. Modelo base CON sesgos temporales (por temporada)

Este modelo extiende el anterior incorporando sesgos que varían con el tiempo (temporada). La idea es que tanto la "calidad" o rendimiento promedio de un jugador, como la "popularidad" o nivel de un equipo, pueden cambiar a lo largo de las temporadas. Se introducen β_{jt} (sesgo del jugador j en la temporada t) y α_{et} (sesgo del equipo e en la

temporada t). La predicción es:

$$\hat{R}_{je}(t) = \mu + \beta_j(t) + \alpha_e(t)$$

donde $\beta_j(t) = \beta_j + \beta_{jt}$ y $\alpha_e(t) = \alpha_e + \alpha_{et}$. Como se dispone de pocos datos por temporada para muchos jugadores/equipos, la inclusión de β_{jt} y α_{et} no busca predecir ratings futuros con alta granularidad temporal, sino capturar fluctuaciones o ruido específico de una temporada, permitiendo que β_j y α_e capturen tendencias de sesgo más estables y de largo plazo [14]. Se utilizó una regularización L2 de 0.01 para todos. Este modelo fue incluido en el blending final como `pred_baseline_con_temp`.

4.1.3. Modelo base CON sesgos temporales (por Bins)

Este modelo es una modificación leve del anterior. En este caso, el tiempo no va a representar temporadas únicas sino varias temporadas unidas. Esta técnica fue mencionada anteriormente como Binning o uso de Bins. De esta manera agrupamos temporadas de a 3, para tener más datos por unidad de tiempo. Los bins se agruparon iniciando en la última temporada (ejemplo: 2023, 2024 y 2025 representan el último Bin). Se utilizó una regularización L2 de 0.01 para todos los vectores de sesgos y se entrenó durante 10 épocas. Este predictor base se utilizó como base de otros modelos pero no fue incluido en el blending final por su alto error con respecto al modelo anterior, muy similar.

4.1.4. Modelo base con más variables (adaptado de TimeSVD++)

Inspirado en las extensiones temporales de Koren (2009) [14], se exploró un modelo de sesgos más complejo. Para los sesgos de equipo, se agruparon las temporadas en "bins" como en el modelo anterior (bloques de 3 temporadas), asumiendo que el sesgo de un equipo ($\alpha_{e,bin(t)}$) podría ser constante dentro de esos períodos. Se estima que esto podría capturar el sesgo del paso de un técnico por el club o simplemente una buena camada de jugadores que duró algunas temporadas. Para los sesgos de jugador, se modeló la siguiente función ($\delta_j \cdot dev_j(t)$) además del sesgo base y el sesgo por temporada, resultando en $\beta_j(t) = \beta_j + \delta_j \cdot dev_j(t) + \beta_{jt}$. La función $dev_j(t) = \text{sign}(t - t_j) \cdot |t - t_j|^\gamma$ busca capturar el crecimiento en la carrera del jugador, donde t_j es la temporada media de actividad del jugador j y γ es un hiperparámetro que le quita linealidad a la función ([14] usa $\gamma = 0{,}4$ por lo que se tomó ese valor). Además, δ es un parámetro que aprende el modelo y busca capturar la pendiente de ese crecimiento. Probablemente esta función no sea lo más acorde para modelar la variación del desempeño de un jugador de fútbol, pero se tomó la decisión de mantener la estructura que propusieron en el artículo de Netflix debido a que es una función que introduce únicamente un parámetro extra. Además, no está claro cuál sería una función acorde en general, ya que al estar estudiando el caso de jugadores activos, hay algunos más veteranos y otros más jovenes, lo que dificulta encontrar un modelado acorde,

sumado a que los jugadores tienen muchos altibajos en su carrera y se ven funciones muy distintas al graficar los distintos desempeños de los mismos en función del tiempo.

Otros parámetros no fueron tomados en cuenta por la hipótesis de que no se aplicaban al área. Estos fueron la incorporación de frecuencia de rating, que en nuestro caso es siempre uno por temporada excepto que el jugador se haya quedado en el club; el otro caso corresponde al de agregar un sesgo por escala relativa de cada usuario. Se decidió no tomarlo en cuenta ya que el rating lo hace siempre la misma página en nuestro caso, por lo que la escala debería ser similar para todo jugador y equipo (más allá de los sesgos particulares de los mismos). La predicción resultante es:

$$\hat{R}_{je}(t) = \mu + (\beta_j + \alpha_j \cdot dev_j(t) + \beta_{jt}) + (\alpha_e + \alpha_{e,bin(t)})$$

Se utilizó regularización L2 de 0.01 y 10 épocas.

Este modelo fue agregado al blending final con el nombre de `pred_baseline_TimeSVD++` y fue utilizado como predictor base para otros modelos más complejos.

4.2. Modelos de Vecindario

Los modelos de vecindario (o *neighborhood models*) se basan en la idea de que el comportamiento o las preferencias pasadas de los usuarios pueden utilizarse para inferir gustos futuros. En el contexto de recomendación, esto se traduce en encontrar "vecinos"(ítems similares o usuarios similares) para realizar predicciones.

4.2.1. Item-based Similarity model

Este modelo, tomado de [13], se destaca por su simpleza y porque no depende de parámetros o variables que se entrenan sino que simplemente depende de una función de similaridad. Su enfoque se basa en predecir el rating de un jugador j para un equipo e basándose en cómo ese jugador j ha sido calificado en equipos e' similares a e. La similaridad $s_{ee'}$ entre dos equipos e y e' se calcula típicamente usando la correlación de Pearson entre los vectores de ratings recibidos por e y e' de los jugadores que han pasado por ambos. La predicción se ajusta restando los sesgos del jugador y equipo del rating conocido $R_{je'}$ y pesando por la similaridad entre los equipos. Por último, se suman los sesgos de jugador y equipo que se buscan predecir. En este caso, los sesgos no son aprendidos mediante descenso de gradiente sino que son la diferencia entre la media del equipo o jugador a la media global. De este modo la predicción del modelo es la siguiente:

$$\hat{R}_{je} = \beta_j + \alpha_e + \frac{\sum_{e' \in S^k(e;j)} s_{ee'}(R_{je'} - b_{je'})}{\sum_{e' \in S^k(e;j)} s_{ee'}}$$

donde $S^k(e; j)$ es el conjunto de los k equipos más similares a e por los que el jugador j ha pasado, y b_{je} es el predictor base $\mu + \beta_j + \alpha_e$. En nuestro caso, debido a la poca cantidad de datos, se decidió utilizar k como lo máximo posible, es decir, utilizar todos los datos que correspondan y pesarlos por la similaridad. El tercer término vale 0 cuando la similaridad entre un equipo y todos los demás equipos es 0. Justamente debido a esto último es que [13] critica estos modelos, sumado a su poca justificación formal.

Este modelo fue agregado al blend con el nombre de `pred_similaridad`. Su alto RMSE sugiere que la simple correlación de Pearson y una estructura de predicción básica no son suficientes en este dominio, pero podría aportar diversidad al ensamble.

4.2.2. Modelo de Vecindario con Pesos Entrenados

Este es un modelo más avanzado donde las "similaridades"no son precalculadas, sino que son parámetros entrenables w_{ef} y c_{ef}. w_{ef} captura la influencia del rating (desviado de su base) del equipo f en la predicción para el equipo e, mientras que c_{ef} captura una influencia directa (similar a un sesgo de interacción). La predicción para el jugador j en el equipo e se forma considerando todos los equipos f por los que el jugador j ha pasado:

$$\hat{R}_{je} = \mu + \beta_j + \alpha_e + |\mathcal{R}(j)|^{-1/2} \sum_{f \in \mathcal{R}(j)} (R_{jf} - \beta_{jf}) w_{ef} + |\mathcal{R}(j)|^{-1/2} \sum_{f \in \mathcal{N}(j)} c_{ef}$$

En este caso, se puede considerar que w_{ef} representa el peso del paso de un jugador en el equipo e sobre el equipo f, dividido por la diferencia de la media de e a la media global. Como este parámetro depende tanto de esa media, este modelo le suma un término absoluto c_{ef} que representa un valor absoluto que quita o agrega valor al rating. La normalización por $|\mathcal{R}(j)|^{-1/2}$, tomado de [13] al igual que todo el modelo, es una forma de manejar la varianza en el número de equipos por los que pasa un jugador. Se utilizó un Learning Rate de 0.01, inicialización normal (0, 0.01), regularización L2 de 0.02 y 10 épocas.

Este modelo fue agregado al blend con el nombre de `pred_item_vector_model`. Además, se agregó otro modelo sin la normalización $|\mathcal{R}(j)|^{-1/2}$ llamado `pred_item_vector_model_simplified`.

4.3. Modelos de Factores Latentes

Los modelos de factores latentes buscan descubrir características subyacentes tanto de jugadores como de equipos. La idea es que los ratings pueden ser explicados por cómo los factores de un jugador se alinean con los factores de un equipo. Estos modelos suelen ser muy efectivos para capturar patrones generales en los datos y pueden encontrar relaciones interesantes en sus distintas dimensiones.

4.3.1. SVD Simple

El modelo más básico de factorización matricial predice el rating $\hat{R}_{je}$ como el producto interno de un vector de factores latentes del jugador P_j y un vector de factores latentes del equipo Q_e:

$$\hat{R}_{je} = Q_e^T P_j$$

Este modelo se utilizó principalmente para estudiar la dependencia del error con respecto a la dimensión de los embeddings, que se presentará en la Sección 5.3. Aunque conceptualmente simple, su performance es limitada por la ausencia de sesgos. Lo interesante de estos modelos es que si bien su error es alto, sus predicciones son informativas ya que concentran toda la información únicamente en esos vectores que son lo único que compone a los ratings predichos. Para este modelo se utilizó dimensión de embedding 20. Esto se explica mejor en la sección de resultados pero la decisión radica en la curva de error y en que es el único parámetro del modelo. Más adelante se utilizan como dimensiones 10, en modelos más simples y 3 en más complejos.

Este modelo se incluyó en el blend final con el nombre de `pred_basicSVD_0`.

4.3.2. SVD Básico

Este modelo es una ampliación directa del anterior, incorporando los predictores base (sesgos estáticos):

$$\hat{R}_{je} = \mu + \beta_j + \alpha_e + Q_e^T P_j$$

Este es el modelo de factorización matricial más comúnmente referido en la literatura inicial de los ganadores del Netflix Prize [10, 6]. Siguiendo su metodología, y al igual que en todos los modelos siguientes, se entrenaron primero los sesgos y luego los embeddings, teniendo en cuenta los sesgos ya entrenados. Se usó regularización L2 de 0.02 para todos los parámetros, learning rate de 0.01 y una dimensión de embedding de 10 (se explicará la elección de la dimensión más adelante en la Sección 5.3). Fue agregado al blend con el nombre de `pred_basicSVD_v1`.

Sin embargo, como se verá más adelante, a pesar de su bajo error, los sesgos de este modelo explican la mayoría del rating, mientras que los embeddings modifican muy poco los mismos. Esto genera recomendaciones de jugadores muy similares para equipos con sesgos similares. Por este motivo, Se probó aumentar más la regularización de los sesgos pero esto simplemente llevó a reducir estos parámetros sin aumentar el efecto de los factores latentes y acercando los valores de los ratings a la media. Lo que sí funcionó para modificar esas estadísticas fue disminuir la regularización de los factores latentes (a 0.0025). Esto aumentó significativamente el error pero sus recomendaciones aportan más información a la investigación.

Esta segunda versión del modelo también fue agregada al blending con el nombre de `pred_basicSVD v2`.

4.3.3. SVD con Biases Temporales

Este modelo es similar al anterior, pero incorpora los predictores base con sesgos temporales básicos (Sección 4.1.2):

$$\hat{R}_{je}(t) = \mu + \beta_j(t) + \alpha_e(t) + Q_e^T P_j$$

Los hiperparámetros fueron los mismos, exceptuando la regularización del sesgo temporal del jugador que se aumentó a 0.1 para controlar el sobreajuste de este parámetro.

Fue agregado al blend con el nombre de `pred_basic_SVD_2`.

4.4. Modelos que Incorporan Ambos Métodos (Factorización y Vecindario)

Estos modelos buscan combinar la capacidad de los modelos de factores latentes para capturar señales globales con la habilidad de los modelos de vecindario para modelar interacciones más locales o explícitas.

4.4.1. SVD++

El modelo SVD++ original [13] aumenta la representación del usuario (jugador) sumando los factores latentes de los ítems (equipos) con los que ha interactuado, representando así una forma de contexto de vecindario. La predicción resultante del modelo es:

$$\hat{R}_{je}(t) = \mu + \beta_j(t) + \alpha_e(t) + Q_e^T(P_j + |\mathcal{R}(j)|^{-1/2} \sum_{k \in \mathcal{R}(j)} Y_k)$$

Donde Y_k es otro vector de factores latentes para el equipo k. Los sesgos $\beta_j(t)$ y $\alpha_e(t)$ son los sesgos temporales básicos. De esta manera se aprende mediante entrenamiento un vector extra por equipo que incorpora la idea de agregarle a las predicciones, variables más explícitas que tengan en cuenta las interacciones entre equipos, idea que habíamos visto aplicada de manera distinta en el `pred_item_vector_model`. Además, se tiene en cuenta el factor $|\mathcal{R}(j)|^{-1/2}$ que regula la participación del vector Y_k de cada equipo considerando la cantidad de equipos por los que pasó el jugador. Es decir que a más equipos jugados, menos influye cada uno en particular. En cuanto a los hiperparámetros, la regularización para embeddings fue de 0.02, para sesgos base 0.02 y para el sesgo temporal de jugador 0.1 para controlar el sobreajuste.

Fue agregado al blend con el nombre de `pred_SVDpp`. Además, se agregó el mismo modelo sin la normalización $|\mathcal{R}(j)|^{-1/2}$. Este modelo se llama `pred_SVDpp_simple`.

4.4.2. TimeSVD++

Este modelo es una extensión de SVD++ que incorpora dinámicas temporales no sólo en los sesgos, sino también en los factores latentes del jugador $P_j(t)$. La idea es que el "perfil latente"de un jugador puede cambiar con el tiempo (e.g., evolución de su estilo de juego, madurez, velocidad, experiencia) y modificando de esta forma sus características, además posibles sesgos de rating. Siguiendo la lógica de los sesgos de jugador en el predictor base más complejo (Sección 4.1.4), se decidió seguir con la idea de separar las temporadas en 3 bins, buscando captar esas diferencias con más de un dato por jugador. $P_j(t) = P_j + \delta_j \cdot dev_j(t) + P_{jt}$.

Al igual que en el caso de las películas de Netflix, se considera que los clubes no cambian tanto su ideología de juego y su estructura, por lo que en este caso alcanzan los sesgos temporales para modelar cambios de plantilla o momentos buenos del club. Por lo tanto, no se incorporan estas dinámicas temporales en sus vectores Q_e ni Y_k. De este modo, la predicción completa sería, utilizando el predictor base más complejo (Sección 4.1.4):

$$\hat{R}_{je}(t) = (\mu + \beta_j + \delta_j \cdot dev_j(t) + \beta_{jt} + \alpha_e + \alpha_{e,bin(t)}) + Q_e^T(P_j(t) + |\mathcal{R}(j)|^{-1/2} \sum_{k \in \mathcal{R}(j)} Y_k)$$

Este modelo es una adaptación del modelo TimeSVD++. Se utilizó una regularización para embeddings de 0.002, y para los sesgos de 0.02.

Fue agregado al blend con el nombre de `pred_timeSVDpp`.

4.4.3. TimeSVD++ Simplificado

Una variante del anterior, donde los factores del jugador $P_j(t)$ son dinámicos en el tiempo, pero los sesgos β_j y α_e son estáticos (modelo base sin sesgos temporales, Sección 4.1.1). Esto busca aislar el efecto de la temporalidad únicamente en los embeddings de los jugadores, lo que nos lleva a la siguiente predicción:

$$\hat{R}_{je}(t) = (\mu + \beta_j + \alpha_e) + Q_e^T(P_j(t) + |\mathcal{R}(j)|^{-1/2} \sum_{k \in \mathcal{R}(j)} Y_k)$$

Se utilizaron los mismos valores de hiperparámetros que en el caso anterior.

Fue agregado al blend con el nombre de `pred_timeSVDpp_simplified`.

4.5. Blending

El blending es una técnica de ensamble donde las predicciones de múltiples modelos diversos (modelos base o de nivel 0) se utilizan como características de entrada para un meta-modelo (o modelo de nivel 1) que aprende a combinarlas óptimamente para producir la predicción final [25]. Koren y Bell [15] destacaron su importancia para alcanzar el

máximo rendimiento en el Gran Premio de Netflix.

Para implementar el blending, el conjunto de test original se dividió: un 90 % se utilizó para entrenar el meta-modelo y el 10 % restante para evaluarlo. Esta separación interna del conjunto de test es para asegurar que el meta-modelo se entrene con predicciones "out-of-sample"de los modelos base (que fueron entrenados con el conjunto de entrenamiento), evitando así el sobreajuste y una evaluación sesgada de su capacidad de generalización.

Se exploraron tres conjuntos de características para el meta-modelo: Primero, se estudiaron los modelos a través de las features clásicas con las que contamos en el dataset. Estas son estadísticas descriptivas básicas del jugador y el equipo en la temporada (goles, asistencias, minutos jugados, partidos jugados, temporada) y no provienen ni se utilizan en los modelos de filtrado colaborativo. Luego utilizamos las predicciones de modelos. Es decir, se usaron como features las salidas (predicciones de rating) de los modelos de filtrado colaborativo descritos anteriormente. Por último, se evaluó un conjunto de features combinadas. Esta es la unión de los dos conjuntos anteriores.

Además, se evaluaron tres algoritmos de blending como meta-modelos: primero, una Regresión Lineal, Un modelo simple y robusto que aprende una combinación lineal ponderada de las características de entrada. Es muy útil ya que suele ser rápido de entrenar e interpretar. Luego se probó con Gradient Boosted Decision Trees (GBDT), un potente algoritmo de ensamble que construye árboles de decisión secuencialmente, donde cada nuevo árbol corrige los errores de los anteriores [9]. Fue el meta-modelo principal utilizado por los ganadores del premio de Netflix y es capaz de capturar interacciones no lineales complejas entre las predicciones de los modelos base. Por último, se estudió la capacidad del Stacking con Árboles, un método similar a GBDT en el uso de árboles, pero en este caso se entrenan en paralelo y se aprende a combinar las predicciones. [25]

4.6. Hiperparámetros

Respecto a los hiperparámetros, se tomaron como referencia inicial los valores propuestos por los ganadores del premio de Netflix. Sin embargo, se estudió la regularización, el learning rate y la dimensión del embedding para el caso del modelo `pred_basicSVD` (como se detalla en la Sección 5.3). Además, en algunos casos se realizaron modificaciones de regularizaciones sobre parámetros específicos que generaban sobreajuste, tomando valores muy altos y explicando la mayor parte del rating predicho. En los casos de los parámetros de jugadores, esto se podía ver en las predicciones, ya que les recomendaba a todos los equipos los mismos. En el caso de los parámetros de equipos, esto se podía ver mediante los ratings, ya que para un mismo equipo, todos los ratings eran sumamente parecidos.

5. EVALUACIÓN Y RESULTADOS

En este capítulo se presentan los resultados obtenidos tras la implementación y entrenamiento de los diversos modelos de filtrado colaborativo y benchmarks descritos en el Capítulo 4. El objetivo es hacer un análisis generar del enfoque, cuantificar el rendimiento predictivo de cada método y analizar las diferencias entre ellos, utilizando el conjunto de test previamente definido y las métricas de evaluación detalladas.

La evaluación se estructura en tres partes. Primero, se exponen los resultados de los modelos individuales junto a los benchmarks introducido por un análisis de hiperparámetros. Luego, se presentan los resultados obtenidos mediante la aplicación de técnicas de blending, donde se combinan las predicciones de los modelos individuales para generar una predicción final potencialmente más precisa. Por último, se presentan recomendaciones reales de algunos modelos.

Es importante recordar que todas las evaluaciones se realizan sobre un conjunto de test estrictamente separado, correspondiente a la última temporada disponible (2025) y filtrado para incluir únicamente jugadores y equipos presentes también en el conjunto de entrenamiento.

5.1. Métricas de evaluación

La evaluación de sistemas de recomendación es multifacética, ya que un "buen"sistema no sólo debe ser preciso en sus predicciones de ratings, sino también útil para el usuario final, por ejemplo, al ordenar correctamente las posibles recomendaciones. En principio no está claro que un modelo con bajo RMSE (métrica propuesta por Netflix para evaluar las predicciones) brinde mejores recomendaciones que uno con un poco más alto. Por ello, se utilizaron diversas métricas para obtener una visión un poco más amplia y comprensiva del rendimiento de los modelos.

- **Root Mean Squared Error (RMSE):** Es una métrica estándar para medir la precisión de las predicciones numéricas. Penaliza más los errores grandes. Se calcula como:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{K}_{\text{test}}|} \sum_{(j,e)\in\mathcal{K}_{\text{test}}} (R_{je} - \hat{R}_{je})^2}$$

  donde $\mathcal{K}_{\text{test}}$ es el conjunto de test de tuplas jugador-equipo o jugador-equipo-temporada y $|\mathcal{K}_{\text{test}}|$ indica su cardinalidad. Fue la métrica oficial del Grand Netflix Prize.

- **Mean Absolute Error (MAE):** Mide el promedio de las diferencias absolutas

entre los ratings reales y los predichos. Es menos sensible a outliers que el RMSE.

$$\text{MAE} = \frac{1}{|\mathcal{K}_{\text{test}}|} \sum_{(j,e)\in\mathcal{K}_{\text{test}}} |R_{je} - \hat{R}_{je}|$$

Se considera adecuada también porque el interés recae en la magnitud del error absoluto.

- **Mean Absolute Percentage Error (MAPE o PMAE):** Siguiendo con la métrica anterior, indica el promedio del error absoluto como un porcentaje del valor real. Un valor más bajo indica un mejor ajuste. Se calcula como:

$$\text{MAPE} = \frac{1}{|\mathcal{K}_{\text{test}}|} \sum_{(j,e)\in\mathcal{K}_{\text{test}}} \left| \frac{R_{je} - \hat{R}_{je}}{R_{je}} \right|$$

($R_{je} \neq 0$ para todo rating).

Para evaluar la capacidad de los modelos de ordenar correctamente las recomendaciones, lo cual es crucial para tareas como generar una lista de los "top-N"jugadores más adecuados para un equipo, se utilizaron coeficientes de correlación de rangos:

- **Correlación de Rango de Spearman (ρ):** Mide la fuerza y dirección de la asociación entre dos variables ordenadas. En este contexto, se aplica a los rangos de los ratings reales ($rg(R_{je})$) y los rangos de los ratings predichos ($rg(\hat{R}_{je})$) para un conjunto de N ítems evaluados para un usuario (o viceversa). Si d_i es la diferencia entre los rangos de cada par de observaciones, la fórmula es:

$$\rho = 1 - \frac{6\sum d_i^2}{N(N^2-1)}$$

donde $d_i = rg(R_{je_i}) - rg(\hat{R}_{je_i})$ para la i-ésima observación. Un valor de ρ cercano a 1 indica una alta concordancia positiva en el ordenamiento (los rankings son similares), un valor cercano a -1 indica una alta concordancia negativa (ordenamientos inversos), y un valor cercano a 0 indica poca o ninguna concordancia.

- **Tau de Kendall (τ):** Es otra medida de correlación de rangos que cuenta el número de pares concordantes y discordantes entre dos rankings. Es robusta y también mide la similitud en el ordenamiento. La fórmula básica para Tau-a (cuando no hay empates en los rangos) es:

$$\tau_a = \frac{N_c - N_d}{\frac{1}{2}N(N-1)}$$

donde N_c es el número de pares concordantes, N_d es el número de pares discordantes, y N es el número total de ítems. En general, un valor de τ cercano a 1 indica una

fuerte concordancia en el ordenamiento, mientras que un valor cercano a -1 indica una fuerte discordancia.

5.2. Benchmarks

Para contextualizar el rendimiento de los modelos desarrollados, se establecieron varios benchmarks, desde los más simples hasta algunos un poco más complejos. La metodología para desarrollarlos fue comenzar desde los más básicos e ir incrementando su complejidad, siempre siguiendo la lógica del problema y las propuestas del caso del Gran Premio de Netflix. Todos los benchmarks fueron entrenados y evaluados sobre los mismos conjuntos de entrenamiento y test que los modelos principales.

1. **Media Global:** Predice el rating medio de todo el conjunto de entrenamiento para cada instancia del conjunto de test. Es el benchmark más básico.

2. **Promedio por Equipo:** Para un par (jugador j, equipo e), se predice el rating promedio histórico del equipo e en el conjunto de entrenamiento.

3. **Promedio por Jugador:** Para un par (jugador j, equipo e), se predice el rating promedio histórico que el jugador j ha obtenido en todos sus equipos en el conjunto de entrenamiento.

4. **Predictores Base (Sección 4.1):** Los modelos de sesgos (estáticos y temporales) descritos anteriormente también sirven como benchmarks importantes, ya que representan el nivel de predicción alcanzable sin considerar interacciones complejas de filtrado colaborativo.

5. **Modelo de Machine Learning Clásico (XGBoost/GBDT con Features Clásicas):** Un modelo XGBoost entrenado únicamente con las "Features Clásicas" con las que contabamos en nuestro dataset (goles, asistencias, etc.) de la Sección 4.5, sin usar las predicciones de los modelos de CF y sin utilizar los ID de jugadores y equipos ya que este modelo no recibe variables categóricas sino numéricas u ordinales. Si bien no es la manera usual que se usan estos modelos en el dominio del fútbol, representa un benchmark sencillo de implementar de un modelo competitivo en muchas áreas que nos permite utilizar las instancias del dataset de manera similar a los métodos de filtrado colaborativo. Utiliza las instancias de entrenamiento sin contar con datos de jugador ni equipo particulares. Aunque es incapaz de generar recomendaciones nuevas, debido a que utiliza las estadísticas del jugador en el equipo para predecir el rating (lo que le permite correr con cierta ventaja sobre los modelos de CF), es decir que carece de la capacidad de predecir el rendimiento de un jugador en un equipo donde no existe un historial previo, una tarea central para la que el filtrado colaborativo está diseñado. Sin embargo, es útil ya que nos permite entrenar

y testear sobre los mismos datos que los demás modelos, obteniendo así las métricas para compararlos.

Para este modelo, se realizó, a través de cross validation, un grid search de los siguientes hiperparámetros: Number of Estimators, Max Depth y Learning Rate, obteniendo como mejor combinación 200, 5 y 0.01 respectivamente. En la tabla de resultados se observan las mejores tres combinaciones.

5.3. Análisis de hiperparámetros y dimensión del embedding

La performance de los modelos de factores latentes, como los basados en factorización matricial, es sensible a la elección de varios hiperparámetros clave. Entre ellos, la dimensión del embedding (f), la tasa de aprendizaje (γ) para el descenso de gradiente estocástico, y los factores de regularización (λ) son cruciales para lograr un buen equilibrio entre la capacidad del modelo para ajustarse a los datos de entrenamiento y su habilidad para generalizar a datos no vistos.

Una dimensión de embedding demasiado baja podría impedir que el modelo capture la complejidad de las interacciones jugador-equipo, resultando en subajuste (*underfitting*). Por el contrario, una dimensión excesivamente alta, junto con una regularización inadecuada, podría llevar al sobreajuste (*overfitting*), donde el modelo memoriza el ruido de los datos de entrenamiento en lugar de aprender patrones generalizables. De manera similar, la tasa de aprendizaje y el factor de regularización deben ser cuidadosamente calibrados para asegurar una convergencia estable y evitar el sobreajuste, respectivamente.

En esta subsección, se presenta un análisis exploratorio del impacto de estos hiperparámetros en el rendimiento del modelo SVD básico (sin sesgos, solo producto interno $Q_e^T P_j$). El objetivo fue obtener una comprensión inicial de la sensibilidad del modelo y guiar la elección de valores razonables para los modelos más complejos, considerando las limitaciones computacionales y temporales inherentes a una búsqueda exhaustiva para cada variante de modelo.

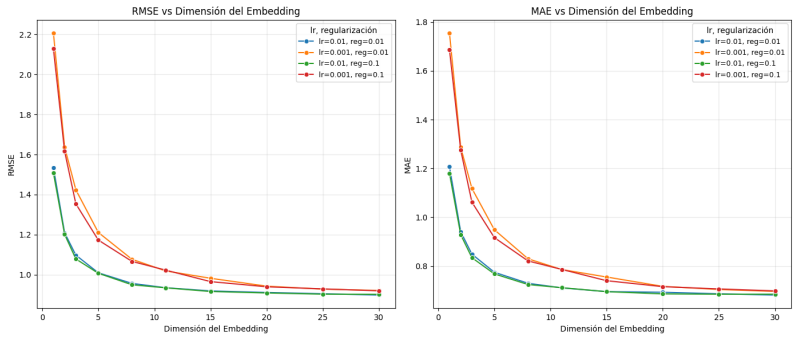

Fig. 5.1: Impacto de la Dimensión del Embedding, Tasa de Aprendizaje y Regularización en el RMSE y MAE del modelo SVD básico.

Manteniendo fijos la tasa de aprendizaje y el factor de regularización en valores razonables (e.g., $\gamma = 0{,}01, \lambda = 0{,}01$, valores que se utilizaron por ser similares a los del caso de Netflix), las curvas del RMSE y MAE en función de la dimensión del embedding exhiben una forma similar a un "codo"(ver Figura 5.1). Inicialmente, incrementar la dimensión de embedding produce mejoras sustanciales, tanto en el MAE como en el RMSE, ya que el modelo gana capacidad expresiva. Sin embargo, a partir de cierto punto, las ganancias se vuelven marginales.

Se observa que esta curva comienza a aplanarse alrededor de una dimensión de embedding de $f = 11$, y las mejoras adicionales son mínimas más allá de $f = 20$. Basándonos en esta observación, y buscando un equilibrio entre rendimiento y complejidad del modelo (un mayor número de dimensiones implica más parámetros a estimar y mayor riesgo de sobreajuste si no se cuenta con datos suficientes), se tomó la decisión de utilizar una dimensión de embedding de 20 para el modelo SVD básico que solo incluía los factores latentes.

Para los modelos de factorización matricial que también incorporan predictores base (sesgos), como el `pred_basicSVD_2`, se optó por una dimensión de embedding ligeramente menor, de 10. La justificación radica en que los sesgos ya capturan una porción significativa de la varianza en los ratings, permitiendo que los factores latentes se enfoquen en capturar las interacciones con una dimensionalidad más reducida, sin una pérdida considerable de precisión.

Finalmente, para los modelos más complejos y con un mayor número de parámetros (como aquellos que incorporan dinámicas temporales tanto en sesgos como en factores latentes, e.g., `pred_timeSVDpp`), se observó empíricamente que reducir aún más la dimensión de embedding, por ejemplo a 3, permitía una disminución considerable del costo computacional (y por lo tanto del tiempo de entrenamiento), sin incurrir en una penalización drástica en términos de error predictivo. Adicionalmente, trabajar con un espacio latente de baja dimensionalidad (como 3 dimensiones) ofrece la ventaja de facilitar un futuro análisis cualitativo e interpretación del significado semántico que el modelo podría estar asignando a cada uno de estos factores latentes, una tarea que se vuelve más compleja a medida que la dimensionalidad aumenta.

Se buscó utilizar estos valores como guía inicial para los modelos, aunque en algunos casos (destacados en el Capítulo 4) hubo que realizar pequeñas modificaciones para evitar sobreajuste. Es importante destacar que, si bien este análisis proporcionó una guía útil, la selección óptima de hiperparámetros puede variar entre diferentes datasets y arquitecturas de modelo específicas. Una búsqueda más exhaustiva, aunque algo costosa, podría potencialmente refinar más estos valores y mejorar los resultados.

Se estima que la curva observada en los gráficos está totalmente relacionada con la cantidad de datos. Esto es debido a que en el caso de los ganadores del Gran Premio de Netflix, donde cuentan con cerca de 1 millón de registros, prueban modelos con hasta 200

dimensiones que disminuyen el error (aunque levemente) con respecto a modelos de menos dimensiones.

5.4. Resultados de Modelos y Benchmarks

El objetivo es cuantificar el rendimiento predictivo de cada enfoque y analizar las diferencias entre ellos, utilizando el conjunto de test previamente definido y las métricas de evaluación detalladas en la sección 5.1.

A continuación, se muestran los resultados de los diferentes benchmarks y modelos de filtrado colaborativo implementados. Para cada modelo, se reportan las cinco métricas de evaluación: Error Cuadrático Medio (RMSE), Error Absoluto Medio (MAE), Error Absoluto Medio en Porcentaje (PMAE), Correlación de Rango de Spearman (ρ) y Tau de Kendall (τ). La media como predictor no posee valores en las métricas de ordenamiento dado que predice siempre el mismo valor.

Tab. 5.1: Resultados de Benchmarks y Modelos Individuales (Ordenados por RMSE ascendente)

ID	Modelo	RMSE	MAE	Spearman	Kendall	PMAE
1	XGBoost (Config. 3)	0.8130	0.6040	0.3095	0.2155	0.1212
2	XGBoost (Config. 2)	0.8165	0.6099	0.3089	0.2148	0.1212
3	Item_Vector_Model	0.8196	0.6238	0.2117	0.1440	0.1151
4	XGBoost (Config. 1)	0.8234	0.6179	0.3041	0.2116	0.1212
5	basicSVD_v1	0.8262	0.6203	0.2110	0.1440	0.1154
6	Baseline_con_temp	0.8263	0.6225	0.2050	0.1401	0.1159
7	Baseline_sin_temp	0.8292	0.6245	0.2067	0.1410	0.1160
8	Basic_SVD_2 (SVD con Biases Temporales)	0.8356	0.6368	0.1369	0.0927	0.1181
9	TimeSVD++	0.8361	0.6320	0.2045	0.1394	0.1179
10	SVD++ simplified (2)	0.8363	0.6373	0.1186	0.0798	0.1179
11	Item_Vector_Model_simplified	0.8365	0.6332	0.0892	0.0599	0.1180
12	TimeSVD++ simplified	0.8375	0.6332	0.1891	0.1289	0.1182
13	baseline_timeSVD++	0.8392	0.6346	0.0542	0.0366	0.1184
14	media	0.8396	0.6350	NaN	NaN	0.1212
15	Media Equipo	0.8451	0.6437	0.1234	0.0833	0.1192
16	basicSVD_v2	0.8659	0.6536	0.1199	0.0814	0.1212
17	Baseline_con_temp_bins	0.8667	0.6577	0.1565	0.1060	0.1215
18	Media Jugador	0.9248	0.6864	0.1694	0.1157	0.1265
19	SVD++	0.9768	0.7501	0.0458	0.0309	0.1365
20	similaridad	1.0318	0.7723	0.1806	0.1227	0.1406
21	basicSVD_0 (SVD Simple sin biases)	1.6872	1.4931	0.1565	0.1060	0.2440

Además, para complementar el análisis y ofrecer una representación más visual del rendimiento comparativo de los modelos individuales y benchmarks, se presentan a continuación gráficos de barras que resumen algunas de las métricas clave discutidas anteriormente.

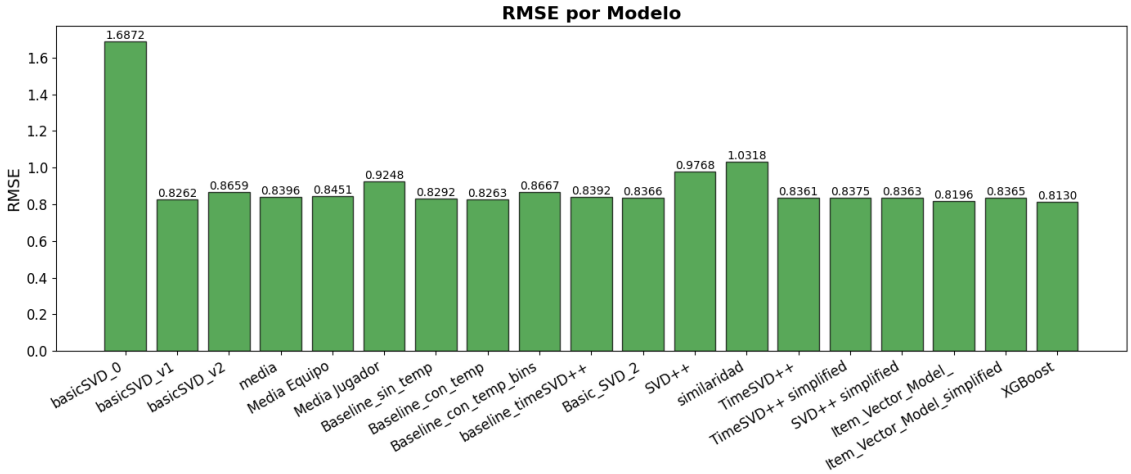

Fig. 5.2: Comparación del RMSE por Modelo para Benchmarks y Modelos Individuales.

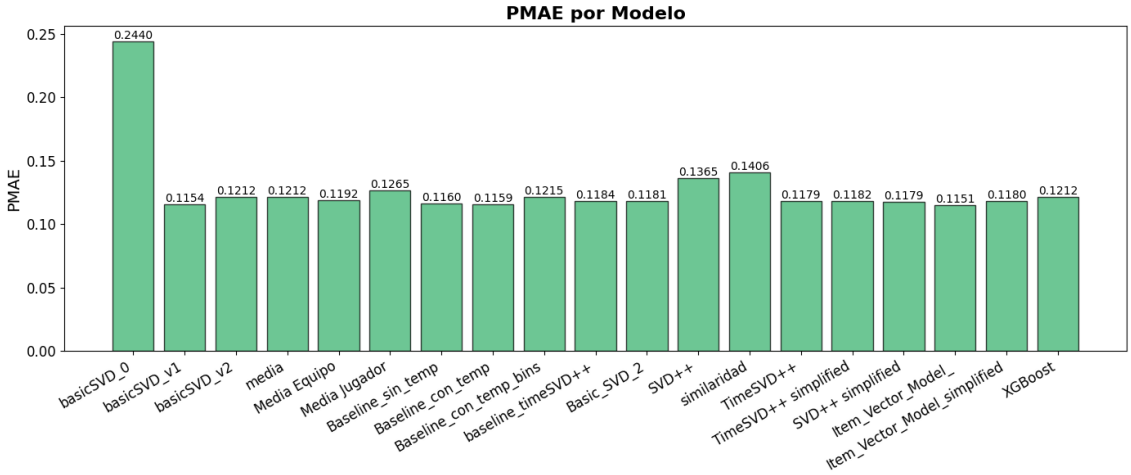

Fig. 5.3: Comparación del PMAE por Modelo para Benchmarks y Modelos Individuales.

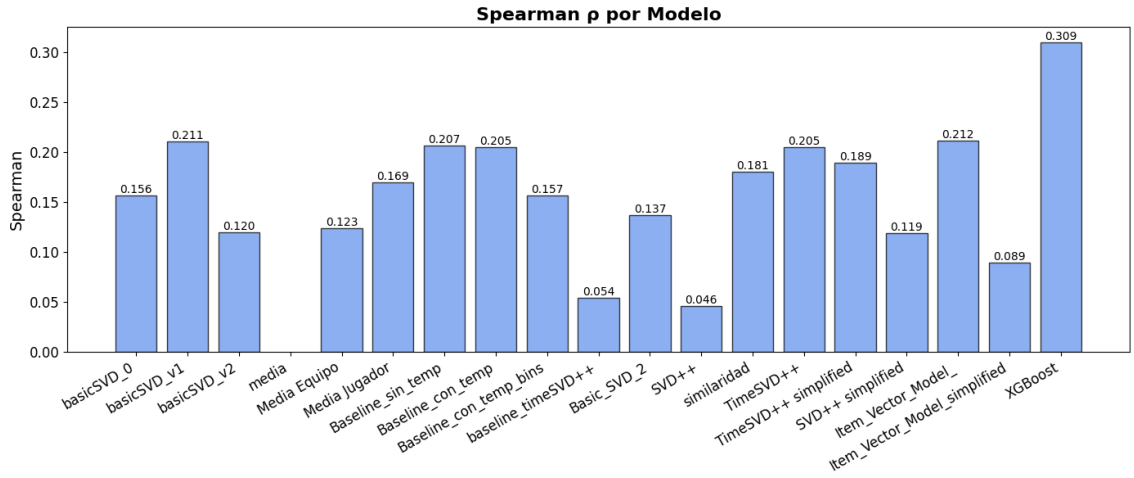

Fig. 5.4: Comparación de la Correlación de Rango de Spearman (ρ) por Modelo para Benchmarks y Modelos Individuales.

5.5. Resultados de Blending

A continuación, se presentan los resultados obtenidos al aplicar los tres algoritmos de blending (Regresión Lineal con regularización Ridge, LightGBM como representante de GBDT, y StackingRegressor) sobre los tres diferentes conjuntos de características: estadísticas clásicas del fútbol (estadísticas), las predicciones de los modelos de filtrado colaborativo (predictores CF), y la combinación de ambas (combinado). Es importante destacar que los modelos que utilizan estadísticas clásicas son incapaces de producir predicciones nuevas ya que necesitan las mismas para generar las predicciones. Por ello, son incapaces de generar recomendaciones nuevass. Esta es una clara ventaja de los métodos de filtrado colaborativo y de las técnicas de blending que utilizan solamente esos métodos. Sin embargo, es interesante estudiar cuánta información extra aportan.

Al igual que en la sección anterior, se incluyen tres gráficos de barras para complementar el análisis y ofrecer una representación más visual del rendimiento

Tab. 5.2: Resultados de Modelos de Blending (Ordenados por RMSE ascendente)

ID	Modelo (Conjunto de Features)	RMSE	MAE	Spearman	Kendall	PMAE
1	Regresión combinado	0.7685	0.5682	0.3401	0.2355	0.1043
2	Stacking combinado	0.7691	0.5694	0.3327	0.2299	0.1044
3	LightGBM combinado	0.7790	0.5762	0.3271	0.2257	0.1057
4	Regresión estadísticas	0.7800	0.5764	0.3168	0.2224	0.1062
5	Stacking estadísticas	0.7803	0.5764	0.3186	0.2225	0.1063
6	LightGBM estadísticas	0.7864	0.5807	0.3088	0.2141	0.1071
7	LightGBM predictores (CF)	0.7877	0.5970	0.2250	0.1532	0.1094
8	Regresión predictores (CF)	0.7913	0.5990	0.1962	0.1339	0.1098
9	Stacking predictores (CF)	0.7922	0.6007	0.1947	0.1321	0.1100

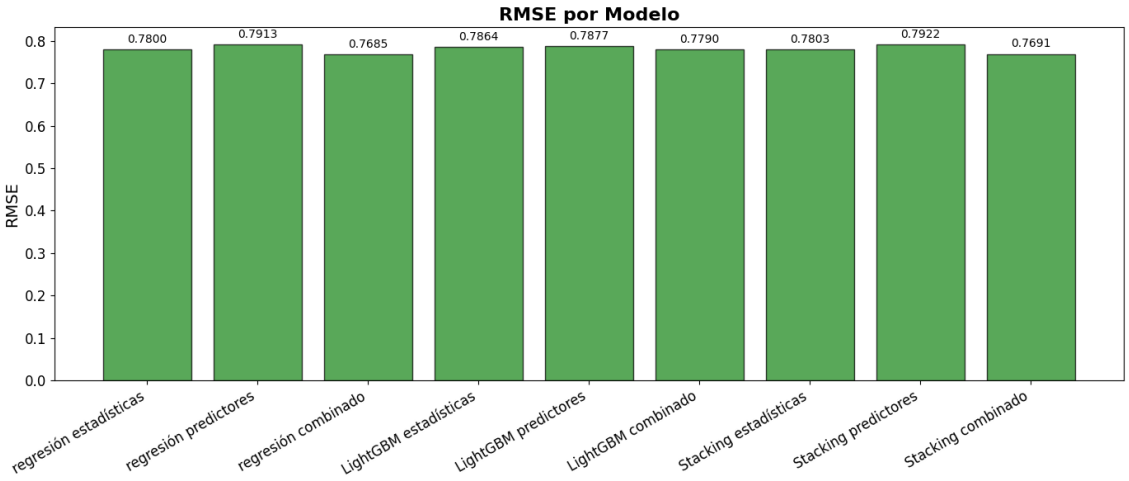

Fig. 5.5: Comparación del RMSE por Modelo para Estrategias de Blending.

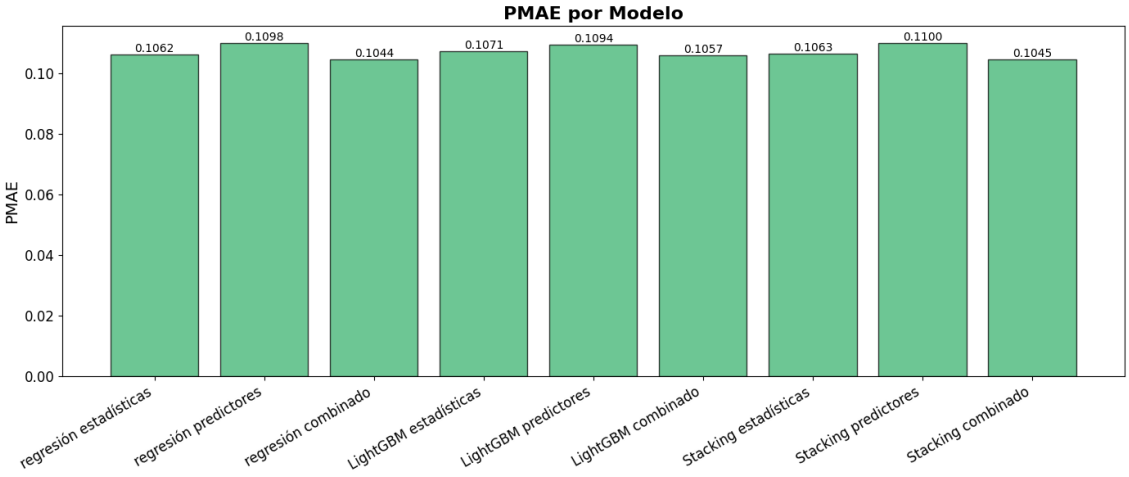

Fig. 5.6: Comparación del PMAE por Modelo para Estrategias de Blending.

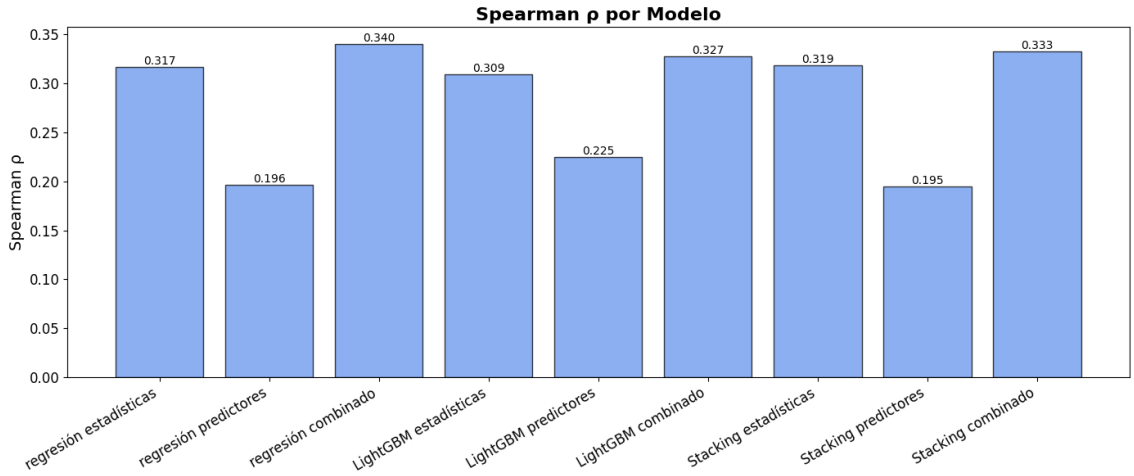

Fig. 5.7: Comparación de la Correlación de Rango de Spearman (ρ) por Modelo para Estrategias de Blending.

5.6. Análisis de Recomendaciones Top-N

Más allá de las métricas de error y ranking globales, resulta ilustrativo investigar las recomendaciones específicas que generan los modelos. Este análisis cualitativo nos permite obtener una intuición sobre cómo los diferentes enfoques interpretan los datos y qué tipo de perfiles de jugadores tienden a sugerir para determinados equipos. Se seleccionaron modelos representativos, ya sea por su bajo error o por tener un enfoque distinto que se consideró interesante para analizar. De este modo, se entrenaron los modelos seleccionados utilizando la totalidad de los datos disponibles (conjunto de entrenamiento más el conjunto de test). Posteriormente, para un conjunto de equipos de interés y para el año actual (considerado como 2025), se guardaron los 5 jugadores con el mayor rating predicho por cada modelo para cada equipo.

La selección de equipos para este análisis se basó en los siguientes criterios:

- **Racing de Santander:** Como caso de estudio central de la tesis y por la disponibilidad de un número considerable de sus jugadores (65) en el dataset BeSoccer.
- **Chelsea y Liverpool:** Representantes de la Premier League inglesa, una de las ligas más competitivas del mundo. Se eligieron por tener una alta representación de jugadores en el dataset (67 y 51, respectivamente), lo que permite generar recomendaciones más robustas.
- **Milan:** Como representante destacado de la Serie A italiana, siendo el equipo grande de esta liga con más jugadores (59) en el dataset.
- **Sevilla:** Como otro representante de La Liga española (al igual que Racing), con una presencia significativa de jugadores (71) en el dataset.

A continuación, se presentan las recomendaciones Top-5 generadas por modelos representativos para los equipos seleccionados, correspondientes al año 2025.

5.6.1. Modelo simple SVD

Este modelo incorpora únicamente factores latentes.

- **Recomendaciones para Milan (2025):**
  1. E. DŽEKO, Rating Estimado: 5.61
  2. S. SIRIGU, Rating Estimado: 5.48
  3. R. FALCAO, Rating Estimado: 5.45
  4. MILOŠ KERKEZ, Rating Estimado: 5.44
  5. C. LENGLET, Rating Estimado: 5.43
- **Recomendaciones para Racing (2025):**
  1. SERGIO GONZÁLEZ, Rating Estimado: 5.95
  2. PABLO VÁZQUEZ, Rating Estimado: 5.78
  3. VÍCTOR RUIZ, Rating Estimado: 5.69
  4. DIEGO LLORENTE, Rating Estimado: 5.36
  5. Á. FERLLO, Rating Estimado: 5.34
- **Recomendaciones para Chelsea (2025):**
  1. RICHARLISON, Rating Estimado: 5.36
  2. JUAN CASTILLO, Rating Estimado: 5.28
  3. T. DIDILLON, Rating Estimado: 5.26
  4. SEOK-HO HWANG, Rating Estimado: 5.25
  5. RĂZVAN MARIN, Rating Estimado: 5.17
- **Recomendaciones para Liverpool (2025):**
  1. HARRY KANE, Rating Estimado: 5.35
  2. MARK FLEKKEN, Rating Estimado: 5.18
  3. ZANKA, Rating Estimado: 5.17
  4. P. HØJBJERG, Rating Estimado: 5.09
  5. YVON MVOGO, Rating Estimado: 5.02
- **Recomendaciones para Sevilla (2025):**

1. M. RYAN, Rating Estimado: 5.68
2. M. SALAH, Rating Estimado: 5.53
3. H. MORENO, Rating Estimado: 5.45
4. NICOLAS TAGLIAFICO, Rating Estimado: 5.40
5. C. ZAPATA, Rating Estimado: 5.23

5.6.2. Modelo SVD Básico versión 2

Este modelo incorpora sesgos estáticos a los factores latentes. Analizaremos únicamente la versión que tiene un parámetro más bajo de regularización para los factores latentes, ya que genera recomendaciones diferentes para cada equipo a pesar de tener un error mayor.

- **Recomendaciones para Racing (2025):**
  1. C. CORVALÁN, Rating Estimado: 7.06
  2. LUUK DE JONG, Rating Estimado: 7.01
  3. A. BASTONI, Rating Estimado: 7.00
  4. M. CAPASSO, Rating Estimado: 7.00
  5. P. AGUILAR, Rating Estimado: 7.00
- **Recomendaciones para Milan (2025):**
  1. RUBÉN LÓPEZ, Rating Estimado: 7.46
  2. GREGOR KOBEL, Rating Estimado: 7.45
  3. CALLUM DOYLE, Rating Estimado: 7.39
  4. M. GREENWOOD, Rating Estimado: 7.29
  5. K. SCHMEICHEL, Rating Estimado: 7.29
- **Recomendaciones para Chelsea (2025):**
  1. LUCA MARINAUCCI, Rating Estimado: 7.77
  2. FRAN GONZÁLEZ, Rating Estimado: 7.56
  3. HANNES WOLF, Rating Estimado: 7.46
  4. R. PICCOLI, Rating Estimado: 7.36
  5. EREN DINKÇI, Rating Estimado: 7.33
- **Recomendaciones para Liverpool (2025):**
  1. R. RAMOS, Rating Estimado: 8.24
  2. ANTÓN ESCOBAR, Rating Estimado: 7.66

3. MARIO CLIMENT, Rating Estimado: 7.60
4. R. ECHEVERRÍA, Rating Estimado: 7.44
5. T. SCHÜTZENAUER, Rating Estimado: 7.43

- **Recomendaciones para Sevilla (2025):**

1. LUCAS PAES, Rating Estimado: 7.52
2. S. MKRTCHYAN, Rating Estimado: 7.38
3. M. SCHULZ, Rating Estimado: 7.33
4. LÉO JARDIM, Rating Estimado: 7.32
5. CONOR HAZARD, Rating Estimado: 7.24

5.6.3. Modelo de Vecindario Basado en Similaridad

(`pred_similaridad`) Este modelo se basa en la correlación de Pearson como medida de similaridad entre los perfiles de rating de los equipos. Es interesante notar que los ratings predichos por este modelo pueden exceder la escala original de los datos (1-10), lo cual es una característica de algunos modelos de vecindario puros si no se aplica una normalización posterior a la predicción.

- **Recomendaciones para Chelsea (2025):**

1. XAVI QUINTILLÀ (ID 141829), Rating Estimado: 10.18
2. E. TAŞKIRAN (ID 74417), Rating Estimado: 9.27
3. M. DORIA (ID 237319), Rating Estimado: 8.65
4. TOM HEATON (ID 56714), Rating Estimado: 8.64
5. NICK POPE (ID 206005), Rating Estimado: 8.41

- **Recomendaciones para Milan (2025):**

1. C. CARTER-VICKERS (ID 247458), Rating Estimado: 9.77
2. K. SCHMEICHEL (ID 50), Rating Estimado: 9.43
3. E. TAŞKIRAN (ID 74417), Rating Estimado: 9.36
4. D. LUCKASSEN (ID 232111), Rating Estimado: 9.25
5. CALEB OKOLI (ID 693104), Rating Estimado: 9.20

- **Recomendaciones para Liverpool (2025):**

1. JOSHUA ZIRKZEE (ID 459633), Rating Estimado: 8.49
2. D. FARAONI (ID 249679), Rating Estimado: 8.32

3. REMO FREULER (ID 191669), Rating Estimado: 8.23
4. C. O'HARE (ID 353087), Rating Estimado: 8.18
5. A. PLÉA (ID 138887), Rating Estimado: 8.17

- **Recomendaciones para Racing (2025):**

1. ÉDGAR BADÍA (ID 110659), Rating Estimado: 8.23
2. R. LEWANDOWSKI (ID 53536), Rating Estimado: 7.94
3. BORJA IGLESIAS (ID 162561), Rating Estimado: 7.80
4. BARANDALLA (ID 1024465), Rating Estimado: 7.79
5. UMAR SADIQ (ID 239961), Rating Estimado: 7.78

- **Recomendaciones para Sevilla (2025):**

1. L. HENDERSON (ID henderson), Rating Estimado: 8.98
2. TOM HEATON (ID 56714), Rating Estimado: 8.52
3. W. FALCONE (ID 162352), Rating Estimado: 8.44
4. S. MOORE (ID 247622), Rating Estimado: 8.43
5. JAIR (ID 306735), Rating Estimado: 8.31

5.6.4. Modelo de Vecindario con Pesos Globales(`pred_item_vector_model`)

Este modelo aprende pesos de interacción global entre equipos (w_{ef} y c_{ef}). Este es el modelo con más bajo error dentro de los estudiados bajo el enfoque de filtrado colaborativo (ver tabla 5.2). Sin embargo, se podría decir que es demasiado "seguro". Esto es debido a que tiende a recomendar jugadores que ya habían tenido un paso exitoso por el club al que se recomienda. Por este motivo, se decidió filtrar las recomendaciones a jugadores que no han jugado en ese club.

- **Recomendaciones para Racing (2025):**

1. TOMEU NADAL (ID 27665), Rating Estimado: 7.67
2. ROB NIZET (ID 813042), Rating Estimado: 6.90
3. BONO (ID 139107), Rating Estimado: 6.85
4. F. BOHNERT (ID bohnert), Rating Estimado: 6.85
5. N. SCHLOTTERBECK (ID 337248), Rating Estimado: 6.83

- **Recomendaciones para Milan (2025):**

1. L. OLIVAS (ID 448599), Rating Estimado: 6.84

2. JOÃO CARVALHO (ID 247067), Rating Estimado: 6.78
3. A. CEITIL (ID 301833), Rating Estimado: 6.76
4. RIKI MANGANA (ID 416210), Rating Estimado: 6.76
5. S. ICHAZO (ID 130450), Rating Estimado: 6.74

- **Recomendaciones para Chelsea (2025):**

1. JORDAN WILLIAMS (ID 335424), Rating Estimado: 6.88
2. T. VAN DEN BELT (ID 712028), Rating Estimado: 6.77
3. M. CAILLARD (ID 139669), Rating Estimado: 6.70
4. TRISTHAN JAIMES (ID 829675), Rating Estimado: 6.68
5. TOMA BAŠIĆ (ID basic), Rating Estimado: 6.68

- **Recomendaciones para Liverpool (2025):**

1. BRAHIM DÍAZ (ID 297592), Rating Estimado: 7.46
2. H. MKHITARYAN (ID 139087), Rating Estimado: 7.15
3. M. TEPE (ID 459712), Rating Estimado: 6.96
4. COHEN BRAMALL (ID 357091), Rating Estimado: 6.81
5. J. BONHAM (ID 190825), Rating Estimado: 6.76

- **Recomendaciones para Sevilla (2025):**

1. SERGIO MOLINA (ID 136596), Rating Estimado: 6.73
2. JOEL LÓPEZ (ID 775948), Rating Estimado: 6.62
3. D. PETKOVIC (ID 123852), Rating Estimado: 6.56
4. DAVID ALABA (ID 58931), Rating Estimado: 6.55
5. D. GONZÁLEZ (ID 72753), Rating Estimado: 6.55

5.6.5. Modelo TimeSVD++ Simplificado(`pred_timeSVDpp_simplified`)

Este modelo incorpora factores latentes de jugador que evolucionan con el tiempo, junto con sesgos estáticos. Se presenta sólo el caso de Racing debido a que el modelo presenta el mismo top 5 para todos los equipos analizados, con una leve diferencia de rating.

- **Recomendaciones para Racing (2025):**

1. R. LEWANDOWSKI (ID 53536), Rating Estimado: 6.08
2. C. RONALDO (ID 28185), Rating Estimado: 6.07

3. HULK (ID 22295), Rating Estimado: 6.07
4. H. JELASSI (ID 324852), Rating Estimado: 6.07
5. S. NANASI (ID 432796), Rating Estimado: 6.07

6. ANÁLISIS DE RESULTADOS Y DISCUSIÓN

En este capítulo, se analizarán los resultados presentados anteriormente, con el objetivo de interpretar su significado, discutir las implicaciones tanto técnicas como futbolísticas, y reflexionar sobre las limitaciones inherentes al estudio. Se busca conectar los hallazgos numéricos con el contexto del fútbol profesional, evaluando no solo la precisión predictiva sino también la utilidad de las recomendaciones reales generadas.

6.1. Análisis de resultados con métricas clásicas (RMSE, MAE, PMAE)

La evaluación inicial del rendimiento de los modelos se basa en métricas clásicas de error como el RMSE, MAE y PMAE. La Tabla 5.1 sirve como referencia principal para esta discusión.

6.1.1. Análisis general

La primera consideración a tener en cuenta a la hora de analizar los resultados de la Tabla 5.1 es el bajo error que tiene la media como predictor. Esto indica que estamos ante un problema difícil inherente a los datos, ya que hay poca variabilidad por explicar a través de los modelos. Observando el PMAE, notamos que el Item Vector Model, el mejor modelo (que supera incluso a los XGBoost), explica apenas un 5 % de esa variabilidad por explicar. Esto no indica que los modelos no sean una herramienta útil, ni que los métodos de filtrado colaborativo no sean aplicables al dominio del fútbol, pero sí que utilizar sus recomendaciones sin un análisis previo del funcionamiento del modelo, sus errores y sus fortalezas implicaría cometer un error.

Además, al observar otras métricas como RMSE o MAE, si bien es un poco mayor la disminución del error por parte de algunos modelos con respecto a la media, la diferencia es muy pequeña ante un predictor tan básico como este.

Por este motivo, se plantean algunas explicaciones posibles ante este fenómeno de poca explicación de la variabilidad por parte de los modelos: En primer lugar, tendría sentido plantearse si el rating es una variable viable para predecir, teniendo en cuenta su poca variabilidad y que estamos calculando, no el desempeño de la temporada, sino el promedio de los ratings de cada partido jugado en la temporada. Ante esta situación, se podrían buscar variables alternativas relacionadas al desempeño, como la cantidad de partidos por encima de los 7 puntos. Otra propuesta interesante sería utilizar estos métodos para predecir algún aspecto particular del jugador que resulte interesante para el equipo. En este caso se podría filtrar el target por posición, buscando predecir la cantidad de amagues y goles, si es delantero, de pases al último tercio si es mediocampista, de quites si es defensor,

y de atajadas, en caso de los arqueros. También tiene sentido preguntarse si la cantidad de datos es suficiente. Un dataset más completo y con datos de jugadores antiguos podría enriquecer el estudio, permitiendo también un filtrado menor de jugadores y con ello un análisis más completo del caso. Por último sí queda preguntarse si es plausible capturar la variabilidad del dominio con modelos y técnicas de filtrado colaborativo. Si bien no es algo que se pueda responder por medio de esta investigación, al comparar con un modelo de machine learning potente y muy utilizado como es el XGBoost, con datos del jugador ya habiendo pasado por el equipo, se observa que tampoco puede explicar mucho más que la media, incluso mostrando una performance comparable a la de algunos modelos de CF. Por este motivo, se estima que no es la causa principal de la poca reducción del error con respecto a la media.

A la hora de comparar los modelos entre sí, Un aspecto que llama la atención es el rendimiento de los modelos más simples, incluyendo los propios predictores base (sesgos). Modelos como los `basicSVD` (factorización matricial con sesgos) y el `Item_Vector_Model` (vecindario con pesos globales aprendidos) se sitúan competitivamente, obteniendo un RMSE y MAE considerablemente bajos y superando a variantes más complejas que incorporan dinámicas temporales. Este fenómeno podría atribuirse a varios factores: Primero, dada la cantidad de datos disponibles, es posible que los modelos con más parámetros, especialmente aquellos con componentes temporales, tiendan a sobreajustarse a las particularidades del conjunto de entrenamiento. Esto podría llevar a una menor capacidad de generalización y, por ende, a un rendimiento inferior en el conjunto de test. La dinámica temporal podría requerir un volumen de datos aún mayor para ser estimada de manera robusta. Además, si bien se procuró adaptar los modelos inspirados en el trabajo de los ganadores del Gran Premio de Netflix al contexto del fútbol, es posible que ciertas asunciones o parametrizaciones óptimas para la recomendación de películas no se traduzcan directamente al rendimiento futbolístico, pudiendo resultar en arquitecturas subóptimas para este dominio específico. Un caso de ejemplo puede ser la función utilizada para modelar las dinámicas temporales en el predictor base del TimeSVD++.

6.1.2. Impacto de agregar sesgos temporales

Al comparar modelos con y sin componentes temporales explícitos en los sesgos, los resultados sugieren una imagen compleja. Por ejemplo, el `Baseline_con_temp` (sesgos de jugador y equipo dependientes de la temporada) muestra un RMSE ligeramente más bajo que el `Baseline_sin_temp`. Sin embargo, el `Basic_SVD_2` (SVD con sesgos temporales) presenta un RMSE algo mayor al `basicSVD` (SVD con sesgos estáticos).

Esto podría indicar que, si bien la temporalidad es un factor intrínsecamente relevante en el rendimiento futbolístico, su modelado incorrecto o con parámetros insuficientemente

regularizados puede no traducirse directamente en una mejora del error predictivo general, o incluso podría inducir sobreajuste.

Estos resultados no descartan el valor de incorporar dinámicas temporales. Modificaciones en la parametrización (e.g., funciones más suaves, regularización más fuerte para los componentes temporales), en el modelado o una exploración más exhaustiva de hiperparámetros podrían conducir a resultados diferentes y potencialmente más beneficiosos.

6.1.3. Análisis de modelos de vecindario vs modelos de factores latentes

Al observar la Tabla 5.1, no se observa una superioridad categórica de una familia de técnicas sobre la otra en términos de RMSE y MAE. Si bien el `Item_Vector_Model` (vecindario) se ubica como el mejor modelo en cuanto a error, modelos que únicamente incorporan factores latentes (como el `Basic_SVD_2`) se ubican por encima de otros que incorporan dinámicas provenientes del enfoque de vecindario (como el `SVD++` o el `TimeSVD++`).

Además, observamos que modelos más complejos que intentan combinar explícitamente ideas de ambos mundos no superan consistentemente a sus contrapartes más simples en estas métricas. Un ejemplo de esto sería que el `Basic_SVD_2` se encuentra por sobre el `SVD++`. Esto podría reforzar la hipótesis del sobreajuste en presencia de pocos datos para estimar el gran número de parámetros que poseen estos modelos.

La intuición derivada de la literatura [13] es que los modelos de factores latentes son buenos capturando señales globales y tendencias generales, mientras que los modelos de vecindario pueden ser mejores identificando relaciones más locales y específicas. En un escenario con más datos, la combinación de ambas filosofías podría ser la estrategia más robusta.

6.1.4. Comparación de modelos de blending

Al analizar los resultados de la Tabla 5.2 la primera observación clave es que, para los tres algoritmos de blending probados (Regresión Lineal, LightGBM, Stacking), la combinación de "Todas las Features"(estadísticas clásicas + predicciones de modelos CF) supera a los conjuntos de features por separado en los tres algoritmos estudiados. Esto indica que tanto las estadísticas descriptivas tradicionales del fútbol como las predicciones generadas por los modelos de filtrado colaborativo aportan algo de información valiosa y complementaria, es decir que ninguno de los dos tipos de información es completamente redundante respecto al otro.

El modelo de `Regresión` utilizando todas las features emerge como el de mejor rendimiento en términos de RMSE, MAE y PMAE, aunque con resultados similares a los modelos de GBDT y Stacking.

Es fundamental recordar que estos resultados de blending se obtuvieron sobre una partición del conjunto de test original, por lo que no son directamente comparables con

los RMSE de los modelos individuales de la Tabla 5.1, que se evaluaron sobre un conjunto de test diferente (o una porción mayor del mismo).

6.1.5. Interpretación de resultados en el contexto del fútbol

Interpretar estas métricas de error puramente numéricas en el amplio contexto del fútbol es una tarea compleja. Un RMSE de 0.82 puede ser bueno o malo dependiendo de la escala y la varianza de los ratings originales, y de la dificultad inherente del campo de estudio. Como se mencionó anteriormente, es un desafío comprender qué y cuánto de la realidad del fútbol puede ser capturada por estos modelos.

Por estos motivos, y para trascender el análisis puramente numérico, se analizan las recomendaciones generadas por algunos de estos modelos en escenarios con jugadores y equipos reales. Este análisis cualitativo y tal vez algo abstracto para quienes no están adentrados en este deporte, busca evaluar y estudiar qué aspectos futbolísticamente relevantes pueden estar capturando los modelos.

6.2. Comparación y análisis de recomendación de jugadores

Para evaluar la utilidad práctica y la interpretabilidad futbolística de los modelos, se generaron recomendaciones "top-5"para varios equipos conocidos (Racing, Milan, Chelsea, Liverpool, Sevilla). Las recomendaciones se hicieron para la temporada 2025, utilizando diferentes modelos. Podemos imaginar el caso en que la recomendación se está dando a mitad de temporada, ya que usa datos hasta esta temporada (que a la hora de tomar los datos no está terminada) y predice para esa misma última temporada.

Este análisis cualitativo, aunque más subjetivo, es importante. Muestra que diferentes modelos, incluso con RMSE similares, pueden generar recomendaciones muy distintas, capturando aspectos muy diferentes de las interacciones entre jugadores y equipos.

6.2.1. Modelo simple SVD o BasicSVD 0

Este modelo genera ratings muy bajos. Sin embargo, algunas de las recomendaciones parecen tener sentido. Al hacer un análisis futbolístico, podemos ver que a todos los clubes grandes les recomienda como primera recomendacion, delanteros clásicos y constantes que han tenido éxito en su liga (Dzeko al Milan, con éxito en el Inter, Richarlison al Chelsea, éxito en el Tottenham, Kane al Liverpool, éxito en el Tottenham) Cómo otra similitud a destacar, a todos los equipos les recomienda un arquero que haya pasado por otro equipo de su liga. Al no adquirir dinámicas temporales, parece que por lo general recomienda jugadores constantes en su desempeño a lo largo del tiempo y que hayan jugado en la misma liga que el equipo. Esto último no es algo que el modelo haya entrenado particularmente, pero seguramente el hecho de que los equipos tengan más jugadores en común con esos

clubes influyó en la modificación de factores latentes logrando finalmente que sean elegidos por sobre otros.

6.2.2. Modelo basicSVD versión 2

Algo a destacar de este modelo es la recomendación de jugadores atípicos. La mayoría de los futbolistas recomendados no están dentro de los jugadores más reconocidos a nivel mundial, lo que muestra que los factores latentes seguramente expliquen una buena parte de la recomendación, quitándole peso a los sesgos. Al analizar las recomendaciones de este modelo para el Racing de Santander desde un punto de vista futbolístico, llama la atención la presencia de tres jugadores con pasado en el fútbol argentino que no jugaron en la liga española, junto a otros dos jugadores de nivel internacional como son Bastoni y De Jong. Esto se pudo deber a un nivel alto de estos jugadores en su liga, sumado a algun factor latente compartido por Racing de Santander y algunos equipos de la Liga Argentina que el modelo pudo haber encontrado.

6.2.3. Modelo TimeSVD++_simplified

Este modelo es un claro ejemplo de un modelo inútil. Como se menciona en la sección 5.6.5, únicamente se muestran las recomendaciones para Racing de Santander debido a que para los demás son las mismas. Recomienda únicamente un conjunto similar o igual de jugadores de "alta calidad general percibida"(Lewandowski, Ronaldo, Hulk). Esto podría indicar que algún parámetro está dominando el valor predicho. Una posibilidad es que los factores latentes temporales estén capturando una noción de "calidad general en forma actual"que tiende a dominar sobre las interacciones específicas equipo-jugador. Otra opción es que los sesgos temporales están dominando la recomendación o que la dimensionalidad del embedding (f=3) es demasiado baja para capturar interacciones más finas y específicas del contexto del equipo. también podría ocurrir que la regularización de los sesgos base de jugador y equipo no sea lo suficientemente fuerte, permitiendo que estos sesgos dominen la predicción por sobre la interacción de factores.

6.2.4. Modelo pred_similaridad

Los ratings predichos por este modelo son notablemente más altos y con mayor varianza que los de los modelos de factorización (esto explica mucho del error del modelo). Esto es característico de los modelos de vecindario que no están regularizados como los de factorización de matrices. Al analizar las recomendaciones de este modelo, nuevamente la aparición de jugadores menos "estelares" es al menos sorpresiva y podría indicar que el modelo está capturando similaridades basadas en perfiles de rendimiento más que en reputación global. Dado que el modelo pesa el rating en base a la similaridad del equipo con los otros equipos del jugador, se esperaban jugadores de clubes parecidos en presupuesto y

liga al club que se recomienda, esperando una correlación de Pearson alta entre ellos. O en otro caso, si existe una correlación alta con algún club de otra liga, se esperaba que muchas de las recomendaciones provengan de jugadores exitosos en ese otro club. Sin embargo, se observan jugadores muy variados, de distintas ligas y países en las recomendaciones de un mismo club. Un ejemplo de esto es el Milan: Carter-Vickers y Schmeichel desarrollaron la mayor parte de su carrera en inglaterra y actualmente juegan en Celtic (Escocia). Taskiran jugó mayormente en el fútbol turco, mientras que Luckassen lo hizo mayormente en el futbol holandés, belga y alemán. Por último, Caleb Okoli jugó en su mayoría en el fútbol italiano.

Sin embargo, una excepción a esto es justamente el Racing de Santander, donde todas sus recomendaciones provienen de jugadores de las ligas españolas. Una posible explicación podría ser que en ligas inferiores la mayoría de los traspasos de jugadores se hacen dentro de la misma liga, generando más correlación entre equipos de la misma.

La comparación entre este modelo y el timeSVD++ es un claro ejemplo de que error más bajo no implica recomendaciones más útiles; ya que si bien no sabemos si las recomendaciones del modelo de similaridad son buenas, podemos decir que nos aportan algo más de información y son más interesantes de estudiar que las del timeSVD++.

6.2.5. Modelo pred_item_vector_model

Este modelo, que aprende pesos de interacción global entre equipos (w_{ef} y c_{ef}), tiende a generar recomendaciones con ratings más acotados y posiblemente más realistas numericamente en comparación con el modelo de similaridad anterior. Recordemos que las recomendaciones de este modelo se filtraron debido a la recomendación de jugadores que ya habían pasado por el club.

Al igual que el caso anterior, en las recomendaciones observamos muchos futbolistas que no pertenecen a la élite. Además, las recomendaciones para un equipo por lo general no provienen de su misma liga, o de un club similar, ya sea en presupuesto o en popularidad de una liga similar, como tal vez se esperaba en un principio. Esto es muy interesante proveniendo del modelo con error más bajo, lo que invita a estudiar y analizar los pesos del modelo por separado, pudiendo brindar información de similaridad entre clubes no trivial. Sin embargo, esa investigación quedará por fuera del marco de esta tesis.

6.3. Análisis de la Distribución de Parámetros y Componentes de Predicción

Más allá de las métricas de error globales, resulta instructivo analizar cómo las diferentes componentes de los modelos contribuyen a la predicción final y la distribución de los ratings predichos en comparación con los reales. Este análisis puede ser muy útil, revelando sesgos inherentes en los modelos, la influencia relativa de los factores latentes

versus los efectos base, y ayudando a interpretar distintos comportamientos en las recomendaciones. Para ello, se seleccionaron tres modelos representativos: las dos versiones del SVD Básico (`pred_basicSVD v1` y `pred_basicSVD v2`, que difieren en su regularización) y el modelo `TimeSVD++ simplificado`, que mostró un comportamiento no deseado en sus recomendaciones.

6.3.1. Distribución de los ratings predichos de los modelos

Una herramienta útil para visualizar la distribución de un conjunto de datos, combinando características de un box plot con una estimación de la densidad del kernel son los Violin Plots. En las Figuras 6.1, 6.2 y 6.3, se comparan las distribuciones de los ratings reales del conjunto de test con las distribuciones de los ratings predichos en ese conjunto por cada uno de los tres modelos seleccionados, cuando los mismos fueron entrenados únicamente en el conjunto de entrenamiento.

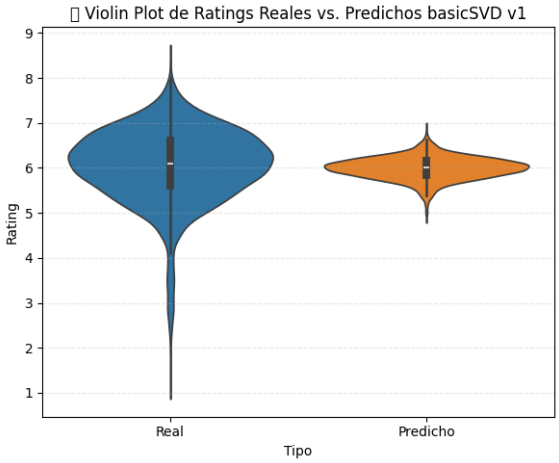

Fig. 6.1: Violin Plot de Ratings Reales vs. Predichos para el modelo `pred_basicSVD v1`.

A pesar de las diferencias claras en las recomendaciones (recordemos que se quitó regularización a los factores latentes del `pred_basicSVD v1` debido a que las recomendaciones reales eran muy similares para todos los equipos), la comparación entre los gráficos de `pred_basicSVD v1` y `v2` no muestra grandes diferencias más que unos ratings algo más alejados de la media por parte del segundo. El caso del `TimeSVD++ simplificado` (Figura 6.3) es particularmente interesante. Al igual que el caso del `pred_basicSVD v1`, sus predicciones son muy similares para todos los equipos. Sin embargo, en este caso esto se refleja claramente en un violin plot extremadamente angosto para los ratings predichos, mostrando que sus predicciones están muy cerca de la media. Esto muestra por qué el modelo no aporta información a la hora de analizar las recomendaciones.

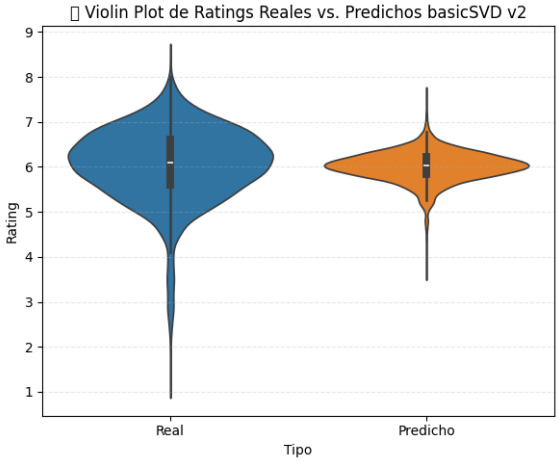

Fig. 6.2: Violin Plot de Ratings Reales vs. Predichos para el modelo `pred_basicSVD v2`.

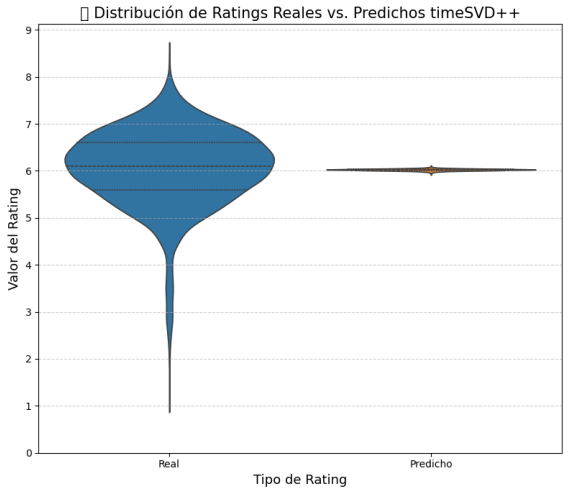

Fig. 6.3: Distribución de Ratings Reales vs. Predichos para el modelo `TimeSVD++ simplificado`.

6.3.2. Descomposición de la Predicción para un Jugador Específico

Un beneficio de estos métodos de filtrado colaborativo es el hecho de que permiten analizar los valores de los parámetros para entender mejor cómo se componen las predicciones particulares. Esto ayuda mucho a la interpretabilidad y explicabilidad de las recomendaciones y es útil para tomar decisiones comprendiendo la potencia de la herramienta y utilizándola de manera adecuada.

En este caso se analiza el desglose para Cristiano Ronaldo, un jugador representativo, recomendado en varias ocasiones por distintos modelos, en el contexto del equipo Racing de Santander para la temporada 2025.

Modelo `basicSVD v1:` Como se describió en la Sección 4.3.2, este modelo utiliza sesgos estáticos y una regularización estándar para los factores latentes.

```
Descomposición de predicción para:
Jugador: C. RONALDO
Equipo : Racing
Componentes del modelo:
 (media global)      : 6.0275
alpha_u (sesgo jugador)   : 0.5671
beta_i (sesgo equipo)    : 0.0519
pu · qi (producto dot): -0.0114

Predicción total   : 6.6351
```

Modelo `basicSVD v2:` Esta variante, también descrita en la Sección 4.3.2, modificó la regularización de los factores latentes (disminuyéndola a 0.0025) con la intención de aumentar su influencia relativa.

```
Descomposición de predicción para:
Jugador: C. RONALDO
Equipo : Racing

Componentes del modelo:
 (media global)      : 6.0275
alpha_u (sesgo jugador)   : 0.2482
beta_i (sesgo equipo)    : -0.0311
pu · qi (producto dot): 0.2124

Predicción total   : 6.4569
```

En la primera versión, el sesgo del jugador ($b_u = 0{,}5671$) y el sesgo del equipo ($b_i = 0{,}0519$) son los principales contribuyentes a la predicción. La interacción de factores latentes ($pu \cdot qi = -0{,}0114$) tiene un impacto casi despreciable en este caso particular. Con la regularización de los factores latentes disminuida en la versión 2, se observa un cambio significativo. El sesgo del jugador ($b_u = 0{,}2482$) es menor que en la versión 1, y el sesgo del equipo ($b_i = -0{,}0311$) incluso se vuelve ligeramente negativo. Crucialmente, la contribución del producto de los factores latentes ($pu \cdot qi = 0{,}2124$) es ahora mucho más sustancial y positiva. Esto explica lo que se ve en las recomendaciones: en la versión 1, los sesgos dominan las predicciones y por lo tanto todos los equipos obtienen las mismas recomendaciones, ya que no hay casi interacción entre equipos y jugadores; Por el otro lado, la versión 2 sacrifica un error más bajo a cambio de recomendaciones personalizadas para los equipos y jugadores, dando más peso en la recomendación a las interacciones entre ellos a través de los factores latentes.

Modelo `TimeSVD++ simplificado`: Este modelo (Sección 4.4.3) incorpora factores latentes de jugador que evolucionan con el tiempo, junto con dinámicas temporales más complejas para los sesgos.

```
--- Desglose del Rating para C. RONALDO  al Racing---
 (media global)             :   6.025963 |  98.73%
alpha_u (sesgo jugador)          :   0.020194 |   0.33%
_u*dev_ut (efecto t-dev jug):   0.036403 |   0.60%
  (_u=0.0266, dev_ut=1.3708)
alpha_u_bin (sesgo jug-bin)      :   0.025949 |   0.43% (idx: 6980)
beta_i (sesgo equipo)           :  -0.000034 |  -0.00%
beta_i_bin (sesgo equipo-bin)   :  -0.005351 |  -0.09% (idx: 3725)
Interacción (q_i · z_u)      :   0.000117 |   0.00%
Predicción final    :   6.103242
```

Por último, El desglose para el modelo `TimeSVD++ simplificado` es revelador. La media global (μ) constituye la abrumadora mayoría de la predicción (98.73 %). Esto explica por qué las recomendaciones de este modelo eran tan similares para todos los equipos. Este es un claro indicio que, para esta configuración y con estos datos, el modelo no está logrando capturar interacciones personalizadas significativas, y podría estar sufriendo de un problema de cantidad insuficiente de datos para la complejidad de las dinámicas temporales que intenta modelar.

6.4. Análisis de resultados con métricas de ordenamiento

A la hora de analizar la tabla 5.1, resulta interesante resaltar que las métricas de orden no se ven ordenadas, a diferencia del MAE que, salvo algunas excepciones, se ordena junto

al RMSE. Mientras que los modelos que utilizan el XGBoost obtienen claramente los mejores valores de estas métricas, los modelos de filtrado colaborativo son muy variables y modelos como el `basicSVD v2` o el de similaridad muestran resultados comparables con los de modelos con un MAE menor, como los Baselines básicos o el `TimeSVD++`.

Justamente estos modelos son aquellos que, a pesar de su alto error, aportan más información a través de sus recomendaciones que modelos con un error menor. Esto nos da indicio de que estas métricas pueden ser útiles para estudiar cuánto sirve un modelo más allá de su error predictivo.

Distinto sucede con los resultados de blending con estas métricas (tabla 5.2). En este caso, el ordenamiento dado por el RMSE es relativamente similar para las métricas de ordenamiento. Es posible entonces que esta diferencia se deba a que estas métricas nos dan indicios sobre la capacidad de generalización de cada modelo individual, por lo que en el proceso de blending, al capturar múltiples perspectivas a la vez, se genera un ranking final más estable y alineado con su error predictivo.

6.5. Comparación con métodos tradicionales

Comparar directamente los modelos de filtrado colaborativo con el scouting tradicional es complejo, ya que operan de maneras fundamentalmente diferentes. El scouting humano se basa en la observación detallada, la evaluación de habilidades técnicas, tácticas, físicas y mentales, y la consideración de factores contextuales y de personalidad que son difíciles de cuantificar. Su fortaleza radica en la profundidad del análisis individual y la experiencia humana.

Los modelos de CF, por otro lado, se basan en patrones estadísticos extraídos de grandes volúmenes de datos de rendimiento pasado. Su fortaleza radica en la capacidad de procesar información a una escala inmanejable para humanos y descubrir relaciones no obvias, difíciles de encontrar con otros métodos, como vimos con el caso del XGBoost.

En cuánto a otros modelos mencionados en la sección 1.2 como el caso de redes neuronales profundos u ootros modelos de aprendizaje automático más avanzado, la comparación resulta difícil debido a que no se cuenta con los datos y los indicadores que esos modelos utilizan, sumado al soporte computacional que requieren. Sin embargo, los modelos de CF ofrecen una perspectiva diferente. Además, como vimos, pueden ser una mejor herramienta de análisis a la hora de incorporar fichajes, debido a que en muchos casos los modelos más complejos pierden interpretabilidad.

La visión más realista es que los modelos de CF no reemplazan ningún método de scouting o sistema de recomendación, sino que lo complementan y potencian.

6.6. Limitaciones, sesgos y su impacto

Es fundamental reconocer las limitaciones y los posibles sesgos de este estudio. Comprender estas limitaciones es crucial para utilizar la herramienta de manera correcta y en algunos casos puede ser el primer paso para abordarlas en trabajos futuros:

- **Calidad y Naturaleza de los Datos de Rating:**
  - *Pocos Datos Relativos:* Aunque se trabajó con decenas de miles de registros, el universo de jugadores y equipos profesionales es mucho mayor. La dispersión de la matriz sigue siendo un desafío, especialmente para modelar dinámicas temporales complejas o interacciones muy específicas.
  - *Sesgo de Supervivencia/Visibilidad:* Los datasets probablemente sobrerrepresentan a jugadores que han tenido carreras más largas o que han jugado en equipos más visibles (de donde es más fácil obtener datos). Esto podría dificultar la recomendación de talentos emergentes de ligas menores y jugadores que se mantienen en clubes.
  - *Definición del Rating:* El rating es un promedio de rendimiento en una temporada. Esto suaviza las fluctuaciones intra-temporada y no captura el impacto de lesiones, cambios de rol a mitad de temporada, etc. La fuente y metodología de cálculo de estos ratings también pueden introducir sus propios sesgos.
- **Limitaciones del Modelado:**
  - *Sesgo por Jugadores Actuales (Dataset BeSoccer):* Si el dataset tiene únicamente jugadores actualmente activos, puede haber un sesgo que dificulta un análisis histórico profundo de la evolución de los clubes a lo largo de muchas décadas, perjudicando la idea del modelado de dinámicas temporales.
  - *Filtro por Múltiples Equipos:* Al requerir que los jugadores hayan pasado por varios equipos para densificar la matriz, se excluyen jugadores jóvenes con poca trayectoria o aquellos que se mantuvieron en un solo club. Esto puede sesgar las recomendaciones hacia jugadores con más experiencia o más abiertos a cambiar de equipo. El modelo, por tanto, seguramente tienda a recomendar jugadores más veteranos.
  - *Supuestos de los Modelos de CF:* Los modelos asumen que el rendimiento pasado es predictivo del futuro y que los patrones de interacción son significativos. Sin embargo, es posible que no se pueda (o que sea demasiado difícil) modelar el problema de la recomendación de jugadores a equipos mediante técnicas de filtrado colaborativo.
  - *Ausencia de Contexto Externo:* Los modelos implementados son puramente colaborativos y no incorporan información contextual explícita como el estilo

táctico del entrenador, el estilo de juego del jugador en el equipo, la complementariedad con otros jugadores del equipo, la liga, etc. Esta información es crucial en el mundo real y es muy tenida en cuenta por otros métodos de scouting.

7. CONCLUSIONES Y FUTUROS PASOS

7.1. Conclusiones finales

Este trabajo se propuso adaptar y evaluar técnicas de filtrado colaborativo, inspiradas en las soluciones del Netflix Prize, para la compleja tarea de recomendar jugadores a equipos de fútbol. Los resultados obtenidos, aunque preliminares en muchos aspectos, ofrecen varias conclusiones.

En términos generales, estamos ante un problema difícil, donde existe poco variabilidad para explicar debido a que la media es buen predictor. Aunque superando a los benchmark clásicos, los modelos no muestran ganancias suficientes en error como para confirmar que explican la variabilidad restante. Sin embargo, en un problema con datos tan poco dispersos, ver que modelos con más varianza explican lo mismo que modelos muy cercanos a la media en sus predicciones da indicios de que el poco porcentaje de error que mejoran los modelos se puede explicar de diferentes formas. Por este motivo, puede ser útil analizar y estudiar las recomendaciones de estos modelos para entender factores que expliquen esa variabilidad.

Otra conclusión recurrente es la tensión entre la complejidad del modelo y el riesgo de sobreajuste, especialmente en un contexto con datos limitados para la gran cantidad de parámetros que algunos modelos podrían requerir. Modelos más simples, como el `basicSVD` o el `Item_Vector_Model`, mostraron un rendimiento robusto, a veces superando a sus contrapartes más elaboradas. Esto sugiere que, con el volumen de datos actual, capturar correctamente los sesgos base y las interacciones latentes fundamentales es prioritario. Modelos más complejos pueden requerir una mayor cantidad de datos, así como ajustar cuidadosamente las parametrizaciones y regularizaciones para lograr modelar correctamente el ruido de los factores temporales.

El **blending** se confirmó como una estrategia poderosa, donde la combinación de información proveniente de estadísticas futbolísticas clásicas y de las predicciones de los modelos de filtrado colaborativo arrojó los mejores resultados globales en términos de error. Esto subraya que ambos tipos de información son complementarios y que los modelos de filtrado colaborativo podrían aportan una perspectiva distinta a otros enfoques.

Es crucial enfatizar que un menor error (e.g., RMSE) no siempre se traduce directamente en un mejor modelo desde una perspectiva práctica. El análisis cualitativo de las recomendaciones reveló que diferentes modelos, incluso con errores similares, pueden tener estilos muy distintos. Además, mostró que modelos con error razonable como el `TimeSVD++_simplified`, pueden ser modelos inútiles ya que para los distintos equipos recomiendan los mismos futbolistas.

Esto abre la discusión sobre qué se busca exactamente en una recomendación y cómo definir métricas que capturen mejor la utilidad real para un director deportivo. Las métricas de ordenamiento propuestas (Spearman y Kendall) mostraron ser un complemento interesante a las métricas de error clásicas, brindando en algunos casos más información sobre el verdadero aporte y utilidad de un modelo que estas últimas. De este modo, se subraya la necesidad de una evaluación multifacética de los sistemas de recomendación y de seguir investigando y desarrollando métricas que capturen de manera más fidedigna su utilidad práctica.

Finalmente, todos los modelos implementados, desde los predictores base hasta los ensambles más complejos, tienen un potencial considerable y representan un punto de partida para exploraciones futuras. A medida que la disponibilidad de datos y registros en el fútbol continúe creciendo seguramente mejorarán también los resultados de estos métodos. Este trabajo buscó principalmente introducir y validar la metodología en este nuevo dominio, y los resultados son, al menos, interesantes y alentadores para continuar la investigación en un ámbito que genera tanta pasión en todo el planeta como es el fútbol y el deporte en general.

7.2. Aportes metodológicos y prácticos de la tesis

Los principales aportes de esta tesis pueden resumirse en:

1. **Adaptación de Modelos de CF al Dominio del Fútbol:** Se consiguió adaptar y aplicar un amplio espectro de técnicas de filtrado colaborativo, originalmente popularizadas en dominios como la recomendación de películas o productos a usuarios, al desafío de la recomendación de jugadores de fútbol.

2. **Implementación Personalizada y Exploración:** Se desarrolló e implementó desde cero la mayoría de los algoritmos, lo que permitió una comprensión profunda de su funcionamiento y la flexibilidad para experimentar con variantes. Se exploraron diferentes arquitecturas de modelo y diferentes métodos para la incorporación de dinámicas temporales.

3. **Análisis Comparativo Exhaustivo:** Se realizó una evaluación comparativa utilizando métricas clásicas (RMSE, MAE, PMAE) sobre un conjunto de datos del mundo real, proporcionando benchmarks y evaluando el rendimiento de diferentes enfoques. Además, se propusieron y se mostraron útiles otras métricas (Spearman, Kendall) para evaluar a los modelos.

4. **Validación del Blending:** Se corroboró al blending como una técnica que generaliza y estabiliza las predicciones, mostrando que la combinación de predictores diversos y de diferente naturaleza mejora la precisión.

5. **Análisis Cualitativo de Recomendaciones:** Se fue más allá de las métricas numéricas al generar y analizar recomendaciones concretas para equipos reales, lo que permitió una primera aproximación a la interpretabilidad y utilidad futbolística de los modelos. Esto es un paso importante para conectar la investigación algorítmica con aplicaciones prácticas.

6. **Identificación de Limitaciones y Direcciones Futuras:** El trabajo identifica las limitaciones inherentes a los datos y los modelos, sentando las bases para futuras investigaciones y mejoras.

7.3. Futuros pasos: Posibles líneas de continuación de la investigación y breve motivación de cada una

La investigación presentada en esta tesis abre numerosas vías para trabajos futuros. A medida que se profundizaba en el estudio, surgían constantemente nuevas ideas y experimentos posibles, lo que refleja la riqueza y complejidad del dominio. Algunas líneas prometedoras que se busca destacar son:

- **Mejora de la Incorporación de Dinámicas Temporales:**
  - *Funciones Temporales más Sofisticadas:* Explorar funciones más flexibles para los sesgos para capturar trayectorias de carrera más complejas y adaptarse mejor a cada caso. Esto incluye un análisis del dominio y de los desempeños en las trayectorias de los jugadores.
  - *Modelado de Eventos Discretos:* Incorporar el impacto de eventos específicos drásticos (e.g., cambio de entrenador, lesión grave, ascenso/descenso de categoría del equipo) en los parámetros temporales.

- **Análisis Profundo de Hiperparámetros y Regularización:**
  - *Grid Search Exhaustivo:* Realizar búsquedas más amplias y sistemáticas para los hiperparámetros clave (dimensiones de embedding, tasas de aprendizaje, factores de regularización, número de épocas), especialmente para los modelos más complejos. En esta investigación se dejaron fijos muchos hiperparámetros pero, como vimos con el caso de la regularización en el basic SVD, tienen un impacto directo en los resultados y en las recomendaciones.

- **Incorporación de Atributos Adicionales y contexto:**
  - *Características de Jugadores/Equipos:* Integrar explícitamente atributos de jugadores (posición detallada, pierna hábil, características físicas, valor de mercado de Transfermarkt) y equipos (estilo de juego, presupuesto, calidad de la

liga) en los modelos, por ejemplo, mediante técnicas de factorización híbrida o como features adicionales en el blending. Se podrían probar embeddings pre-entrenados de texto (si se dispone de descripciones de jugadores/equipos) o crear embeddings particulares para el caso del fútbol mediante redes neuronales para aprender representaciones de estos atributos.

- *Modelado de Interdependencias entre Jugadores:* Una diferencia fundamental de este caso con la recomendación de películas es que muchas veces el rendimiento de un jugador depende mucho de sus compañeros de equipo, tanto por la química que puede tener, como la relación con o el nivel de estos. Investigar formas de incorporar este contexto de la plantilla en el modelo sería un avance significativo. Se podría explorar la adaptación de ideas como los vectores extra de los equipos pero de jugadores actuales en el equipo, donde se considere alguna métrica de la interacción entre el jugador candidato y los jugadores clave ya existentes en el equipo.
- *Redes de Jugadores/Equipos (Grafos):* Utilizar técnicas de similaridad de grafos para capturar relaciones más complejas entre jugadores (e.g., compañeros de equipo frecuentes, nacionalidad) o equipos (e.g., rivalidades, compañeros en común) y aplicarlo en modelos de vecindad.

- **Profundización en la Interpretabilidad y Análisis Cualitativo:**
  - *Análisis Semántico de los Embeddings:* **Analizar qué representan futbolísticamente las dimensiones de los embeddings (factores latentes)** aprendidos por los modelos de factorización, especialmente en los casos de baja dimensionalidad. ¿Se correlacionan con estilos de juego, roles tácticos o el paso por ciertas ligas?
  - *Estudios de Caso Detallados de Jugadores:* **Realizar un análisis más profundo de las recomendaciones para jugadores y específicos**, contrastando las predicciones con la opinión de expertos o scouts, para entender mejor las fortalezas, debilidades y posibles sesgos de cada modelo en escenarios futbolísticos concretos.

- **Integración con Criterios del Mundo Real y Aplicabilidad Práctica:**
  - *Desarrollo de Filtros Post-Recomendación Basados en Características:* A partir de las recomendaciones "crudas" generadas por los modelos, implementar un sistema de filtrado que permita a los usuarios finales refinar las listas según criterios prácticos como el precio de mercado del jugador (e.g., utilizando datos de Transfermarkt), su posición principal y secundaria, edad, nacionalidad, o incluso estadísticas de rendimiento clave (cantidad de goles, asistencias mínimas

deseadas). Esto es crucial para traducir las predicciones algorítmicas en información accionable y alineada con las necesidades y restricciones específicas de un club.

- **Métricas de Evaluación y Funciones Objetivo Orientadas al Ranking:**
  - *Optimización Directa para Ranking:* Explorar funciones de pérdida que optimicen métricas de ranking en lugar del RMSE, con el objetivo de mejorar directamente la calidad de los ranking "top-N".
  - *Métricas de Diversidad y Novedad:* Desarrollar e incorporar métricas que evalúen no solo la precisión, sino también la diversidad y la originalidad de las recomendaciones, para evitar sugerir siempre a los jugadores más obvios.

Bibliografía

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005. URL: https://doi.org/10.1109/TKDE.2005.99.

[2] American Soccer Analysis. Clustering. American Soccer Analysis Blog, mar 2020. URL: https://www.americansocceranalysis.com/home/2020/3/3/clustering.

[3] AnalyiSport. What can data do for a football club? the case of brentford fc. AnalyiSport Insights, 2022. URL: https://analyisport.com/insights/what-can-data-do-for-a-football-club/.

[4] Abdessatar Ati, Patrick Bouchet, and Roukaya Ayachi Ben Jeddou. Using multi-criteria decision-making and machine learning for football player selection and performance prediction: A systematic review. *Data Science and Management*, 6:18–30, 2023. Pre-proof version available at https://hal.science/hal-04281291v1. doi:10.1016/j.dsm.2023.11.001.

[5] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 2007. doi:10.1145/1345448.1345465.

[6] Robert M. Bell, Yehuda Koren, and Chris Volinsky. Netflix prize documentation (progress prize 2007). Netflix Prize, 2007.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[9] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The annals of statistics*, 29(5):1189–1232, 2001.

[10] Simon Funk. Netflix update: Try this at home. [Blog post], 2006. URL: http://sifter.org/~simon/journal/20061211.html.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.

[13] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2008. doi:10.1145/1401890.1401944.

[14] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456. ACM, 2009.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009. doi:10.1109/MC.2009.263.

[16] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[17] Felipe Mara-Empinotti, F. E. Fontana, Sue Nimphius, and Antonio Tessitore. From absolute to individual speed thresholds in football. *SportRxiv*, 2024. Preprint server.

[18] Adrian Mendez-Domenech, Athos Trecroci, Andrea Lamberti, Lorenzo Cavaggioni, Andrea Rossi, and G. Alberti. Football analytics: Assessing the correlation between workload and performance. *Applied Sciences*, 14(16):7217, 2024. doi:10.3390/app14167217.

[19] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of the KDD cup and workshop*, pages 5–8. ACM, 2007.

[20] Antonio Piña-Albo, Juan Miguel Pérez-Sánchez, Mario Martínez-Gómez, Javier Sampedro, and Jacobo Gómez-Romero. Beyond goals: A comprehensive plus-minus player rating in soccer using expected threat and gini impurity. arXiv preprint arXiv:2407.17832, 2024. arXiv:2407.17832.

[21] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, Boston, MA, 2011.

[22] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001.

[23] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[24] Andreas Toscher, Michael Jahrer, and Robert M. Bell. The bigchaos solution to the netflix grand prize. Netflix Prize Documentation, 2009.

[25] David H. Wolpert. Stacked generalization. *Neural networks*, 1992.

[26] Łukasz Ćwikliński, Artur Kesik, Szymon Gładysz, and Tomasz Podgórski. Machine learning in predicting a soccer player's success in a new club after transfer. *PeerJ Computer Science*, 7:e364, jan 2021. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7826718/`, `doi:10.7717/peerj-cs.364`.