Abstract

Sistema de Recomendação Autoexplicativo

de Jogadores de Futebol

**Duarte Meneses Patrícia Costa**

duartemeneses@student.dei.uc.pt patriciacosta@student.dei.uc.pt

Departamento de Engenharia Informática

Inteligência Artificial Centrada no Humano - MECD

Na sociedade atual, a quantidade de dados disponível impõe que exista uma seriação da informação que chega ao consumidor final. Os sistemas de recomendação têm um papel importante nessa questão visto que apenas apresentam ao utilizador dados que lhe interessam (definidos por ele manualmente ou através de um perfil). No entanto, para os sistemas de recomendação serem verdadeiramente aceites pelos humanos, devem explicar as recomendações que fazem. Só assim um utilizador se sente esclarecido quanto às opções que o sistema lhe gerou. É com isso em vista que decidimos desenvolver um sistema de recomendação autoexplicativo de jogadores de futebol. Pretendemos que este simplifique a vida aos clubes de futebol, poupando-lhes tempo, dinheiro e recursos humanos, explicando sempre os prós e os contras de contratar determinado jogador.

**Palavras-chave –** futebol, sistema de recomendação, inteligência artificial, jogadores, explicabilidade

1 Problema e Motivação / Introdução

Atualmente, o futebol gera e faz gerar milhares de milhões de euros. No entanto, as recentes regras do fair play financeiro, sobretudo na Europa (impostas pela UEFA – *Union of European Football Associations*), fazem com que os clubes tenham de ser bem geridos e saber exatamente onde e como gastar o seu dinheiro. Não são poucas as vezes em que um clube gasta milhões de euros em contratações e depois esses jogadores ou não se encaixam na equipa ou não rendem o que era esperado. Com isto, os clubes perdem dinheiro já que não têm retorno financeiro nem desportivo com esses jogadores. Uma das razões para o falhanço nas contratações pode estar numa fraca rede de olheiros e analistas de dados.

Se um clube tivesse à sua disposição um sistema de recomendação de jogadores que lhe sugeria futebolistas com as características pretendidas/semelhantes a outros atletas, talvez o erro nas contratações reduzisse.

Imagine-se que o Benfica pretende vender o Gonçalo Ramos na próxima janela de transferências e quer um substituto com as mesmas características para que a ideia de jogo da equipa não se altere. Em vez de ter de procurar por uma alternativa vendo milhares de jogos de outros jogadores, utilizando para isso inúmeros olheiros espalhados pelo mundo (levando a um esforço gigantesco em termos de recursos humanos e um enorme gasto financeiro) como é comum, se o Benfica tiver um sistema de recomendação como o referido acima, pode apenas pedir que este lhe recomende jogadores com as características semelhantes às do Gonçalo. Deste modo, apenas tem de analisar os atletas que o sistema sugere (teoricamente, o filtro já está tão apertado que não irão perder tempo a analisar jogadores muito diferentes do pretendido).

Este é assim um processo mais rápido, eficaz e menos dispendioso financeiramente que o tradicional. Se, aliado a isto, o sistema conseguir explicar a razão para recomendar determinado jogador e aspetos negativos e positivos da contratação do mesmo, o clube teria ainda o trabalho mais facilitado. Apenas indicando as características que gostava que um jogador tivesse, o clube obteria uma lista restrita de atletas (os que mais se enquadrariam nas preferências), tendo ainda a explicação para serem estes os recomendados e, em contrapartida, algumas razões que poderiam levar a que as contratações dos mesmos não fossem bem-sucedidas. Isto facilitaria em muito o trabalho de *scouting* e de analistas de dados dos clubes, reduzindo também o tempo despendido, o gasto financeiro e os recursos humanos necessários.

2 Trabalho relacionado

Existem inúmeros sistemas de recomendação *content-based* já implementados e com enorme sucesso. É o caso dos sistemas das lojas de roupa online que recomendam artigos semelhantes aos que o cliente viu ou adquiriu. Outro sistema muito conhecido é utilizado pelo *Spotify* que gera um perfil de utilizador através da música que este ouve, as classificações que dá, … Tendo o perfil de utilizador traçado, fica mais fácil para o *Spotify* recomendar músicas que o cliente possa gostar de ouvir.

Segundo o artigo “*Content-based Recommendation Systems*” [10], o que um sistema de recomendação faz não é mais do que filtrar informação de acordo com as preferências do utilizador. Enquanto que, nos exemplos explanados acima, o sistema vai construindo perfis de utilizadores de acordo com a sua atividade, no nosso sistema, a ideia passa por ser o utilizador a introduzir as suas preferências aquando de cada recomendação. De qualquer das maneiras, o que o sistema faz é comparar (correlacionar) as preferências do utilizador com os inúmeros objetos presentes na base de dados, recomendando assim aqueles que mais se assemelham.

Posto isto, fica evidente que sempre que existe maneira de categorizar informação (com *features*), fica relativamente fácil de comparar algo. Por conseguinte, definindo um objetivo, consegue-se fazer uma recomendação, visto que, inevitavelmente numa comparação, há sempre uma entidade que se sobrepõe a outra.

É com isso em vista que decidimos desenvolver um sistema de recomendação autoexplicativo de futebolistas.

Autoexplicativo porquê? De acordo com o artigo “*Towards Controllable Explanation Generation for Recommender Systems via Neural Template*” [8], a explicabilidade num sistema de recomendação leva a que este se torne mais transparente e confiável por parte dos utilizadores, o que, por conseguinte, aumenta o seu grau de satisfação.

Há duas grandes formas de efetuar a explicação: através de *templates* pré definidos ou gerando o próprio texto. Enquanto os *templates* são mais limitativos, visto terem uma estrutura não moldável, a geração de texto, apesar de ser mais livre e, assim, poder explicar algo melhor, pode também produzir conteúdo menos útil já que não é tão controlável [8].

Posto isto, decidimos utilizar os *templates* pré definidos no nosso projeto, já que são mais fáceis de controlar e de apresentarem a informação que se pretende.

3 Objetivos -> fim da intro

O grande objetivo deste projeto passa por conseguir oferecer aos clubes uma forma de conseguirem encontrar jogadores com as características que pretendem de forma rápida, eficaz e com menos impacte financeiro. Para tal, o nosso sistema de recomendação deve ser capaz de sugerir jogadores que tenham características semelhantes às pretendidas. Aliado a isto, deve ainda explicar a razão para recomendar determinado jogador, apresentando ainda alguns pontos menos positivos da contratação desse mesmo futebolista. Por exemplo, embora tenha características semelhantes, um jogador que tenha muitas lesões não deve ser tão recomendado e o sistema deve explicar essa mesma razão. Deste modo, pretendemos que a nossa aplicação recomende 10 jogadores, ordenados por semelhança, e explique a razão para aquele jogador estar a ser recomendado, a par de alguns contras para tal contratação.

Encontramos algumas implementações de sistemas de recomendação para este tipo de problema. No entanto, a maior parte dos programas já existentes ou apenas recomendam os futebolistas, não explicando a razão para tal, ou utilizam gráficos para a questão da explicabilidade. A nossa ideia assenta em utilizar como referência as implementações existentes para a questão da recomendação. Quanto à explicabilidade, pretendemos melhorar o que encontramos adicionando explicação textual para que as razões que levam o sistema a sugerir determinado jogador (tanto prós como contras) sejam mais compreensíveis. Pensamos que a explicação textual, trará benefícios a quem irá analisar as recomendações dos futebolistas.

4 Materiais (dados incluídos) -> análise estat

No que diz respeito à questão do sistema de recomendação, iremos utilizar como referência algumas implementações presentes na Internet [1, 4, 6].

Quanto ao *dataset*, vamos utilizar os dados/estatísticas dos jogadores presentes no FIFA22 [2]. Este é um jogo de simulação de futebol baseado na realidade, pelo que os ratings dos jogadores estão de acordo com as suas qualidades e capacidades na vida real.

5 Abordagem

Decidimos utilizar Python e algumas das suas bibliotecas para implementar o nosso sistema.

Como o dataset utilizado possui alguns valores None, decidimos começar o nosso trabalho por tratar isso, bem como remover alguns dados que não nos seriam necessários e evitar que existissem nomes de jogadores (a nossa forma de os procurar no Dataframe) repetidos.

Após isto, fizemos uma pequena análise estatística dos dados à nossa disposição para termos uma melhor visualização da das idades dos jogadores, bem como a distribuição das posições e qualidade dos mesmos por liga.

De seguida, começamos a implementar o sistema de recomendação. Para calcular a semelhança entre jogadores decidimos utilizar a distância de cosseno, uma vez que é uma ótima forma de calcular a semelhança entre dois vetores com dimensionalidade elevada (ao contrário da distância euclidiana por exemplo). Já na parte da explicabilidade, a ideia passou por utilizar *templates* pré-definidos em que variáveis servem para tornar o texto coerente e explicativo.

6 Experimentação

Tal como dissemos na secção anterior, começamos o nosso trabalho por analisar o dataset e tratá-lo de modo a que o conseguíssemos utilizar como pretendíamos. Eliminamos colunas que não nos interessavam, como por exemplo as que diziam respeito ao ID do jogador no FIFA, o URL correspondente, o número da camisola que utiliza na sua seleção, entre outros.

Tendo isto, continuamos o tratamento dos dados eliminando os valores None. Ter valores nulos no nosso dataset prejudica claramente a performance do nosso sistema, uma vez que a ausência de dados impossibilita a comparação e a própria explicação do nosso sistema de recomendação autoexplicativo. Por esta razão trocamos todos os valores None por algo que nos fosse útil. Por exemplo, se um jogador não tiver clube, não tem valor na coluna correspondente. Para contornar isto, optamos por, nestes casos, colocar o valor “Free Agent” nessa coluna. Se um jogador não tem clube, também não tem liga. Deste modo, os valores na coluna correspondente são None, pelo que os trocamos por “Free Agent”. Por último, caso os jogadores não estejam emprestados, o valor que diz respeito a esse aspeto é também nulo, pelo que alteramos por “No loan”.

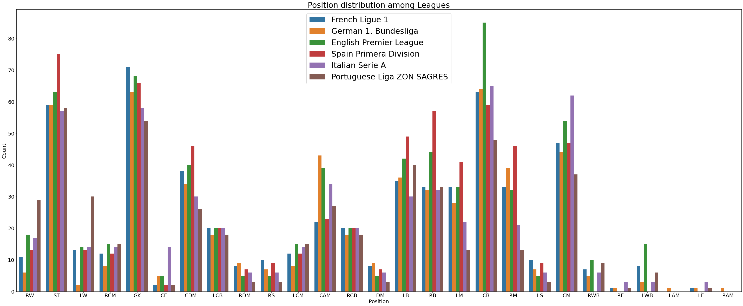
Após este tratamento, decidimos fazer uma breve análise estatística dos dados. Consideramos importante analisar a distribuição das posições e qualidade dos jogadores por liga, bem como a distribuição de idade dos mesmos. Nas análises por liga, como o nosso dataset tem imensos dados, optamos por apenas apresentar para as seis principais ligas (English Premier League, Spain Primera Division, German 1. Bundesliga, Italian Serie A, French Ligue 1 e Portuguese Liga ZON SAGRES). Os resultados foram os seguintes (para melhor visualização, enviamos as imagens em anexo):

Figura 1 - Distribuição das posições dos jogadores por liga

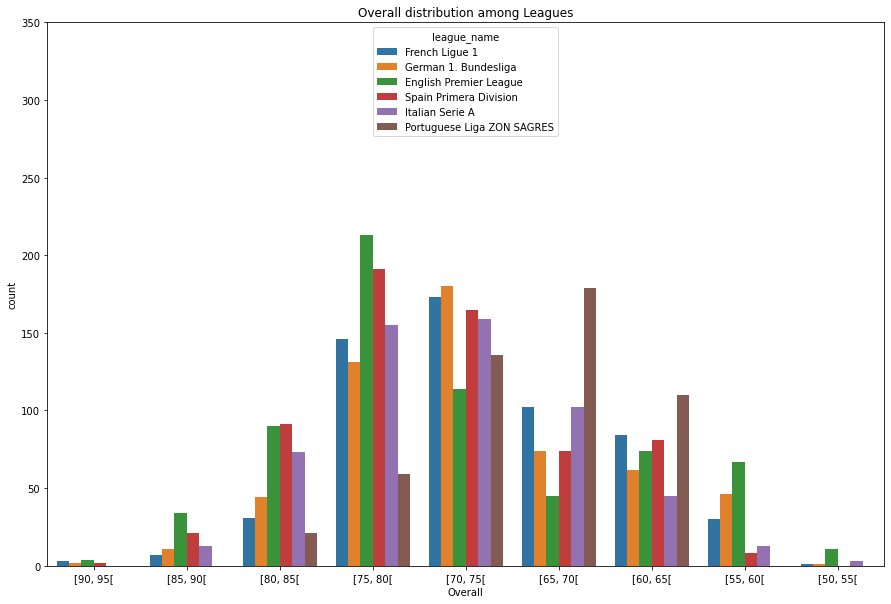


Figura 2 - Distribuição da qualidade dos jogadores por liga

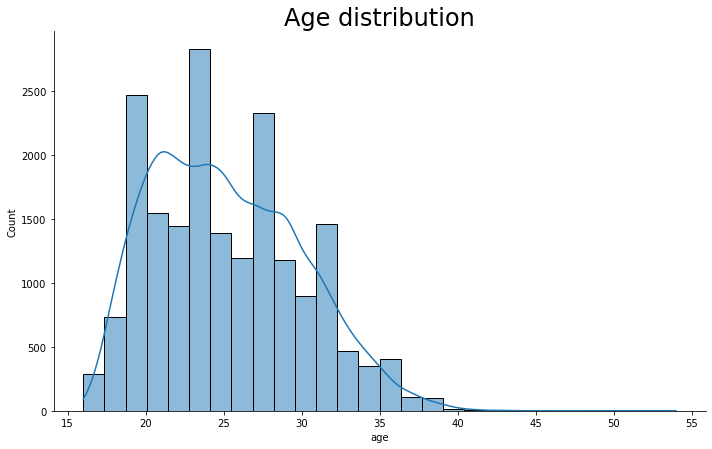


Figura 3 - Distribuição das idades dos jogadores

Pelos gráficos acima, percebemos que dentro das seis principais ligas de futebol, a qualidade dos jogadores varia, bem como as suas posições. Quanto à qualidade, percebemos que nas ligas portuguesa e italiana, não existem jogadores com qualidade superior a 89. Inclusive, na liga portuguesa, não existem jogadores com qualidade superior a 84. Isto é relevante, uma vez que, se pretendermos uma recomendação de um jogador semelhante, por exemplo, ao Cristiano Ronaldo, muito dificilmente serão recomendados jogadores destas duas ligas acima mencionadas, uma vez que não têm jogadores com qualidade semelhante.

Quanto às posições, pela figura 1 percebemos que, por exemplo, a probabilidade de um defesa central recomendado jogar na liga inglesa é maior, visto que há mais quantidade de jogadores dessa posição nessa liga. Pelo contrário, se pretendermos um avançado centro, há mais probabilidade de ser da liga espanhola.

Quanto às idades dos jogadores, fica evidente, tal como era espectável, que a maioria se encontra na faixa entre os 20 e os 31. Isto é positivo pois, em princípio, é a idade em que os jogadores estão a evoluir ou na sua melhor forma, sendo mais atrativos para contratações.

Antes de avançarmos para o desenvolvimento do sistema de recomendação, tivemos ainda de fazer mais um breve tratamento de dados. Existiam jogadores no nosso dataset que tinham posições no clube ou na seleção diferentes daquelas que eram anunciadas nas posições do jogador. Deste modo, caso essa posição não fosse suplente ou reserva, optamos por juntá-la às posições do jogador em si, uma vez que, se joga assim no clube ou na seleção, tem que ser contabilizada como posição do jogador.

Muitos dados numéricos que demonstram a qualidade do jogador em determinados atributos vinham com potencial, isto é, em vez de ter um valor fixo, vinham com uma conta (84+3, por exemplo). Isto significa que atualmente o jogador tem qualidade em de 84 mas pode, se evoluir no futuro, chegar a 87. Decidimos que no nosso sistema de recomendação iríamos considerar a avaliação com a conta feita, ou seja, neste caso, consideraríamos uma nota de 87.

Para finalizar o tratamento dos dados, resolvemos a questão dos nomes repetidos. Sendo que é através dos nomes que indexamos os jogadores no Dataframe, convém que cada identificador seja único. No entanto, percebemos que existiam jogadores com o mesmo nome. Para resolver este problema, decidimos que a melhor opção seria o identificador passar a ter o nome do jogador e a equipa em que joga (“Rui Patricio (Roma)”, por exemplo). Contudo, mesmo assim, descobrimos que existiam jogadores com o mesmo nome a jogar na mesma equipa. Com isto, o problema subsistia. Lembramo-nos que os números dos jogadores no seu clube são únicos. Com isto, definimos os identificadores com o nome e o clube e número entre parênteses, ou seja, “Rui Patricio (Roma, 1)”. Esta foi a forma por nós encontrada de indexar os jogadores no Dataframe com algo que se percebesse e única.

Tendo tudo isto tratado, estávamos finalmente aptos a desenvolver o nosso sistema de recomendação. Dissemos anteriormente que iríamos calcular as semelhanças entre jogadores através da distância de cosseno. Deste modo, necessitamos de selecionar apenas os valores numéricos (as avaliações que têm nos diversos atributos), uma vez que nenhum outro tipo de dados pode ser comparado com esta técnica. Optámos por utilizar os atributos cujos valores são textuais na parte da explicabilidade, ajudando a justificar se determinado jogador é uma boa ou má contratação. Mais à frente iremos exemplificar melhor de que forma estes dados são utilizados.

Com os dados numéricos selecionados, decidimos utilizar PCA para melhorar o desempenho do nosso sistema de recomendação, diminuindo o tempo de execução (que é elevado, já que cada jogador tem que ser comparado com todos os outros e o nosso dataset tem quase 20 mil). Este é um algoritmo que reduz o número de atributos a considerar, descartando aqueles que estão fortemente correlacionados. Analisando o gráfico de cotovelo seguinte, percebemos que 50 features são suficientes para distinguir os jogadores entre si, sem perder informação.

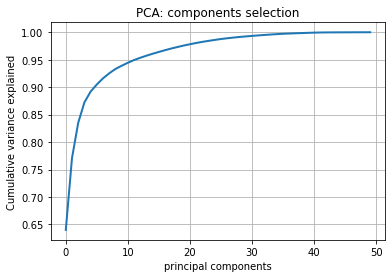


Figura 4 - Gráfico de cotovelo para seleção de features

Com isto, reduzimos a dimensionalidade do nosso problema, eliminando alguma redundância.

Nesta fase, os dados estão já prontos a serem utilizados para a comparação. Para tal, todos os jogadores são comparados

A partir dos resultados provenientes da distância de cosseno, pretendemos apresentar os jogadores recomendados, não só em tabelas, mas também em gráficos (*plots*). Os atletas deverão estar ordenados segundo o grau de semelhança entre si mesmos e as características iniciais requeridas. Com isto, tencionamos avaliar as recomendações comparando as características que se pretendiam com as dos jogadores recomendados. Esta será uma forma subjetiva de avaliar as recomendações.

Sendo que o nosso programa deve explicar a razão de recomendar um determinado jogador e apresentar alguns pontos menos positivos acerca do mesmo, vamos também avaliar se a explicação corresponde à realidade. Esta é, uma vez mais, uma forma subjetiva de avaliar a explicação. No entanto, no caso da explicabilidade, vamos também utilizar o *DoXpy* [5]: uma forma de avaliar se um texto efetivamente explicou o que devia explicar. Teoricamente, como as explicações vão advir de *templates* pré-definidos, pensamos que os resultados da avaliação serão sempre positivos, uma vez que, o programa apenas terá de introduzir no *template* as características para cada caso, o que é algo bastante controlado.

7 Resultados

8 Conclusões

Referências

[1] [Saini A. S., 2021, 30 de junho] *Building a Player Recommender Tool,* Medium, https://medium.com/analytics-vidhya/building-a-player-recommender-tool-666b5892336f acedido a 15 de outubro de 2022

[2] FIFA 22 complete player dataset. (s.d.), Kaggle: *Your Machine Learning and Data Science Community.* https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset/code acedido a 15 de outubro de 2022

[3] Author's Kit | IJCAI. (s.d.). Welcome to IJCAI | IJCAI. https://www.ijcai.org/authors\_kit acedido a 15 de outubro de 2022

[4] [mbnb8317, 2020, 5 de junho], *⚽FIFA - Recommender System - if you are manager,* Kaggle: Your Machine Learning and Data Science Community, https://www.kaggle.com/code/mbnb8317/fifa-recommender-system-if-you-are-manager/notebook acedido a 15 de outubro de 2022

[5] [GitHub - Francesco-Sovrano, (s.d.)], *DoXpy:* *For Computing the Degree of Explainability*, GitHub, https://github.com/Francesco-Sovrano/DoXpy acedido a 15 de outubro de 2022

[6] [kushojha12, (2022, 22 de maio*)], FIFA22 EDA and Feature Analysis using XGBoost*, Kaggle: Your Machine Learning and Data Science Community, https://www.kaggle.com/code/kushojha12/fifa22-eda-and-feature-analysis-using-xgboost acedido a 15 de outubro de 2022

[7] [Das S., ( 2022, 11 de Agosto)], *Beginners Guide to learn about Content Based Recommender Engine*, Analytics Vidhya. https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/ acedido a 15 de outubro de 2022

[8] [Li L., Chen L., & Zhang Y. (2020, abril)], *Towards Controllable Explanation Generation for Recommender Systems* via Neural Template | Companion Proceedings of the Web Conference 2020. ACM Conferences. https://dl.acm.org/doi/abs/10.1145/3366424.3383540?casa\_token=L3Ida-tQ7xMAAAAA:Hn3PaqwKo1mKLrQIOlikzV2kVYt4MrszDyHoFijVbBrv0fy1WWU2MVBVXX86HTUT\_w39qEzBOk0 acedido a 15 de outubro de 2022

[9] [Afchar D., Melchiorre A., Schedl M., Hennequin R., Epure E., & Moussallam M. (2008, novembro)], *Content-Based Recommendation Systems,* ResearchGate, https://www.researchgate.net/publication/236895069\_Content-Based\_Recommendation\_Systems acedido a 15 de outubro de 2022

[10] [Zisopoulos C., Karagiannidis S., Demirtsoglou G., & Antaris S. (2008, novembro)], *Content-Based Recommendation Systems*, ResearchGate, https://www.researchgate.net/publication/236895069\_Content-Based\_Recommendation\_Systems acedido a 15 de outubro de 2022

[11] [Maanijou R. & Mirroshandel S. (2019, 25 de janeiro)], *Introducing an expert system for prediction of soccer player ranking using ensemble learning* - Neural Computing and Applications, SpringerLink. https://link.springer.com/article/10.1007/s00521-019-04036-9 acedido a 15 de outubro de 2022