Insper Instituto de Ensino e Pesquisa

Relatório Final - Projeto 3

Clustering

Grupos Naturais a partir de atributos - FIFA 18

Jadson Silva Oliveira de Jesus

Luigi Portugal Gouvêa Consonni

SÃO PAULO
2018

Jadson Silva Oliveira de Jesus

Luigi Portugal Gouvêa Consonni

Relatório Final - Projeto 3

Clustering

Grupos Naturais a partir de atributos - FIFA 18

Relatório Final do Projeto 3 da disciplina de Ciência dos Dados dos cursos de Engenharia do Insper.

Professor: Fábio Ayres

SUMÁRIO

INTRODUÇÃO

Nascidos no país do futebol, fomos e somos rodeados por ele desde sempre. Daí a nossa curiosidade em abordar este tema. A partir de um dataset com informações sobre jogadores do game FIFA 2018, buscamos descobrir se é possível determinar a posição dos jogadores com base em seus atributos (Aceleração, Agilidade, Força, Energia, etc…).

Utilizamos o *Jupyter Notebook* e de uma das bases do Machine Learning, o método de 'Clustering'. Este consiste em, a partir de dados, formar grupos naturais de modo que elementos do mesmo grupo tenham certo grau semelhança em seus dados. Por fim, comparamos a capacidade de dois algoritmos de clusterização na realização dessa atividade, foram eles o K-means e o Gaussian mixtures.

OBJETIVOS

O presente trabalho tem como objetivo verificar se jogadores de futebol que jogam em posições de ataque, meio de campo ou defesa podem ser assim agrupados com base apenas em seus atributos no jogo FIFA 18. Além de avaliar também a capacidade de diferentes algoritmos de clusterização na criação desses grupos naturais.

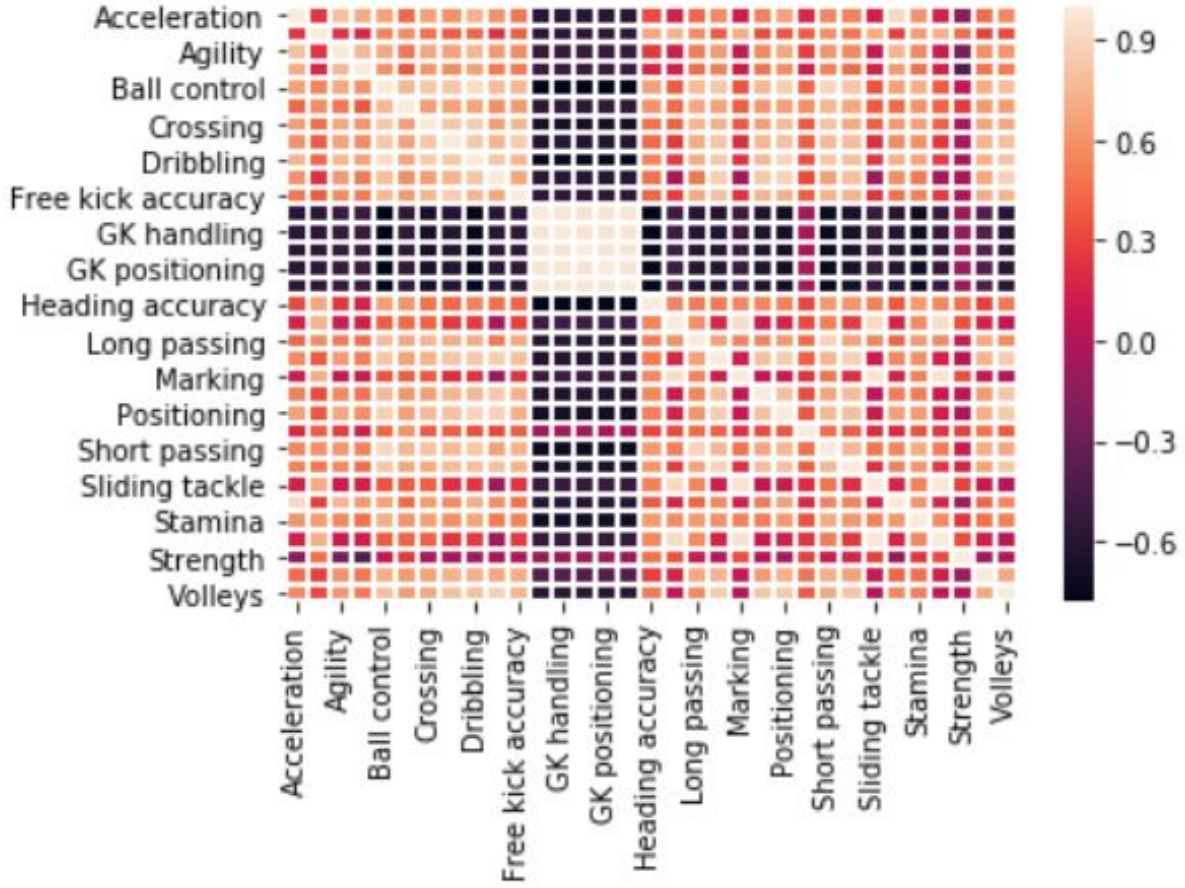
ANÁLISE EXPLORATÓRIA

O Dataset escolhido para realizar o projeto é " FIFA -18 Complete Player Database". A priori haviam muitos dados irrelevantes para a análise proposta por este projeto, assim o primeiro passo adotado foi uma limpeza dos dados retirando informações irrelevantes (Nacionalidade, Bandeira, Nome, Idade e etc…), mantendo-se apenas os atributos de cada jogador representados por números inteiros e as suas posições representadas por strings.

Após isso foi necessária a re-classificação das posições dos jogadores em quatro grandes grupos: goleiros, defesa, meio de campo e ataque.

O segundo passo foi visualizar a correlação entre os atributos, para isso utilizamos o método ".corr()" e, utilizando a biblioteca "sns", plotamos um heatmap com os atributos no qual: cores mais claras são melhor índice de correlação e cores mais escuras pior índice. Fomos capazes de concluir que os atributos referentes a goleiros são os que apresentam menor

correlação com os demais e maior correlação entre si, assim já nos indicando que a posição "goleiro" se destaca facilmente das outras.



Com o dataset ajeitado o próximo objetivo foi descobrir quais são os atributos mais relevantes para cada posição em campo. Para isso utilizamos o comando ".describe()" e, selecionando somente os jogadores da mesma posição, utilizamos o valor da mediana para revelar quais os atributos de maior destaque para uma determinada posição. Como resultados obtivemos que os dez atributos com maior mediana, em ordem decrescente, foram:

- Atk (Ataque) :

Sprint speed 73.0

Acceleration 72.0

Agility 70.0

Strength 67.0

Balance 67.0

Shot power 67.0

Finishing 65.0

Stamina 65.0

Ball control 65.0

Positioning 65.0

- Mid (Meio-Campo) :

Stamina 68.0

Acceleration 67.0

Strength 67.0

Sprint speed 67.0

Balance 67.0

Agility 67.0

Short passing 65.0

Ball control 64.0

Reactions 63.0

Dribbling 62.0

- Def (Defesa) :

Stamina 72.0

Sprint speed 72.0

Acceleration 71.0

Balance 68.0

Strength 67.0

Agility 67.0

Standing tackle 66.0

Sliding tackle 65.0

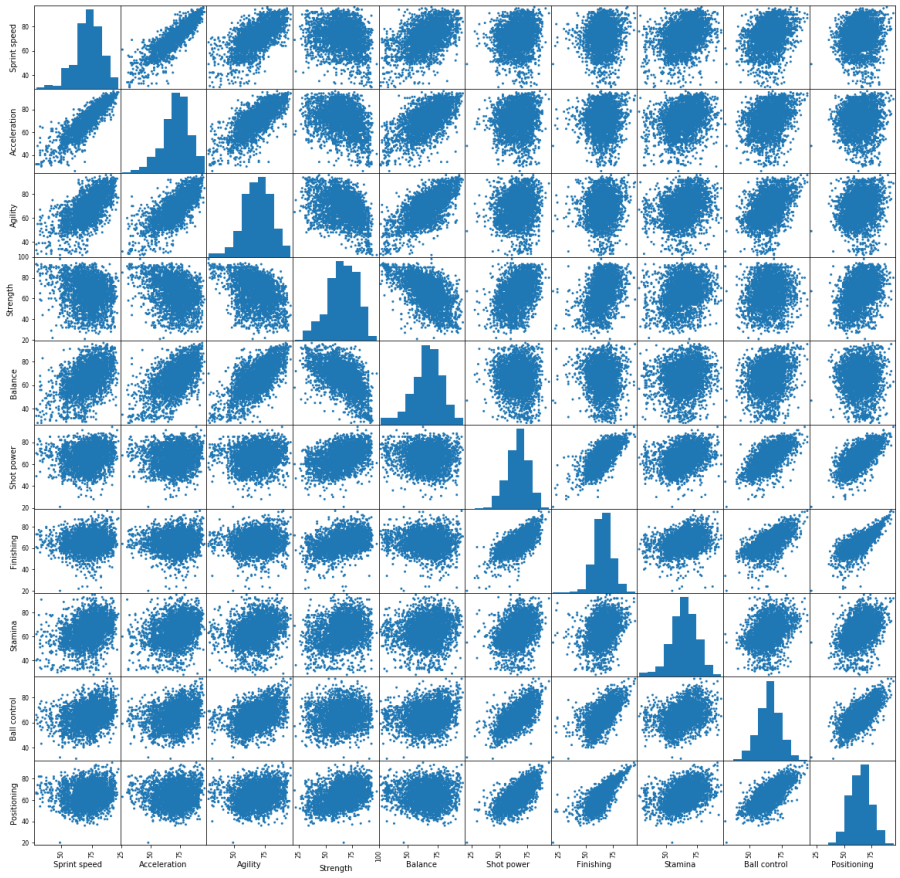
Aggression 65.0

Marking 63.0

- GK (Goleiro) :

GK reflexes 66.0
GK diving 65.0
GK positioning 63.0
GK handling 63.0
Strength 62.0
GK kicking 61.0
Reactions 60.0
Balance 43.0
Sprint speed 41.0
Acceleration 40.0

De posse dos atributos que melhor classificam cada posição, buscamos visualizar o grau de correlação entre estes atributos. Para tanto utilizamos o método ".scatter_matrix()" para plotar uma série de gráficos de dispersão relacionando somente os atributos supracitados, repetimos o processo para cada posição.



- Atk (Ataque) :
    - “Acceleration” e “Sprint speed”
    - “Finishing” e “Positioning”
- Mid (Meio-Campo) :
    - “Acceleration” e “Sprint Speed”
    - “Ball Control” e “Dribbling”
    - “Ball Control” e “Short passing”
- Def (Defesa) :
    - “Acceleration” e “Sprint Speed”
    - “Sliding” e “Standing Tackle”
    - “Marking” e “Sliding Tackle”
    - “Marking” e “Standing Tackle”
- GK (Goleiro) :
    - “Acceleration” e “Sprint Speed”
    - “GK reflexes” e “GK diving”

- “GK diving” e “GK positioning”
- “GK positioning” e “GK reflexes”

A partir dessa análise descobrimos que, dos atributos mais significativos para cada posição os representados em pares acima são os que apresentam maior grau de compatibilidade entre si. Desse modo tiramos a conclusão de que “Acceleration” e “Sprint Speed”, por serem relevantes para todas as posições, não são bons atributos para nos auxiliar no processo de “clustering”, apesar de serem importantes para o desempenho do jogador em campo.

METODOLOGIA E MATERIAIS

Após a análise exploratória, utilizamos dois algoritmos para avaliar a criação dos grupos naturais: *K-means* e *Gaussian Mixtures*. Sendo ambos classificados como algoritmos de *Expectation Maximization* (EM). O que significa que o algoritmo ajusta os valores medidos (média, covariância, desvio padrão etc) a cada nova iteração. Por fim, eles caem também na categoria de *flat clustering*, uma vez que os dados são agrupados sem o estabelecimento de uma hierarquia (*hierarchical clustering*) entre eles.

Uma observação importante é que esses métodos são utilizados para variáveis numéricas. O que, felizmente, é o nosso caso; do contrário precisaríamos fazer uma conversão de variáveis categóricas para quantitativas.

K-means

Os inputs do algoritmo K-means são um certo pacote de dados a serem clusterizados (x1, x2, x3,...,xn) e o número de clusters esperados (K). Um dos desafios associados ao uso do método é saber de antemão quantos clusters são esperados. Por sorte, no presente trabalho era de antemão esperada a formação de 4 grandes clusters, uma vez que dividimos os jogadores em 4 grupos: goleiros, defesa, meio-campo e ataque.

O algoritmo começa estabelecendo K-centróides no espaço amostral em locais aleatórios. Em seguida, para cada ponto de dado o algoritmo calcula qual dos centróides está mais próximo (distância euclidiana) e o atribui ao cluster formado pelo mesmo. Feito isso,

após todos os pontos terem sidos atribuídos ao centróides, o algoritmo restabelece os centróides atribuindo-os ao centro dos pontos naquele cluster. Ele repete esse processo até que nenhuma mudança dentro dos clusters aconteça. Dizemos então que o algoritmo convergiu. Como o algoritmo não consegue saber se o primeira distribuição de clusters é a ideal, o processo é repetido diversas vezes com diferentes pontos iniciais para os centróides. Em seguida, ele escolhe a iteração que possuir a menor variância

Gaussian mixture models(GMMs)

Esse é um método probabilístico de clusterização. Cada cluster aqui corresponde a uma distribuição gaussiana e o que algoritmo tenta descobrir são as médias e covariâncias de cada cluster. Em algum senso os GMMs podem ser visto como uma extensão das ideias por trás do K-means. Uma das limitações deste último é o fato dele forçar os dados dentro de círculos. Esse fato está associado ao fenômeno de *overlaps*, que acontece para dados distribuídos em formatos não-circulares (elipses ou oblongos, por exemplo), o que consequentemente compromete a sua capacidade de gerar os grupos naturais. Os GMMs resolvem esse problema ao calcular a medida de incerteza utilizando a distância de cada ponto até todos os centróides, ao invés de focar no mais próximo. Além disso, ele aceita que os cluster possuam formatos não-circulares. Tirando isso, os GMMs trabalham com os mesmo princípios do K-means:

1. Escolhe-se aleatoriamente um ponto de início e formato da curva
2. Repete-se o processo até a convergência:
    1. *Expectation-step*: Para cada ponto, encontra-se a probabilidade de se pertencer a cada cluster
    2. *Maximization-step*: Para cada cluster, atualiza-se a sua localização, normalização e formato com base em todos os pontos de dados e seus pesos relativos.[1]

[1] Traduzido de VANDERPLAS (2016)

RESULTADOS E DISCUSSÃO

A tabela abaixo apresenta o resultado que esperamos encontrar dentro dos grupos naturais formados caso o algoritmo utilizado e o número de clusters escolhidos forem assertivos:

Posição	Número de Jogadores
Mid	10084
Atk	3133
Def	2735
GK	2029

Utilizarmos a biblioteca "Sklearn" para analisar como seria o agrupamento dos jogadores utilizando 2, 3, 4 e 5 grupos naturais. Usamos o código "pd.crosstab()" para plotar uma tabela cruzada que compara o agrupamento feito pelo cluster (do tipo 0 ou 1) com as posições do database.

Clustering com K-means

com 2 grupos

Preferred Positions	Atk	Def	GK	Mid
saida_2				
0	3133	2735	0	10084
1	0	0	2029	0

Agrupando-os em dois grupos naturais fica claro que os goleiros (grupo 1) se destoam facilmente dos outros jogadores, como havíamos previsto com base nas conclusões retiradas do heatmap de correlação entre os atributos. Já o agrupamento referente às outras posições fica confuso, por termos limitado o cluster a somente dois grupos naturais o programa ao

analisar os atributos de todos os jogadores percebe que os goleiros são os que apresentam atributos com maior diferença em relação aos outras posições.

Um erro que a observação dos dados acima pode induzir é acreditar que dois grupos naturais são suficientes, uma vez que ao fazermos o crosstab o grupo 0 acertou exatamente a quantidade de cada jogador. Contudo, essa abordagem é errada pois o intuito do projeto é analisar se é possível agrupar os jogadores nas posições especificadas acima, do modo apresentado todos os goleiros estão no grupo 1 e os jogadores de ataque, defesa e meio-campo estão juntos no grupo 0 o que acaba provando que dois grupos naturais não serve o propósito do projeto. Além disso, o fato do número de jogadores em cada posição do grupo 0 coincidir exatamente com os números totais de cada posição é trivial, uma vez que os goleiros se diferenciam muito do restante dos jogadores, acaba sendo lógico que todos os outros estejam no grupo restante exatamente com seus números totais, pois estes não se encontram em nenhum outro grupo.

com 3 grupos

Preferred Positions saida_3	Atk	Def	GK	Mid
0	0	0	2029	0
1	3018	66	0	3411
2	115	2669	0	6673

Dando maior liberdade de agrupamento ao programa, escolhendo três grupos naturais, percebemos que os goleiros (grupo 0) se mantém em um grupo isolado, ou seja, sem nenhum outro jogador de posição diferente. Já em relação aos outros dois grupos captamos que o grupo 1 se caracterizou em jogadores de ataque e meio-campo e o grupo 2 se concretizou em jogadores de meio-campo com ênfase em defesa. Isto nos revela que ao fornecer o espaço para mais grupos naturais o programa foi capaz de criar grupos de jogadores com ênfase em outras posições.

com 4 grupos

Preferred Positions saida_4	Atk	Def	GK	Mid
0	29	1639	0	3819
1	0	0	2029	0
2	377	1051	0	3771
3	2727	45	0	2494

O clustering com quatro grupos naturais se comporta de maneira semelhante ao de três grupos naturais: ainda garantindo os goleiros isolados em um grupo, agrupando jogadores de meio-campo com ênfase em defesa nos grupos zero e dois e deixando o grupo três para jogadores de ataque, meio-campo e meio-campo com ênfase em ataque.

com 5 grupos

Preferred Positions saida_5	Atk	Def	GK	Mid
0	0	0	2029	0
1	1746	55	0	1569
2	1270	22	0	1859
3	100	1452	0	3646
4	17	1206	0	3010

Realizando o clustering com cinco grupos notamos que quanto maior o número de grupos naturais a serem utilizados mais específicos serão os grupos de jogadores. Isso está de acordo com a teoria por trás do K-means, uma vez que quanto maior o número de clusters menor é a variância total. Todavia, essa característica não nos facilita na identificação das posições, uma vez que decidimos no começo do projeto que iríamos re-classificar as várias posições dos jogadores em quatro. Portanto, um agrupamento muito específico acaba prejudicando nossas conclusões.

Clustering com Gaussian Mixtures

com 3 grupos

Preferred Positions saida3	Atk	Def	GK	Mid
0	3085	217	0	4902
1	48	2518	0	5182
2	0	0	2029	0

com 4 grupos

Preferred Positions saida4	Atk	Def	GK	Mid
0	74	1318	0	3312
1	19	1312	0	3163
2	0	0	2029	0
3	3040	105	0	3609

com 5 grupos

Preferred Positions saida5	Atk	Def	GK	Mid
0	1751	50	0	1576
1	0	0	2029	0
2	1316	81	0	2560
3	14	1187	0	2831
4	52	1417	0	3117

CONCLUSÕES

Uma vez que, por conta da natureza do problema de clusterização, não contamos com uma medida de acurácia para os diferentes algoritmos não podemos também dizer de forma análitca qual dos algoritmos é o mais adequado para os nossos dados. Por conta disso,

precisamos fazer uso de recursos gráficos e tentar de uma forma qualitativa comparar a performance dos mesmos.

Os gráficos abaixo mostram a distribuição dos jogadores . O Gráfico A foi resultante do K-means e o Gráfico B do Gaussian Mixtures. Ambos são gráficos do tipo scatter plotados em 3D, já que estes foram feitos com o intuito de representar visualmente o agrupamento feito pelos dois algoritmos utilizados, fizemos uso de 4 grupos naturais e dos atributos "Finishing", "Marking" e "Dribbling", pois a inserção de mais atributos prejudicaria a visualização do gráfico. Vale apontar que estes três atributos selecionados por nós não foram ao acaso, mas sim são uma consequência da série de "Scatter Matrix" referentes a análise exploratória dos dados, dentre os atributos que apresentaram maior correlação selecionamos um de cada posição para plotarmos os gráficos 3D.

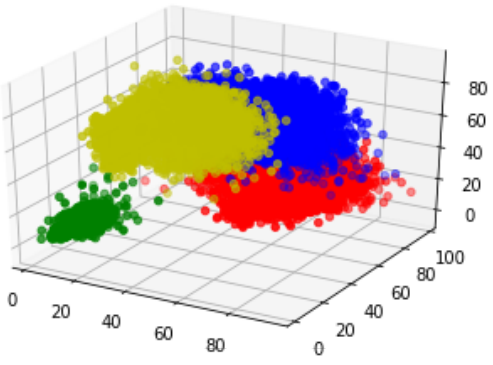


Gráfico A

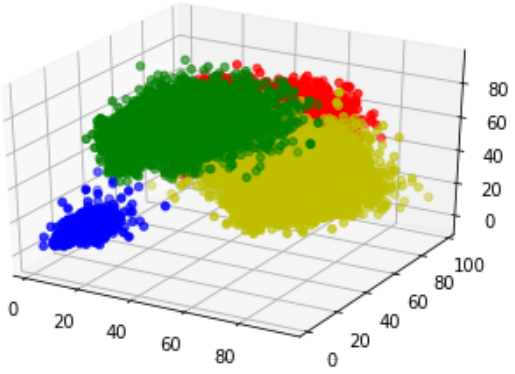


Gráfico B

Com base nas tabelas cruzadas e nos gráficos 3D de ambos algoritmos de clustering podemos inferir que o grupo representado pela cor verde (Gráfico A) e o representado pela cor azul (Gráfico B) se referem aos goleiros por estarem totalmente isolados tanto nos gráficos como nas tabelas cruzadas, para as demais cores pode-se perceber o mesmo padrão de agrupamento nos dois gráficos, os jogadores do meio-campo simbolizados pela cor azul (Gráfico A) estão presentes nos grupos de ataque e defesa, assim como mostram os números da tabela. Já no gráfico B observamos que as fronteiras do grupo natural meio-campo são significamente menos definidas em comparação com o gráfico A, isso ocorre por o Gaussian

Mixture ser um algoritmo mais refinado que o K-means (como mencionado na seção de metodologia deste relatório) e por estarmos limitando-o a somente quatro grupos naturais, assim nos devolvendo um cenário mais próximo da realidade onde, na verdade, a posição meio-campo é uma mistura de jogadores com ênfase em defesa e ataque.

Retomando a pergunta inicial do projeto "É possível determinar a posição dos jogadores com base em seus atributos?" concluímos que é possível mas com as limitações apresentadas, sendo elas: o valor K escolhido por nós e as simplificações das posições dos jogadores. Uma melhoria do projeto poderia ser a criação de mais subgrupos de comparação. Considerando, por exemplo "meio-atacante", "meio-defensores", "ponta direita" etc. Por fim, uma sugestão para trabalhos futuros é utilizar os clusters criados em algum outra classe de problema (classificação, por exemplo) e comparar a acurácia do mesmo nos casos com e sem os dados do cluster.

FONTES

Projeto disponível em: <https://github.com/jadsonsoj/ProjetoFinalCDD>

VANDERPLAS, Jake (2016). Python Data Science Handbook. O'Reilly, Publicado Online; disponível em: <https://jakevdp.github.io/PythonDataScienceHandbook/>

Mixture Models, disponível: < https://goo.gl/gfZrUL >

Clustering, disponível em: <https://scikit-learn.org/stable/modules/clustering.html>

Database, disponível em: <https://www.kaggle.com/thec03u5/fifa-18-demo-player-dataset>

Edx MOOC: *Principles of Machine Learning: Python Edition* by Microsoft

K-means, disponível em: <https://scikit-learn.org/stable/modules/clustering.html#k-means >

Gaussian mixture models, disponível em: <https://scikit-learn.org/stable/modules/mixture.html#mixture>