Football Events

Temática 3: Aprendizagem Supervisionada – Classificação

Gustavo Magalhães - up201705072@fe.up.pt
Nuno Cardoso - up201706162@fe.up.pt
Tiago Silva - up201705985@fe.up.pt
Turma 1

28/05/2020

Referências

1. Duarte, Luís Miguel da Silva. Projeto de Dissertação, 1x2 - Previsão de Resultados de Jogos de Futebol. FEUP. Porto. 29 de Junho de 2015. https://repositorio-aberto.up.pt/bitstream/10216/79327/2/35444.pdf.

2. Fernandes, Felipe Augusto Pereira. Previsão de Resultados no Futebol Por Meio de Técnicas de Aprendizado de Máquina. CEFET-MG. Belo Horizonte. Fevereiro de 2019. https://sig.cefetmg.br/sigaa/verArquivo?idArquivo=2256419&key=6a8ee9c8bd115b18683f4b40fb5ec585.

3. Olson, Randal S. Data-Analysis-and-Machine-Learning-Projects. University of Pennsylvania Institute for Bioinformatics. 3 de Julho de 2019. https://github.com/rhiever/Data-Analysis-and-Machine-Learning-Projects/blob/master/example-data-science-notebook/Example%20Machine%20Learning%20Notebook.ipynb.

4. Documentação scikit-learn, versão 0.23.1, https://scikit-learn.org/stable/search.html?q=learn.

Especificação

Através de algoritmos de aprendizagem supervisionada por classificação, pretendemos que o computador agrupe os dados dos vários eventos que ocorrem durante jogos de futebol, com o objetivo final de o capacitar a classificar jogos quanto ao seu país de origem e a determinar a equipa vencedora.

Estes eventos podem ser faltas, remates, golos entre outros, identificando-se também a zona no campo da sua ocorrência no decorrer de uma partida de futebol.

Dado o vasto e completo conjunto de dados, decidimos fazer uma análise a dois tipos de problemas, de modo a identificar qual fará mais sentido explorar. São estes:

- **Classificar o país onde ocorre o jogo** consoante o número de ocorrências de golos, faltas, cartões, penáltis, cantos, contra-ataques e número de vezes em que se rematou a bola numa dada zona do campo;
- **Classificar o vencedor do jogo ou empate** não contabilizando o número de golos, usando para isso, número de faltas, cartões, penáltis, cantos, contra-ataques e número de remates de uma dada zona do campo, isto tanto para a equipa da casa como a visitante.

Descrição das Ferramentas Usadas

Tendo em conta que a linguagem usada foi **Python na versão 3.7,** para facilitar a nossa implementação usámos as seguintes bibliotecas:

- **Numpy** - para uso facilitado de matrizes e arrays multidimensionais;
- **Pandas** - para guardar dados a analisar de modo fácil e eficiente;
- **Seaborn** e **Matplotlib** - para visualização de gráficos e análise estatística;
- **Scikit-Learn** - para auxílio na implementação dos algoritmos de machine learning;

Recorremos também ao **Jupyter** para agrupar toda a análise estatística de dados feita, assim como a sua exploração.

Pré-processamento de dados

Com programação em Python e com recurso ao conjunto de dados fornecidos, construímos dois *data sets*, de modo a que os dados de um jogo fossem agrupados numa linha de uma tabela. Deste modo, foram criados dois ficheiros .csv distintos, correspondentes às duas opções de classificação apresentadas:

1. gameEvents.csv - com dados de classificação relativos ao país onde o jogo decorreu;
2. gameEvents[HomeAway].csv - com dados de classificação relativos ao vencedor do jogo.

Dados limpos

Analisando os dados provenientes dos ficheiros *.csv* verifica-se que em algumas colunas não contabilizados quaisquer eventos e algumas linhas do *data set* de agrupamento para classificação por país não possuem a coluna “país” definida.

Assim, estas colunas e linhas foram eliminadas.

	goal	foul	corner	offside	yellow_card	red_card	penalty	fastbreak	attempt_1	attempt_2	...	attempt_10
count	9073.000000	9073.000000	9073.000000	9073.000000	9073.000000	9073.000000	9073.000000	9073.000000	9073.0	9073.0	...	9073.000000
mean	2.694258	25.668798	10.051251	4.790918	4.409457	0.137992	0.298027	0.505676	0.0	0.0	...	0.384768
std	1.681543	6.396077	3.408821	2.596853	2.267054	0.390760	0.622206	0.782156	0.0	0.0	...	0.636190
min	0.000000	3.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	...	0.000000
25%	1.000000	21.000000	8.000000	3.000000	3.000000	0.000000	0.000000	0.000000	0.0	0.0	...	0.000000
50%	3.000000	25.000000	10.000000	4.000000	4.000000	0.000000	0.000000	0.000000	0.0	0.0	...	0.000000
75%	4.000000	30.000000	12.000000	6.000000	6.000000	0.000000	0.000000	1.000000	0.0	0.0	...	1.000000
max	12.000000	51.000000	26.000000	21.000000	15.000000	4.000000	5.000000	5.000000	0.0	0.0	...	5.000000

Identificação de Algoritmos

Iremos implementar e testar o nosso projeto com os seguintes algoritmos:

- Árvore de decisão;
- K-ésimo vizinho mais próximo;
- Rede neuronal artificial;

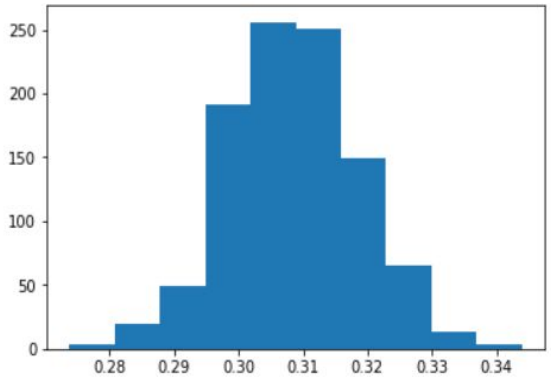
Parameter Tunning

Para otimizar o nosso algoritmo através da *grid search* oferecida pelo scikit learn, decidimos procurar as melhores *accuracies* fazendo variar os seguintes atributos:

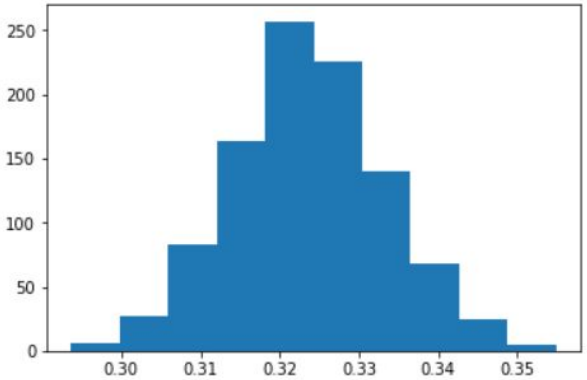
- **Árvore de decisão -** ‘criterion’ ‘splitter’ ‘max_depth’ ‘max_features’
- **K-ésimo vizinho mais próximo -** ‘n_neighbors’ ‘weights’ ‘algorithm’ ‘p’
- **Rede neuronal artificial -** ‘activation’ ‘solver’ ‘hidden_layer_sizes’

Classificação de jogos por país

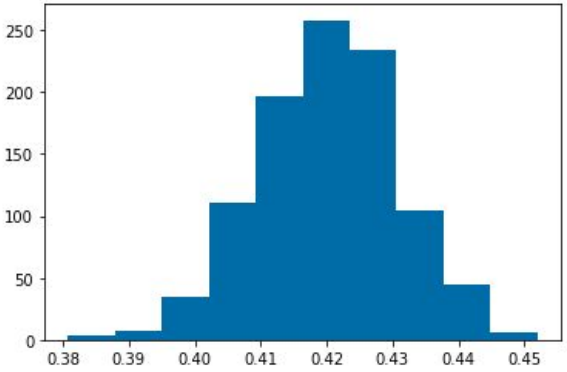
Árvore de Decisão



K-vizinhos mais próximo



Rede Neuronal Artificial



Gráficos de *accuracy* em treino treino/teste para 1000 iterações

Classificação de jogos por país

Árvore de Decisão

Default

0.3096067695370831

Tuning

Best parameters:

{'criterion': 'gini', 'max_depth': 5,
'max_features': 22, 'splitter': 'best'}

{'max_depth': 5, 'max_features': 22}

Resultado finais

Time to test/train:

0.03746294975280762 seconds

Accuracy Score:

0.3877551020408163

K-vizinhos mais próximo

Default

0.29367844698855515

Tuning

Best parameters:

{'n_neighbors': 95}

{'algorithm': 'brute', 'n_neighbors': 95,
'p': 1, 'weights': 'distance'}

Resultado finais

Time to test/train:

0.006525993347167969 seconds

Accuracy Score:

0.4066699850671976

Rede Neuronal Artificial

Default

0.4146341463414634

Tuning

Best parameters:

{'activation': 'logistic', 'solver': 'adam'}

Resultado finais

Time to test/train:

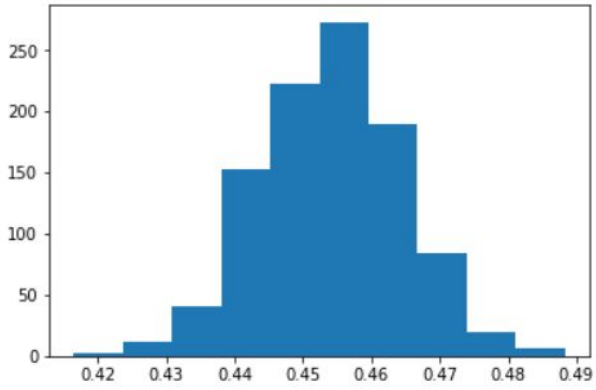
3.6135191917419434 seconds

Accuracy Score:

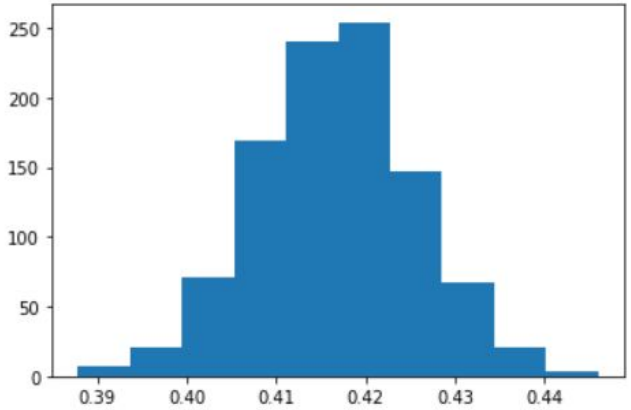
0.43653558984569435

Classificação de jogos por vencedor

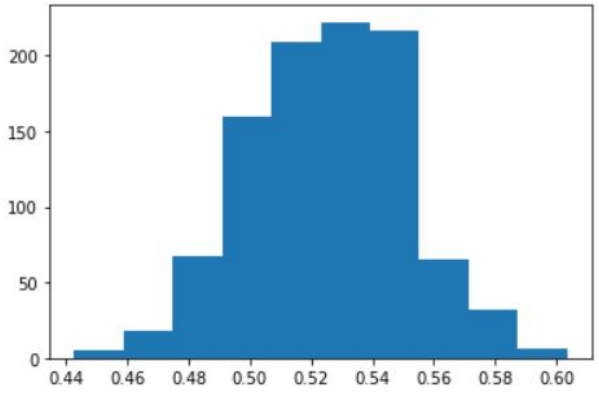
Árvore de Decisão



K-vizinhos mais próximo



Rede Neuronal Artificial



Gráficos de *accuracy* em treino treino/teste para 1000 iterações

Classificação de jogos por vencedor

Árvore de Decisão

Default

0.43940061701189953

Tuning

Best parameters:

{'criterion': 'entropy', 'max_depth': 7, 'max_features': 36, 'splitter': 'best'}

Resultados finais

Time to test/train:

0.02741408348083496 seconds

Accuracy Score:

0.501983252534156

K-vizinhos mais próximo

Default

0.36359629792860293

Tuning

Best parameters:

{'algorithm': 'brute', 'n_neighbors': 75, 'p': 1, 'weights': 'distance'}

Resultados finais

Time to test/train:

0.01480412483215332 seconds

Accuracy Score:

0.499338915821948

Rede Neuronal Artificial

Default

0.5350374614367562

Tuning

Best parameters:

{'activation': 'logistic', 'solver': 'sgd'}

Resultados finais

Time to test/train:

8.593080043792725 seconds

Accuracy Score:

0.5808726311150286

Conclusões

- Ambos os os problemas se revelaram difíceis de classificar, em grande parte, por estes valores serem algo aleatórios por se tratar de um jogo de futebol;
- Constatou-se ser mais fácil classificar o resultado de um jogo do que em que país o jogo está a ser disputado;
- O algoritmo de rede neuronal artificial revelou ser de longe o melhor. Quando otimizado para classificar países não revelou um aumento significativo.
- Os algoritmos de árvores de decisão e k-vizinho mais próximo apresentaram *accuracies* muito semelhantes, especialmente quando foram otimizados.