Desmistificando o Cartola FC

Ciro Ceissler
RA 108786
ciro.ceissler@gmail.com

Lucas de Souza e Silva
RA 140765
lucasonline1@gmail.com

Matheus Laborão Netto
RA 137019
mln.laborao@gmail.com

Ramon Nepomuceno
RA 192771
ramonn76@gmail.com

***Abstract*—"Desmistificando o Cartola FC" é o trabalho final dos autores para a disciplina MC886/MO444 (Machine Learning) na Unicamp. A escolha do tema se deu pela afinidade do grupo com o jogo e esporte, trazendo para nosso dia-a-dia mais direto uma aplicação de aprendizado de máquina. Trabalhamos com baselines desenvolvidos por brasileiros e disponibilidades no Github, conforme será referenciado mais abaixo. A disponibilização de dados pela empresa que administra o jogo facilitou a captura do banco de dados, onde pudemos desenvolver técnicas diferentes de limpeza e tratamento. Com o passar da disciplina e novas técnicas de Machine Learning serem apresentadas, aumentamos o número de aplicações de ML diferentes para resolver o problema. Apresentaremos para o leitor dados mais gerais sobre cada uma delas no relatório que segue. Obtivemos resultado, em pontos do jogo, de 55,6 com nossas predições, número esse desconsiderando o técnico do time, que caso seguisse a média obtida com os jogadores faria com quem chegássemos em aproximadamente 61 pontos. A efeito de comparação, o campeão nacional do ano passado teve média de 66 pontos. Consideramos positivo nosso resultado e aprendizado ao longo do projeto.**

I. Introdução

O futebol é um esporte com diversos fãs ao redor do mundo e com uma imprevisibilidade muito grande, supreendendo os torcedores. Um exemplo recente foi a Copa do Mundo de 2018 no qual o campeão do torneio anterior não conseguiu passar nem da primeira fase, perdendo 2x0 para Coreia do Sul apenas 57ª colocada no ranking FIFA [1], e a presença Croácia na final. Ainda nesta Copa do Mundo; diversos bancos, incluindo o *Goldman Sachs*, utilizaram este evento para demonstrar a capacidade de prever eventos complexos [2], eles chegaram a rodar milhões de variações do torneio para calcular a probabilidade de cada time avançar na competição e mesmo assim não obtiveram um resultado satisfatório.

Os fãs de futebol também participam da "brincadeira" através do Cartola FC, um "fantasy game" sobre este esporte. O Cartola FC é um jogo online fictício no qual você pode montar seu time com jogadores reais da Série A do Campeonato Brasileiro. No jogo é preciso montar seu time, escolhendo técnico, jogadores e o esquema tático. Com as moedas do jogo, inicialmente C$ 100.00 cartoletas, é possível escalar o seu time, comprando e vendendo jogadores a cada rodada, importante adicionar que a (des)valorização do jogador acontece após cada rodada e leva em consideração a pontuação do atleta, além da média dos outros jogadores. A escalação deve ser feita antes do primeiro jogo da rodada e a cada uma, os jogadores recebem pontuações baseadas em suas ações durante as partidas, e.g., gols feitos, chutes defendidos, bolas roubadas, faltas cometidas ou sofridas, cartões recebidos, entre outras. As ações dos jogadores em campo são chamadas de Scouts e são elas que geram a pontuação do time. Por fim, o Capitão tem sua pontuação duplicada e não pode ser o técnico. No Cartola FC de 2017, o jogador vencedor totalizou 2460.19 pontos ao final de todas as rodadas (média de 67.4 pontos por rodada), competindo contra 4 milhões de times escalados.

O projeto propõe um modelo preditivo para maximizar a pontuação do Cartola FC, informando a cada rodada do jogo quais parâmetros devem ser utilizados, ou seja, escolher os jogadores/técnico a cada rodada para o torneio de 2018.

II. Trabalhos Relacionados

TODO(ciroceissler): football [3] [4] [5] [6] [7]

Em [8], um sistema de previsão aplicando técnicas de *machine learning* foi proposto para conseguir a melhor escalação do time a em cada rodada, entretanto os resultados obtidos não permitiria vencer o torneio. Além disso, este trabalho fez uma análise dos dados, explanando algumas observações como: o esquema tático tem uma leve variação, sendo o 4-4-2 a melhor formação.

[9]

[10]

[11] [12]

III. Base de Dados

A base de dados [12] contém as informações sobre os campeonatos de 2014 a 2017 consolidadas e serão utilizadas para aprendizado do nosso modelo, os dados sobre 2018 são adicionadas ao término de cada rodada. No repositório, cada campeonato é dividido em cinco arquivos do formato *csv* com as seguintes informações: nome dos times, posição e time dos jogadores (considera-se o técnico um jogador com a posição de técnico), resultado das partidas do campeonato, lista dos scouts, e a tabela final da pontuação dos times no campeonato.

Abaixo, os itens abaixo complementam a Tabela I com a lista de Scouts, também sendo atualizado a cada rodada:

- **atletas.nome:** nome completo do jogador
- **atletas.apelido:** nome/apelido do jogador
- **atletas.rodada_id:** número da rodada do Brasileirão
- **atletas.clube_id:** abreviação do clube do jogador
- **atletas.posicao_id:** posição do jogador
- **atletas.clube.id.full_name:** clube do jogador
- **atletas.status_id:** status do jogador na rodada
- **atletas.pontos_num:** pontuação dos scouts
- **atletas.preco_num:** preço do jogador

Table I
TIPOS DE SCOUTS

Scouts de Ataque		Scouts de Defesa	
Gol	8.0 pts	Jogo sem sofrer gols	5.0 pts
Assistência	5.0 pts	Defesa de pênalti	7.0 pts
Finalização na trave	3.0 pts	Defesa difícil[1]	3.0 pts
Finalização defendida	1.2 pts	Roubada de bola	1.5 pts
Finalização para fora	0.8 pts	Gol contra	-5.0 pts
Falta sofrida	0.5 pts	Cartão vermelho	-5.0 pts
Pênalti perdido	-4.0 pts	Cartão amarelo	-2.0 pts
Impedimento	-0.5 pts	Gol sofrido[1]	-2.0 pts
Passe errado	-0.3 pts	Falta cometida	-0.5 pts

- **atletas.variacao_num:** variação do preço do jogador
- **atletas.media_num:** média do jogador
- **atletas.jogos_num:** quantidade de jogos disputados
- **atletas.scout:** quantidade de scouts obtidos

IV. TRATAMENTO DOS DADOS

Antes de realizar o treinamento, uma análise dos dados foi feita de maneira preliminar para identificar quais melhorias podem ser realizadas, além de eliminar inconsistências no arquivo, utilizado como entrada para o treinamento do modelo.

A. Limpeza

A primeira etapa realizada para implementar o modelo de predição foi analisar os dados fornecidos pela API do cartola, fazendo as limpezas necessárias para criar amostras corretas e relevantes para o preditor. Os dados devem continuar consistente após a limpeza dos dados, ou seja, nenhuma coluna poderá ser adicionada, alterada ou removida. Após uma análise prévia dos dados, alguns problemas foram detectados nas informações dos jogadores:

- todos os scouts com valor NaN (*not a number*).
- coluna 'ClubeID' com valor NaN.
- coluna 'Status' com valor NaN.
- pontuação não equivalente a soma ponderada dos scouts.

A coluna 'atletas.clube_id' tem campos repetidos e divergentes: por exemplo, todos os Atléticos (MG, PR, e GO) são ATL. Além disso, há jogadores com siglas diferentes das equipes que eles jogam (por exemplo, Maicosuel [id: 37851]). A coluna 'athletes.atletas.scout' não é informativa. Os scouts de 2015 dos jogadores são cumulativos, ou seja, os scouts dos jogadores vão sendo somados a cada rodada. Entretanto, a pontuação não é. Isso também causa o repetimento de dados.

As linhas que possuem as inconsistências citadas acima são removidas ou atualizadas. Além destas modificações, outras remoções de dados irrelevantes são realizadas, por exemplo jogadores que não participaram de nenhuma rodada, técnicos e jogadores sem posição, jogadores sem nome, entre outros.

B. Atualização dos scouts cumulativos de 2015

Como dito anteriomente, os dados sobre os scouts de 2015 foram disponibilizados de maneira cumulativa pela fornecidos pela API do cartola, ou seja, os scouts de uma rodada são adicionados aos scouts anteriores a cada nova rodada que um jogador participa. Portanto, foi necessário tirar essa acumulação para cada jogador, de maneira que a representação dos dados ficasse coerente.

Para isso, dada uma rodada específica, os scouts de um jogador são subtraídos do máximo dos scouts de todas as rodadas anteriores. Repare que assim há chance do scout 'Jogo Sem Sofrer Gols (SG)' ser negativo se o jogador não sofrer gols na rodada anterior e sofrer na rodada atual. Quando isso acontece, esse scout é atualizado.

C. Verificação da pontuação com os scouts

Por inconsistência na base de dados fornecida pela API do cartola, alguns jogadores possuiam pontuações que não condiziam com seus scouts. Para esses casos, o jogador é removido da base de dados para evitar qualquer tipo de ruído. Ao final, mais de 4000 jogadores foram removidos.

D. Remoção das linhas duplicadas

A última operação realizada para limpar a base de dados foi apagar as linhas repetidas. A existência de linhas repetidas deve-se ao fato de que a partir da primeira participação de um jogador no campeonato, ele aparece em todas as rodadas subsequentes, mesmo que não tenha jogado. As entradas redundantes não são necessárias ao modelo, por isso foram removidas.

E. Criação das Amostras

Continuando a implementação após a limpeza da base de dados, o próximo passo foi transformar os dados para tornar utilizáveis na criação dos modelos. Desta forma, duas operações são realizadas e descritadas abaixo:

- **Selecionar somente as colunas de interesse:** colunas como 'atletas.nome', 'atletas.foto', etc não são relevantes para criação do modelo. No entanto, colunas como o 'AtletaID' e 'atletas.apelido', mesmo que não utilizadas para treinamento do modelo, são importante para avaliar o resultado e, portanto, também serão consideradas.
- **Converter todos os dados categóricos para numéricos:** as colunas 'Posicao', 'ClubeID', 'opponent' e 'casa' serão convertidas para número.

V. TREINAMENTO

O problema se encaixa na categoria de esportes, ou seja, na , que é comumente resolvido através de um regressão linear para prever

Nesse método, todas as combinações possíveis entre os parâmetros

Entre as variações de soluções da regressão, algumas utilizam a regularização (Ridge, Bayesian Ridge, ElasticNet) é um método para solucionar problemas nos quais ocorrem

Os dados foram normalizados utilizando a estratégia *MinMaxScaler*, que normaliza cada atributo no intervalo [0-1].

A. Redes Neurais

O primeiro modelo preditor consiste numa rede neural artificial e diferentes configurações para avaliar o desempenho delas. Em resumo, o seu funcionamento acaba sendo bem simples, o modelo recebe como entrada os dados de uma determinada rodada e faz uma predição das pontuações dos jogadores para a próxima rodada.

Para estimar a melhor arquitetura para rede, bem como os hiperparâmetros, foi utilizada a estratégia *GridSearch*. Nesse método, todas as combinações possíveis entre os parâmetros são testados usando uma validação cruzada com 5 *folds*. A combinação de parâmetros que for melhor na média dos *folds*, é considerada a melhor. As seguintes combinações de redes neurais foram utilizadas com a numeração de cada item corresponde ao seu `id`:

1) quatros camadas escondidas cada uma com 50 neurônios.
2) três camadas escondidas sendo os número de neurônios de cada uma delas é, respectivamente, 50, 100, 50.
3) quatro camadas escondidas sendo os número de neurônios de cada uma delas é, respectivamente, 50, 100, 100, 50.
4) apenas uma camada escondida com 128 neurônios.
5) duas camadas escondidas cada uma com 128 neurônios.
6) três camadas escondidas cada uma com 128 neurônios.

Uma vez que trata-se de um problema de regressão, a função de ativação da saída escolhida para a rede foi a função linear e a rede será treinada visando minimizar a Root Mean Squared Error (RMSE).

A Tabela II mostra o resultado da estratégia comentada acima para diferentes configurações de redes neurais, variando a quantidade de camadas escondidas e seus neurônios. Os *scores* negativos obtidos indicam que os modelos avaliados tem um desempenho muito ruim, a documentação do `scikit-learn` informa que o melhor score seria 1.

Table II
EXPERIMENTOS DAS REDES NEURAIS

id	fold 1	fold 2	fold 3	fold 4	fold 5	média
1	-19.514	-22.250	-19.641	-19.224	-17.298	-18.879
2	-22.382	-19.712	-19.404	-19.025	-17.154	-18.601
3	-17.114	-19.669	-22.350	-19.490	-19.038	-18.585
4	-19.791	-19.503	-22.334	-19.013	-17.163	-18.562
5	-22.282	-19.824	-20.116	-17.211	-19.424	-18.592
6	-19.796	-17.251	-19.702	-19.019	-22.412	-18.843

A melhor arquitetura foi a que utiliza apenas uma camada escondida e 128 neurônios, sendo a média do seu *score* igual a -18.562. O resultado mostra que novas arquiteturas precisam ser exploradas para conseguir melhorar este resultado, uma vez que as redes neurais testadas tem comportamento muito similar entre elas. O *score* para o conjunto de validação foi de -18.962 pelos resultados analisados anteriormente esse baixo desempenho já era esperado.

B. Regressão Linear: Random Forest

A regressão linear Random Forest [13] usa o método ensemble para combinar o resultado de diversos estimadores base para aumentar robustez e generalização quando comparado com apenas um estimador. O Random Forest usa um meta-estimador, que consiste numa árvore de decisão, e usa média para incrementar o resultado da predição assim como reduzir a possibilidade overfitting.

No treinamento do modelo, usamos a classe `RandomForestRegressor` do `sklearn` com profundidade de no máximo 500, além disso usamos `GridSearchCV` para explorar o uso de uma quantidade diferente de estimadores, variando o valor entre 10, 100 e 1000, e obtivemos o melhor resultado com esse parâmetro com valor igual a 1000.

C. Regressão Linear: Ridge

A regressão linear de Ridge [14], chamada também regularização de Tikhonov, com o termo de regularização $\alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2$ a função de custo, mantendo o peso o menor possível. O hiperparâmetro α controla quanto o termo deve influenciar. Se for zero, reduzimos o problema a uma regressão linear e, caso seja um valor muito alto, os pesos tendem a ir para zero. Abaixo, temos a equação da função de custo:

$$J(\theta) = MSE(\theta) + \alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2 \tag{1}$$

A execução do modelo é feita de maneira bem simples com o `sklearn` e a classe `RidgeCV` que realiza o uso de cross-validação para realizar o treinamento.

D. Regressão Linear: Bayesian Ridge

A regressão de Bayesian Ridge [15] utiliza a distribuição de propabilidade normal para calcular a reposta sendo caracterizado pela média e variância dos dados, ou seja, seu valor final será proporcional a essa distribuição conforme a equação:

$$y \sim \mathcal{N}(\theta^T X, \sigma^2 I) \tag{2}$$

O objetivo dessa regressão não é determinar o melhor valor dos parâmetros do modelo, mas sim determinar um distribuição posterior para esses parâmetros. Além da resposta do modelo ser proviniente da distribuição de probabilidade, os parâmetros também são oriundos dessa distribuição.

Table III
MELHOR AJUSTE PARA O MODELO BAYESIAN RIDGE

Parâmetro	Valor
α_1	1^{-7}
α_2	1^{-5}
λ_1	1^{-5}
λ_2	1^{-7}

O modelo possui 4 parâmetros para fazer o ajuste da regressão, eles são: α_1, α_2, λ_1 e λ_2. Com o uso da função `GridSearchCV`, fazemos uma combinação de todos os valores definidos para cada parâmetro, que são 1^{-7}, 1^{-6} e 1^{-5}.

Ao final da execução e explorar todas as possibilidades, chega-se ao melhor ajuste, melhor *score*, conforme a Tabela III.

E. Regressão Linear: Elastic Net

A regressão linear Elastic Net [16] é uma combinação de duas regressões, Ridge e Lasso, com uma variável, *r*, para controlar a relação entre as duas. Casos *r* seja igual a um, a função de custo se reduz a de Ridge, mostrada anteriormente, e se for igual a zero, o comportamento será igual a regressão linear de Lasso.

$$J(\theta) = MSE(\theta) + r\alpha\frac{1}{2}\sum_{i=1}^{n}|\theta_i| + \frac{1-r}{2}\alpha\sum_{i=1}^{n}\theta_i^2 \quad (3)$$

A equação acima mostra a nova função de custo, na qual interage as duas regressões.

F. Regressão Linear: Gradient Boosting

G. Regressão Linear: SVR (Support Vector Regression)

VI. Resultados

Os modelos descritos na seção anterior foram todos validados com o conjunto de treinamento, após tratamento dos dados, e compreende os anos de 2014 até 2016. Os dados do campeonato de 2017 foi tratado como o conjunto de validação, ou seja, utiliza-se para comparar a performance do modelo e definir qual será escolhido para a fase final dos experimentos.

A. Conjunto de Validação

O conjunto de validação contempla a temporada 2017 na qual o vencedor totalizou 2460.19 pontos com uma média de 67.4 pontos. A Tabela IV contém as métricas avalidas em cada modelo com os melhores parâmetros explorados. O modelo com melhor desempenho e menor variação foi a regressão linear de Bayesian Ridge, com uma média de 55.79 pontos, levando em consideração apenas da rodada 6 até 38. Outro fator que pode ter impactado no resultado abaixo do campeão foi a não inclusão do técnico e capitão.

Table IV
Experimentos de Validação

Modelo	Score	Média	σ	Total de Pontos
Redes Neurais	-19.25	49.37	15.25	1629.20
Random Forest	-0.03	35.75	16.72	1179.90
Bayesian Ridge	0.01	55.79	14.59	1841.29
Ridge	NaN	52.59	18.27	1735.69
ElasticNet	NaN	55.35	15.71	1826.70
Gradient Boosting	-0.01	39.15	19.14	1291.80
SVR	-0.17	30.88	17.75	1019.00

A Figura 1 complementa os resultados da tabela, mostrando os pontos obtidos por cada modelo para cada rodada comparando com o campeão da temporada. O campeão, com a cor azul, acaba ganhado praticamente todas as rodadas quando comparado com os outros modelos, ocorrendo apenas alguns casos esporádicos de resultado ruim como na rodada 10 e 30. O modelo do Bayesian Ridge mantém um equilíbrio ao longo das rodadas sempre próximo dos outros modelos e quando o resultado obtido a perda é moderada.

B. Conjunto de Teste

O conjunto de teste compreende o Campeonato Brasileiro de 2018, finalizado no dia 02 de dezembro, no qual houve a participação de 20 times e um total de 38 rodadas. O campeão dessa temporada do Cartola FC foi Mateus Gomes Soares com o time Flajuveteus FC, acumulando o total de 3288.15 pontos com uma média de 86.53 pontos por rodada.

TODO(ciroceissler): nossa media pontuação.

TODO(ciroceissler): comparar rodada a rodada.

VII. Conclusão

Neste projeto, criamos um modelo para prever a melhor escalação para cada rodada do *fantasy game* Cartola FC. Diversas técnicas para modelar o problema foram analisadas, testando a eficiência de cada modelo. O teste utilizou a temporada de 2014 a 2016 como conjunto de treinamento e a temporada de 2017 como validação. Ao final, optamos por utilizar o Bayesian Ridge por causa do melhor desempenho comparado aos demais modelos e testamos com a última temporada 2018, que finalizou no dia 02 de novembro.

TODO(ciroceissler): o resultado final foi.

TODO(ciroceissler): analise superficial dos dados.

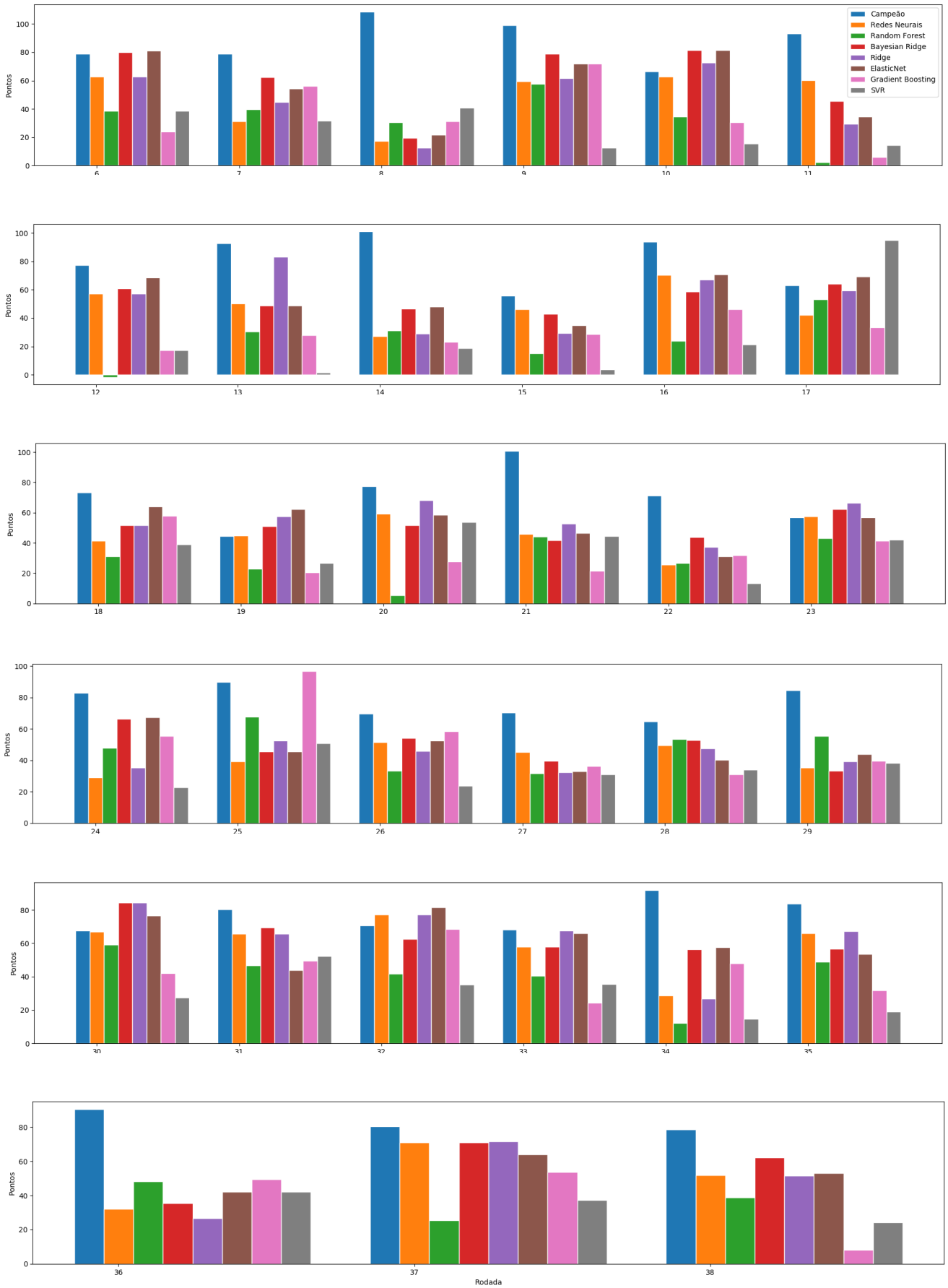

Figure 1. Experimentos Temporada 2017 - Pontuação por Rodada para cada Modelo

REFERENCES

[1] FIFA, "Fifa/coca-cola world ranking." 2018.

[2] W. Martin, "Big banks like goldman sachs spectacularly failed to predict the world cup winner — here's why." *Business Insider*, 2018.

[3] G. Sugar and T. Swenson, "Predicting optimal game day fantasy football teams," 2015.

[4] J. W. Porter, "Predictive analytics for fantasy football: Predicting player performance across the nfl," 2018.

[5] R. Lutz, "Fantasy football prediction," *arXiv preprint arXiv:1505.06918*, 2015.

[6] P. Dolan, H. Karaouni, and A. Powell, "Machine learning for daily fantasy football quarterback selection," *CS229 final report*, 2015.

[7] A. Becker and X. A. Sun, "An analytical approach for fantasy football draft and lineup management," *Journal of Quantitative Analysis in Sports*, vol. 12, no. 1, pp. 17–30, 2016.

[8] G. VISCONDI, D. JUSTO, and N. GARCÍA, "Aplicação de aprendizado de máquina para otimização da escalação de time no jogo cartola fc."

[9] H. L. Schmidt, "Uso de técnicas de aprendizado de máquina no auxílio em previsão de resultados de partidas de futebol." 2017.

[10] E. Mota, D. Coimbra, and M. Peixoto, "Cartola fc data analysis: A simulation, analysis, and visualization tool based on cartola fc fantasy game," in *Proceedings of the XIV Brazilian Symposium on Information Systems*. ACM, 2018, p. 18.

[11] H. Gomide, "Como montamos defesas no cartolafc? com estatística e modelagem de dados." acessado em 05-12-1986. [Online]. Available: https://bit.ly/2E3zMZ8

[12] ——, "Github - repositório cartola." acessado em 05-12-2018. [Online]. Available: https://github.com/henriquepgomide/caRtola/

[13] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[14] A. Tikhonov, *Numerical methods for the solution of ill-posed problems*.

[15] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[16] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[17] H. Lopes, "Aprendizado de mÁquina aplicado a previsÃo de desempenho de jogadores de futebol." 2018.

[18] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2017.

[19] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics) springer-verlag new york," *Inc. Secaucus, NJ, USA*, 2006.

[20] N. Developers, "Numpy," *NumPy Numpy. Scipy Developers*, 2013.

[21] "scikit-learn," acessado em 08-10-2018. [Online]. Available: http://scikit-learn.org/stable/

[22] "Com 2460.19 pontos 'jorgito10 (o mito)' é o vencedor da liga ge ap em 2017." *GloboEsporte.com*, 2017, acessado em 05-12-2018. [Online]. Available: https://globoesporte.globo.com/ap/cartola-fc/noticia/com-246019-pontos-jorgito10-o-mito-e-o-vencedor-da-liga-ge-ap-em-2017.ghtml

[23] "Cartola fc," *GloboEsporte.com*, acessado em 05-12-2018. [Online]. Available: https://cartolafc.globo.com/

[24] J. Hadamard, "Sur les problèmes aux dérivées partielles et leur signification physique," *Princeton university bulletin*, pp. 49–52, 1902.