

Predição do Resultado de Partidas de Xadrez em Lichess.org

Rafael Schubert Campos

Universidade Regional de Blumenau — FURB

Aprendizado de Máquina I: Aprendizado Supervisionado — Especialização em *Data Science*, Turma II

1. Introdução

O Lichess.org é o site dum projeto *open-source* que provê um servidor aberto e gratuito para a realização de partidas de xadrez online¹. O site permite desde a realização de partidas isoladas — no formato tradicional e algumas variantes² — até competições, além de fornecer análises sobre os jogos realizados e permitir que as partidas sejam assistidas em tempo real¹.

Por sua vez, o xadrez é um antigo jogo de tabuleiro abstrato muito conhecido, onde dois jogadores jogam competitivamente pelo objetivo de capturar o *rei* do adversário. O xadrez é jogado sobre um tabuleiro quadriculado de oito linhas por oito colunas, onde os jogadores movem suas peças em turnos alternados entre si. Os jogadores dispõem, cada um, dum conjunto idêntico de peças. Cada tipo de peça possui um padrão único de movimentação, que deve ser considerado na estratégia do jogador.

No Lichess.org, assim como em competições presenciais oficiais, os jogos são disputados dentro dum limite de tempo — um cronômetro regressivo — que cada jogador tem para ponderar e realizar seus movimentos.

O jogo chega ao fim sob qualquer uma das seguintes condições:

- se um jogador é incapaz de realizar um movimento sem que seu rei fosse ser capturado no turno adversário seguinte (o chamado *xequê-mate*);
- se um dos jogadores desiste e concede o jogo ao adversário;
- se o cronômetro dum dos jogadores se esgota.

E os resultados possíveis para uma partida de xadrez no Lichess.org são:

- o jogador das peças brancas vence;
- o jogador das peças pretas vence;
- o jogo empata (sob circunstâncias muito específicas).

Quanto a este projeto, o objetivo foi desenvolver um modelo de classificação, usando técnicas e algoritmos de aprendizado supervisionado, para prever o resultado duma partida de xadrez qualquer no Lichess.org. Estabeleceu-se como meta um mínimo de 80% de precisão para o modelo — isto é, que ele predissesse o resultado correto em pelo menos 80% das vezes.

O conjunto de dados usado neste projeto foi obtido no site Kaggle, onde fora submetido por um usuário há cerca de três anos, intitulado como *Chess Game Dataset (Lichess)*. Trata-se dum conjunto de dados contendo cerca de 20 mil partidas de xadrez na modalidade tradicional, disputados no Lichess.org, com tempos de duração de jogo variados.

2. Metodologia

Por tratar-se dum problema de classificação multi-classe, o modelo de classificação foi composto por *features* que ajudassem a destacar cada classe das demais. Focou-se a exploração principalmente em encontrar variáveis do conjunto de dados que exibissem distribuições da população de cada classe que pudessem ser distinguidas. Além disso, também buscou-se encontrar variáveis independentes categóricas que apresentassem algum nível de influência sobre a variável dependente.

Para chegar ao melhor modelo de classificação possível, decidiu-se testar e avaliar o desempenho de alguns algoritmos de classificação por aprendizado supervisionado diferentes. O algoritmo que melhor se desempenhou foi usado para produzir o modelo de classificação final. Os algoritmos propostos como candidatos foram:

- Árvore de Decisão;
- Floresta Aleatória;
- e Regressão Logística Multi-Classe.

Para realizar os testes e a avaliação, o conjunto de dados foi dividido por *hold-out* em dois subconjuntos: um conjunto de treino, com 80% dos registros, e um conjunto de testes, com os 20% restantes. O conjunto de treino foi usado para selecionar o melhor algoritmo e também treinar um modelo de classificação a partir do algoritmo selecionado. O conjunto de teste foi usado para avaliar o desempenho do modelo de classificação final produzido.

No processo de seleção do melhor algoritmo, os algoritmos candidatos foram submetidos a *cross-validation* por *k-fold* estratificado (com 5 *folds*), usando *grid search* com *cross-validation*, também por *k-fold* estratificado, para otimizar os parâmetros de cada algoritmo. A métrica de comparação adotada foi a média da precisão ponderada (*weighted precision*) dos *folds*.

O algoritmo com o valor mais alto na métrica de comparação foi escolhido para produzir o modelo de classificação. Com o algoritmo selecionado, treinou-se um modelo dele com o conjunto de treino completo, usando os melhores parâmetros encontrados com *grid search cross-validation* por *k-fold* estratificado (com 5 *folds*).

Em seguida, o modelo foi aplicado sobre o conjunto de testes para avaliar seu desempenho com base na precisão ponderada dos resultados. Os resultados são reportados neste pôster.

3. Experimentos

Avaliou-se a integridade do conjunto de dados e removeu-se observações repetidas ou inválidas segundo os conceitos das variáveis, além de *outliers* significativos. Com isso feito, explorou-se o conjunto de dados em busca de variáveis candidatas a produzir *features* para o modelo de classificação.

Depois de extensa análise, formulou-se o seguinte conjunto de *features* para o modelo, extraídos para cada partida de xadrez:

- número de turnos decorridos na partida;
- rating* inicial de cada jogador da partida;
- número de cheques realizados por cada jogador da partida;
- número de capturas realizadas por cada jogador da partida;
- diferença de *rating* favorável a cada jogador da partida;
- diferença do número de capturas favorável a cada jogador da partida;
- indicador se o número de capturas realizadas por cada jogador eram iguais.

Após a limpeza e a preparação, o conjunto de dados completo possuía os dados de 17.793 partidas diferentes de xadrez. Dessas observações, 8.912, 8.112 e 769 pertencem às classes

`white` (vitória do jogador das peças brancas), `black` (vitória do jogador das peças pretas) e `draw` (jogo empatado), respectivamente.

O conjunto de dados foi dividido entre conjunto de treino e conjunto de teste, com 14.234 (80%) e 3.559 (20%) observações, respectivamente. As observações do conjunto de treino apresentam a seguinte composição: 7.129, 6.490 e 615 observações das classes `white`, `black` e `draw`, respectivamente. Já o conjunto de teste apresenta a seguinte composição: 1.783, 1.622 e 154 observações das classes `white`, `black` e `draw`, respectivamente.

As *features* do modelo foram extraídas para as observações do conjunto de treino, e os dados resultantes foram usados para comparar o desempenho dos algoritmos. Modelos de cada algoritmo candidato foram submetidos a *cross-validation*, e a precisão ponderada média foi mensurada separadamente para os dados de treino e de teste dos *folds*.

Os resultados de cada algoritmo candidato são reportados na Tabela 1, onde o melhor desempenho foi destacado.

Tabela 1: Precisão ponderada de cada algoritmo candidato

Métrica	Algoritmo		
	Árvore de Decisão	Floresta Aleatória	Regressão Logística Multi-Classe
Treino	0,802	0,808	0,789
Teste	0,796	0,807	0,783

O algoritmo que melhor se desempenhou foi o classificador de Floresta Aleatória, com uma precisão ponderada média de aproximadamente 80,7%. Ademais, esse resultado foi consistente com seu desempenho de treino, de aproximadamente 80,8%. Portanto, o algoritmo de Floresta Aleatória foi selecionado para produzir o modelo de classificação final.

Então, um modelo do algoritmo foi treinado usando o conjunto de treino completo, empregando *grid search* para determinar os melhores parâmetros. O conjunto de teste foi submetido ao modelo, e o desempenho de suas predições foi avaliado. Os resultados são reportados na Tabela 2, trazendo os melhores parâmetros encontrados e a precisão ponderada do modelo de classificação.

Tabela 2: Desempenho do modelo de classificação final

Configuração	Precisão Ponderada
Algoritmo: Floresta Aleatória # de estimadores: 100 Critério: gini Profundidade Máxima: 4	0,801

O modelo obteve uma precisão ponderada de aproximadamente 80,1%. Pode-se observar o desempenho bruto do modelo de classificação produzido na Tabela 3. Nela, apresenta-se a matriz de confusão das predições para o conjunto de teste em relação a suas classes verdadeiras.

Tabela 3: Matriz de confusão do modelo de classificação final

	Predito			
	<code>black</code>	<code>draw</code>	<code>white</code>	Total
<code>black</code>	1.308	0	314	1.622
<code>draw</code>	66	16	72	154
<code>white</code>	286	0	1.497	1.783
Total	1.660	16	1.883	3.559

Na matriz de confusão, nota-se que 1.308 das 1.660 predições da classe `black` foram corretas. O mesmo aplica-se a todas as predições da classe `draw`, e para 1.497 das 1.883 predições da classe `white`. Isso significa precisões de 78,8%, 100,0% e 79,5% para as classes `black`, `draw` e `white`, respectivamente. Por outro lado, suas coberturas foram de 80,6% (1.308 de 1.622 observações), 10,4% (16 de 154) e 84,0% (1.497 de 1.783), respectivamente.

4. Conclusão

O experimento produziu um modelo de classificação capaz de prever o resultado duma partida de xadrez qualquer no Lichess.org com aproximadamente 80% de precisão. Dentre três algoritmos de classificação de aprendizado supervisionada — Árvore de Decisão, Floresta Aleatória e Regressão Logística Multi-Classe —, o algoritmo de Floresta Aleatória apresentou o melhor desempenho. Em comparação aos outros algoritmos, a Floresta Aleatória parece capturar melhor a variabilidade do xadrez com seus múltiplos estimadores.

O modelo se desempenhou bem, considerando as expectativas iniciais, em prever os resultados de partidas onde um dos jogadores vencera. Percebe-se precisões de 78,8% e 79,5% para as classes `black` e `white`, respectivamente, contra coberturas de 80,6% e 84,0%. No entanto nota-se que o modelo teve dificuldade em reconhecer partidas que resultaram em empate. Embora tivesse uma precisão de 100,0% para a classe `draw`, ele apresentou uma cobertura de apenas 10,4% para a mesma classe.

Acredita-se que esse modelo poderia ser melhorado com um estudo mais aprofundado sobre as características e padrões das partidas de xadrez que resultaram em empates. Sugestões incluem: considerar o tempo decorrido da partida como um critério; obter e avaliar o tempo decorrido nos cronômetros de cada jogador; e considerar a composição do conjunto de peças disponível a cada jogador.

5. Referências

- Lichess.org* — *About lichess.org*. Site: <https://lichess.org/about>, visitado em: 12/10/2020;
- Lichess.org* — *Frequently Asked Questions*. Site: <https://lichess.org/faq#variants>, visitado em: 12/10/2020;