

Trabalho A2 - Aprendizado de Máquinas

Prevendo duração e resultado de partidas do jogo online DotA 2

Tomás Ferranti

*Escola de Matemática Aplicada,
Fundação Getúlio Vargas*

23 de junho de 2021

Sumário

1	Introdução	1
2	Obtenção do dataset	2
3	Pré-processamento dos dados	2
4	Análise Exploratória	3
5	Modelagem	5
5.1	Duração da partida	5
5.2	Resultado da partida	6
6	Resultados dos Modelos	6
7	Conclusão	7
8	Reprodução do Conteúdo	7

1 Introdução

A evolução de métodos sofisticados desenvolvidos para previsão de eventos vem se tornando cada vez mais rápida e eficaz. Desde direção automática em veículos de transporte à valores futuros em mercados financeiros, estas técnicas se encontram presente em diversas ocasiões.

Um desses setores de interesse é a previsão de acontecimentos e resultados em partidas de jogos eletrônicos. O jogo em questão neste trabalho se chama DotA 2, e conta com o maior prêmio de campeonato já disputado na história dos E-Sports (termo usado para diversos torneios que envolvem jogos eletrônicos), chegando a cerca de \$40 milhões de dólares.

Este jogo se encaixa na categoria MOBA (Multiplayer Online Battle Arena), isto é, consiste de vários jogadores online desafiando uns aos outros em diferentes partidas. Ao procurar uma partida, você e mais quatro pessoas são conectadas contra outras cinco. No início, cada jogador pode escolher um herói de 119 heróis sem repetição e este fica com ele o resto da partida. Cada partida dura cerca de 40 minutos e não há empate.

Como cada herói é único e possui um papel diferente na partida, certas combinações e execuções da escolha destes cinco heróis permitem que o seu time obtenha uma vantagem contra o inimigo. Isto torna cada partida única e extremamente interessante, fazendo com que haja espaço para Aprendizado de Máquinas.

A razão de escolha deste tema é devido a este jogo ser extremamente popular e estar crescendo bastante nessa área. A empresa OpenAI com vice-presidente o empreendedor e filantropo sul-africano-canadense-americano Elon Musk publicou em 2018 uma inteligência artificial capaz de controlar o time de cinco heróis sozinho, derrotando até o time ganhador do mundial na época [3].

O objetivo deste trabalho é, a partir das escolhas iniciais dos heróis tentar prever a duração e resultado da partida. O trabalho consiste em cinco partes: coleta dos dados, processamento, análise exploratória, modelagem e análise dos resultados.

2 Obtenção do dataset

Os dados utilizados para este trabalho se encontram na plataforma Kaggle [1]. Consiste de 50000 partidas com diversos detalhes sobre os acontecimentos e atributos. Estas são partidas públicas, ou seja, não são de competições ou torneios, mas de jogadores casuais.

Estes estão particionados em diversos arquivos do tipo CSV contendo informações sobre diferentes aspectos das partidas. Como o que nos interessa são as escolhas dos heróis e o resultado da partida, após uma breve análise sobre os dados foi fácil determinar que os únicos arquivos necessários seriam "hero_names.csv", "match.csv" e "players.csv".

Sobre cada arquivo pode-se obter:

1. **(hero_names.csv):** este arquivo permite identificar o ID do herói e seu respectivo nome, podendo assim converter os dados de outras tabelas aos respectivos heróis;
2. **(match.csv):** já este nos fornece, entre algumas de suas informações, o ID da partida, sua duração e resultado. Portanto apenas nos falta os dados dos heróis de cada partida;
3. **(players.csv):** para finalizar, este armazena o ID da partida, o ID do herói e em que time ele estava, sendo assim possível montar os dados do problema. Para cada ID de partida evidentemente têm-se 10 registros: um para cada jogador.

Sendo assim, após essa análise pode-se montar os dados através do processamento em Python.

3 Pré-processamento dos dados

No total há 50000 partidas e 112 heróis únicos, pois no momento de coleta dos dados ainda não havia adicionados os 7 restantes para o total de 119. O objetivo é montar três variáveis principais, definidas por:

1. X consiste de uma matriz binária em que em cada linha têm um time de uma partida e nas colunas se encontram os heróis escolhidos por aquele time indicados por 1 e os não escolhidos por 0. Sendo assim para cada partida obtêm-se mais dois dados para os modelos;
2. y_{dur} possui também possui mesma dimensão em linhas mas possui apenas uma coluna onde seus valores são a duração da partida em segundos correspondente à respectiva linha (partida) em X ;
3. y_{win} semelhante a y_{dur} , só que seus valores são 0 para caso o time perdeu e 1 caso contrário.

e uma variável secundária *indid_heroname*, que seria um dicionário de modo a indicar o nome do herói correspondente a cada índice da coluna de X . O processamento principal foi realizado nos dados de "players.csv", iterando sobre cada linha e identificando o ID de partida, o ID do herói e o time, levando cerca de 40 segundos.

Após as execuções é removido de X as colunas em que a soma é zero (alguns heróis possuíam ID mas não tinham partidas registradas). Também é retirado as linhas de X e seus índices correspondentes em y onde a soma é diferente de cinco (algum dado de herói do time faltante).

Com isso as variáveis principais ficaram com tamanhos: X com (99964, 110) e os y 's com (99964, 1).

4 Análise Exploratória

Antes de qualquer modelagem é fundamental realizar uma análise visual e estatística dos dados de modo a identificar padrões e possíveis relações.

Na Figura. 1 encontra-se a distribuição de duração das partidas, mostrando um comportamento esperado e sua concentração entre 20 e 60 minutos. O gráfico se assemelha à uma distribuição normal.

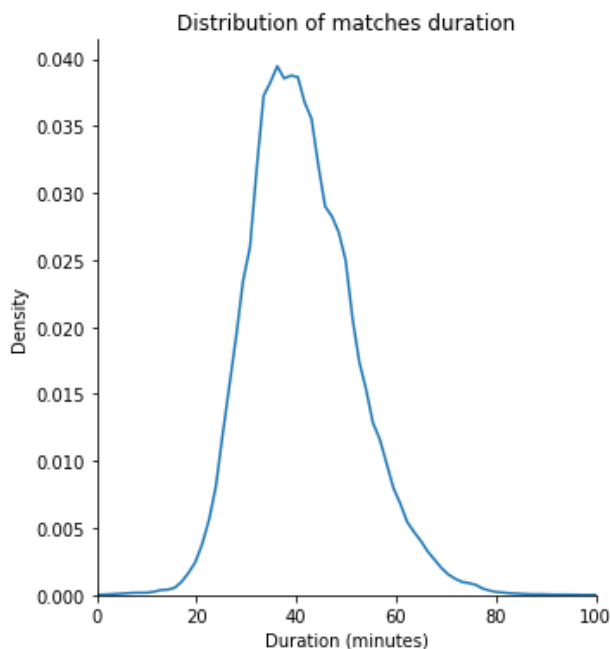


Figura 1: Distribuição da duração das partidas, restringida no intervalo $[0, 100]$ minutos.

Indo um pouco mais fundo na duração das partidas, podemos filtrar esta por heróis. Com isso, na Figura. 2 há o *boxplot* da distribuição dos heróis mais significantes.

Fazendo uso do conhecimento sobre o jogo, os heróis Techies, Tinker e Sniper possuem as maiores médias de duração de partida devido à serem excepcionais em defender a base mesmo com desvantagem, assim prolongando o jogo. Já Medusa e Storm Spirit possuem os maiores desvios padrão pois alcançam seus picos de combate no começo do jogo, terminando-o rapidamente ou atrasando-o bastante. Techies também se encontra neste pois dos heróis que prolongam partida ele é o menos inconsistente, fazendo com que seu desvio padrão seja imenso ¹.

Além dos dados de duração, também temos a taxa com que cada herói é escolhido. Alguns heróis são mais populares que os outros e isto se é bastante interessante de analisar. Esta taxa, chamada de *Pick Rate*, consiste no total de vezes que um herói foi escolhido dividido pelo número total de partidas. Ela se encontra na Figura. 3.

¹O herói Techies consiste em plantar bombas no chão, de modo a pegar pessoas desprevinidas. Um bom time sempre carregaria detecção destas, tornando-o inútil. Entretanto, como são partidas públicas comuns, isto não acontece.

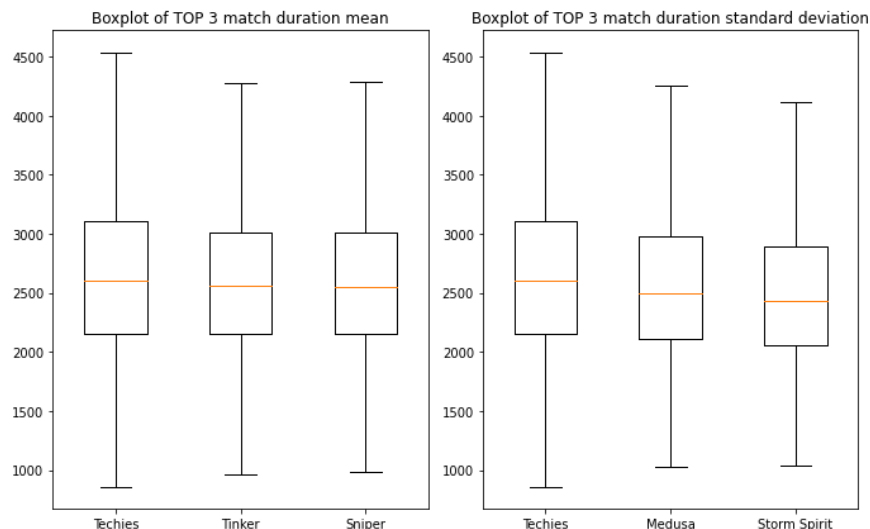


Figura 2: Distribuição da duração das partidas restringidas à cada herói. À esquerda têm-se os três heróis com maiores médias de duração e à direita os três heróis com maiores desvio padrão.

Pick rate (TOP 5) (%)	Hero Name	Pick rate (BOTTOM 5) (%)	Hero Name
20.88%	Windranger	0.58%	Chen
17.0%	Shadow Fiend	0.84%	Elder Titan
11.68%	Invoker	0.93%	Brewmaster
11.33%	Earthshaker	0.97%	Lone Druid
11.18%	Slardar	0.99%	lycan

Figura 3: Nome dos cinco primeiros heróis e cinco últimos na categoria *Pick Rate*.

Os dois heróis (Windranger e Shadow Fiend) são simples e divertidos de se jogar, sendo adorados pela comunidade e portanto os dois mais escolhidos. Já os cinco heróis que se encontram com a menor *Pick Rate* são complicados de se jogar, necessitando bastante atenção e *micromanagement* (controlar mais de uma unidade), fazendo com que sejam menos escolhidos.

Analisando agora os dados de vitórias, temos a taxa de vitória de cada herói. O jogo é constantemente atualizado e em cada *patch* (nome dado ao respectivo *update*) temos heróis que se destacam entre os demais. Geralmente estes são quase sempre escolhidos quando o time busca a vitória. Esta taxa, chamada de *Win Rate*, consiste no total de vezes que um herói ganhou dividido pelo número total de partidas. Ela se encontra na Figura. 4.

Win rate (TOP 5) (%)	Hero Name	Win rate (BOTTOM 5) (%)	Hero Name
59.07%	Omniknight	40.58%	Nature's Prophet
58.35%	Spectre	40.79%	Shadow Demon
57.98%	Wraith King	40.85%	Enchantress
56.88%	Abaddon	41.27%	Broodmother
56.87%	Undying	41.58%	Storm Spirit

Figura 4: Nome dos cinco primeiros heróis e cinco últimos na categoria *Win Rate*.

Dentre os cinco maiores, Omniknight é um herói de suporte relativamente simples de jogar mas com

altíssimo impacto, tornando aliados praticamente invulneráveis. Desde da data em que os dados foram coletados ele sofreu bastante alterações e agora se encontra em um valor menor de *Win Rate*. Abaddon e Undying são semelhantes ao Omniknight, fornecendo bastante ajuda à aliados. Spectre e Wraith King são heróis de *late game*, isto é, há mais chance de ganhar quanto mais a partida demora a acabar, e partidas públicas geralmente não possuem coordenação para encerrar o jogo antes que isso aconteça.

Nature's Prophet e Broodmother possuem dificuldades similares: ambos são heróis de alto risco, com kits que são extremamente versáteis mas que também permitem grande capacidade para erros, estes que geralmente não são recuperáveis, causando a perda da partida.

Com isso, após essa inspeção das variáveis pode-se partir para os modelos.

5 Modelagem

Nesta seção serão apresentados diversos modelos e como seus parâmetros foram calibrados, de modo que a análise e performance são discutidas em Resultados dos Modelos. Os dados foram divididos em 70/30 treino/teste. Possíveis análises de redução de dimensão e de correlação foram descartadas devido à não fazer muito sentido com relação ao formato e significado dos dados.

Tenha em mente que a dimensão dos dados é grande o suficiente para demorar bastante para realizar o *fit* dos dados junto com hiperparametrização (alguns modelos alcançaram horas de calibração).

Os hiperparâmetros foram analisados através de um *cross-fold* de 5 partições no treino, considerando os com média de performance mais alta pelas divisões.

5.1 Duração da partida

Os seguintes modelos foram adequados aos dados de duração da partida: Linear Regression, Ridge Regression, Decision Tree Regression e Random Forest Regression. Abaixo segue mais detalhes sobre a escolha e calibração de cada um deles:

- **Linear Regression:** Como um modelo simples e eficiente, Linear Regression é essencial em qualquer trabalho de predição, possuindo nenhum hiperparâmetro a ser calibrado e balanceando eficiência \times simplicidade;
- **Ridge Regression:** Dado o resultado da regressão linear, os coeficientes foram checados e se mostraram de dimensões inapropriadas. Logo com Ridge Regression pode-se penalizar o modelo por isso, obtendo coeficientes menores e mais confiáveis. O hiperparâmetro de regularização foi calibrado através dos valores [1e-3, 1e-2, 1e-1, 1];
- **Decision Tree Regression:** Simples de analisar quando a profundidade é baixa, Decision Trees são importantes para avaliar a possível realização de métodos de *Ensemble* mais sofisticados. Seus hiperparâmetros foram escolhidos através de: Profundidade Máxima = [1, 5, 10, 20, 50, 100], Mínimo de Amostras no Split = [50, 100, 200, 400] e Mínimo de Amostras em uma Folha = [10, 20, 40, 80];
- **Random Forest Regression:** Aumentando a complexidade temos as Random Forests que agregam Decision Trees de forma inteligente, tentando diversificar o formato das árvores para um resultado melhor. Seus hiperparâmetros foram escolhidos através de: Número de Árvores = [1, 10, 50, 100], Profundidade Máxima = [1, 5, 10, 20, 50, 100, 150, 200] e Mínimo de Amostras no Split = [50, 100, 200, 400], fixando Mínimo de Amostras em uma Folha com 40.

Dado que a duração da partida é uma variável contínua, treinar os modelos leva bastante tempo, chegando a horas de execução. Foi avaliado então que já eram suficientes para o trabalho.

5.2 Resultado da partida

Para o resultado da partida foi realizado a escolha dos modelos: Naive, Linear Classifier, Logistic Classifier, Decision Tree Classifier, Bagged Tree Classifier, Random Forest Classifier e Gradient Boosting Classifier. Logo a seguir segue uma descrição sobre o uso de cada um:

- **Naive Classifier:** Usado geralmente para avaliar o desempenho de outros modelos, o Naive Classifier consiste em atribuir a resposta sempre para a classe com maior quantidade;
- **Linear Classifier:** Semelhante a Linear Regression, só que ao invés de retornar um resultado contínuo, temos um ponto de corte 0.5 onde valores abaixo se tornam 0 (derrota) e valores acima 1 (vitória).
- **Logistic Classifier:** Sendo um modelo linear generalizado que usa como função de ligação a função logit, o Logistic Classifier se mostra como um ótimo candidato como modelo. Seu coeficiente de regularização C é escolhido dentre o melhor de [0.001, 0.01, 0.1, 1, 10, 100, 1000].
- **Decision Tree Classifier:** Semelhante à descrição da seção anterior;
- **Bagged Tree Classifier:** Também agregadores de Decision Trees, as Bagged Trees sacrificam um pouco do viés para diminuir a variância das estimações, fazendo com que fiquem mais consistentes. O único hiperparâmetro calibrado é o Número de Árvores, onde os hiperparâmetros de cada árvore são Profundidade Máxima = 20, Mínimo de Amostras no Split = 200 e Mínimo de Amostras em uma Folha = 40;
- **Random Forest Classifier:** Semelhante à descrição da seção anterior;
- **Gradient Boosting Classifier:** Como mais um método de ensemble, com seu modelo em etapas o Gradient Boosting generaliza permitindo a otimização de uma função de perda diferenciável arbitrária. Seu hiperparâmetro calibrado é apenas o Número de Estimadores em [1, 100, 300, 500] com a Taxa de Aprendizado fixada em 0.1 e a Profundidade Máxima em 1.

Com o consumo de tempo suficiente na treinamento destes (que é menor que o de duração devido aos dados de predição serem binários), foi avaliado que haviam modelos suficientes para análise.

6 Resultados dos Modelos

Abaixo segue cada modelo e seu respectivo desempenho (R^2 na duração, Acurácia no resultado) no treino e no teste.

Model Name	Match Duration		Match Result	
	Train R2	Test R2	Train Accuracy	Test Accuracy
<i>Naive</i>	-	-	0.50106	0.49713
<i>Linear</i>	0.03224	0.03145	0.57431	0.57050
<i>Ridge</i>	0.03224	0.03145	-	-
<i>Logistic</i>	-	-	0.57290	0.57106
<i>Decision Tree</i>	0.01146	0.00853	0.55118	0.55122
<i>Bagged Tree</i>	-	-	0.56015	0.56045
<i>Random Forest</i>	0.02058	0.02134	0.56762	0.56649
<i>Gradient Boosting</i>	-	-	0.56518	0.56292

Cada um destes modelos possui um plot de resíduos e de matriz de confusão contidos no Notebook para verificação de alguma possível tendência.

Temos que para os modelos de duração da partida, Linear e Ridge obtiveram a mesma performance e a mais alta nos dados de teste. Entretanto, este valor de R^2 não é o suficiente para afirmar uma relação estrita entre duração de uma partida e escolha de heróis.

Esta falha em prever a duração da partida provavelmente acontece por causa da distribuição dado os campeões terem uma variância muito alta, tornando complicado realizar previsões pontuais.

Já para os modelos de resultado de partidas, todos exceto o Naive conseguiram uma acurácia acima de 55% nos dados de teste, mostrando que a escolha de heróis possui sim um efeito no resultado da partida. Podemos concluir isto pois os dados de teste estão divididos aproximadamente 50/50 em vitórias/derrotas (através do modelo Naive podemos perceber) fazendo com que qualquer performance acima aponte uma relação.

7 Conclusão

Apesar de nenhum modelo apresentar um resultado significativamente alto, pode-se com certeza afirmar uma relação entre a escolha dos heróis e a duração de uma partida por meio das análises dos dados e R^2 positivos nos dados de teste.

Através das acurácias em torno de 56% dos modelos de resultado também pode-se concluir uma influência da escolha dos heróis no resultado da partida, fazendo com que a fase inicial da partida seja de extrema importância para seu resultado final. Apesar de ser apenas 6% em incremento à média de 50%, tenha em mente que o jogo é muito mais complexo do que apenas a escolha dos heróis, tornando esses 6% extremamente importantes.

Uma das desvantagens deste trabalho foi a dimensionalidade dos dados, tornando a calibração dos parâmetros dos modelos que seriam eficientes extremamente custosa e, com isso, não conseguindo performances altas. Outro problema também são os dados possivelmente formarem apenas uma amostra aleatória da população total (autor não especificou o meio de coleta dos dados), sendo assim um pouco enviesados.

Dentre os ambientes de análise do DotA 2, o próprio jogo fornece algo chamado DotA Plus [4], onde você possui um conjunto de estatísticas lhe auxiliando durante suas partidas. Um possível trabalho futuro interessante seria tentar reproduzir ou superar as ferramentas disponíveis por essa plataforma.

8 Reprodução do Conteúdo

A linguagem de escolha para todo o trabalho foi Python através da plataforma Google Colab. Todos os códigos utilizados para as imagens e resultados mostrados neste documento podem ser reproduzidos através do Jupyter Notebook "*TOMAS_FERRANTI.ipynb*", que pode ser encontrado no GitHub do trabalho [2].

Para isso basta ter os três arquivos CSV baixados do Kaggle ("hero_names.csv", "match.csv" e "players.csv"), o notebook na mesma pasta e executar o código. Lembre-se de que algumas células dos modelos podem levar horas para terminar a execução.

Referências

- [1] Anzelmo, D. (2016). Dota 2 matches. Disponível em <https://www.kaggle.com/devinanzelmo/dota-2-matches>.
- [2] Ferranti, T. (2021). Código utilizado. Disponível em <https://github.com/TomasFerranti/dota2-matches>.
- [3] OpenAI (2018). Openai five. <https://blog.openai.com/openai-five/>.
- [4] Valve (2016). Dota plus. Disponível em <http://www.dota2.com/plus?l=portuguese>.