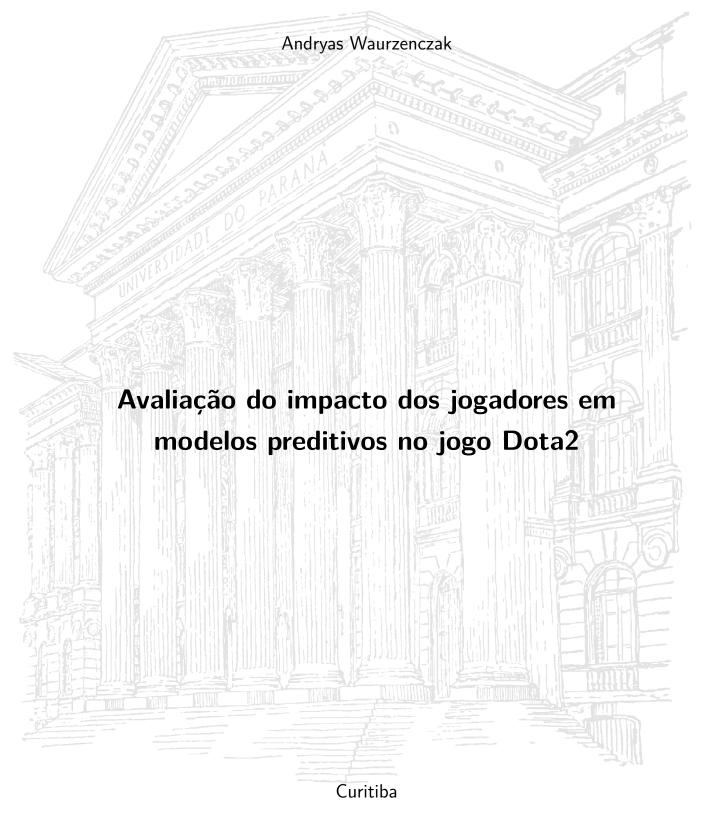
Universidade Federal do Paraná



2019

Andryas Waurzenczak

Avaliação do impacto dos jogadores em modelos preditivos no jogo Dota2

Projeto de Pesquisa apresentado à disciplina Laboratório A do Curso de Graduação em Estatística da Universidade Federal do Paraná, como requisito para elaboração do Trabalho de Conclusão de Curso

Universidade Federal do Paraná Setor de Ciências Exatas Departamento de Estatística

Orientador: Prof. Dr. Walmes Marques Zeviani

Curitiba

2019

Sumário

1	INTRODUÇÃO 3
2	OBJETIVOS
2.1	Objetivos Gerais
2.2	Objetivos Específicos
3	MATERIAIS E MÉTODOS
3.1	Materiais
3.1.1	Sobre os Dados
3.1.2	Recursos Computacionais
3.1.3	Coleta dos Dados
3.2	Métodos
4	CRONOGRAMA DE ATIVIDADES
	REFERÊNCIAS

1 Introdução

Dos astrágalos da antiga Grécia aos jogos de console e computador do século XXI. Ao longo da trágetoria humana os jogos tem desempenhado um papel essêncial para o desenvolvimento da ciência e da tecnologia. Atualmente os jogos tem servido como verdadeiros laborátorios para o desenvolvimento de áreas como Machine Learning e Inteligencia Artificial, isto por que tem-se, geralmente, uma grande quantidade de dados disponiveis e ambientes controlados. (SILVA; CHAIMOWICZ, 2017).

Um dos jogos que tem chamado a atenção de pesquisadores e, também, de grandes empresários é o jogo Dota2¹. Diversos trabalhos relacionados ao jogo foram feitos, como por exemplo, a instituição sem fins lucrativos OpenAI², financiada por um dos grandes magnatas e visionários do mundo, atualmente, Elon Musk³, que criou o primeiro programa capaz de "jogar o jogo" e que teve seu poder computacional demonstrado no The International de 2017 e 2018. Além de vários outros trabalhos que visavam, não somente a criação de algoritmos capazes de jogar o jogo mas também, prever o desfecho da partida no inicio do jogo.

¹ http://dota2.com/

² https://www.openai.com/

³ https://en.wikipedia.org/wiki/Elon Musk

2 Objetivos

2.1 Objetivos Gerais

Avaliar se inclusão de variáveis relacionadas ao jogador tem impacto significativo em prever o desfecho da partida após a seleção dos heróis no jogo Dota2.

2.2 Objetivos Específicos

- Apresentar a temática eSports
- Apresentar a forma de coleta dos dados via API
- Apresentar e Implementar o processo de tabulação dos dados para fins de modelagem
- Ajustar, validar e comparar diferentes modelos com e sem a informação dos jogadores, atráves de métricas de desempenho

3 Materiais e Métodos

3.1 Materiais

3.1.1 Sobre os Dados

Os dados são coletados atráves de uma Application programming interface (API) da plataforma de jogos Steam¹. Este serviço é disponivel somente para usúarios que tenham comprado algum jogo da plataforma, pois, somente após a compra é disponibilizado uma chave de acesso. Cada chave pode fazer uma quantidade limitada de requisições em um curto espaço de tempo, para este trabalho cujo o objetivo é avaliar variáveis relacionadas ao jogador presente em cada partida tem-se um primeiro problema na aquisição dos dados, sendo que para cada partida coletada é necessário buscar o histórico de cada jogador presente na partida, isto faz com que a quantidade de requisições cresca tendo assim problemas na coleta dos dados. Na seção 3.1.3 é apresentado o fluxo de execução programada e dado maiores detalhes para contornar este problema.

3.1.2 Recursos Computacionais

Para as análises e desenvolvimento do trabalho será utilizado o software R (R Core Team, 2018). Para o armazenamento e a tabulação das informações será utilizado dois banco de dados, o primeiro é o MongoDB², que é um banco de dados destruturados e o segundo é o MySQL³ que é um banco de dados relacional para a tabulação dos dados.

3.1.3 Coleta dos Dados

O processo de coleta dos dados é feito atráves de programas em R que ficam coletando dados 24 horas por dias fazendo requisições na API da plataforma Steam para a aquisião de novos IDs de partidas, preparando as informações da partida e dos jogadores que estão presentes nela e armazenando no banco de dados MongoDB.

O fluxo abaixo trás mais detalhes deste processo. As caixas em cinzas é a ordem da execução/fase de cada programa.

¹ https://store.steampowered.com/about/

² https://www.mongodb.com/

³ https://www.mysql.com/

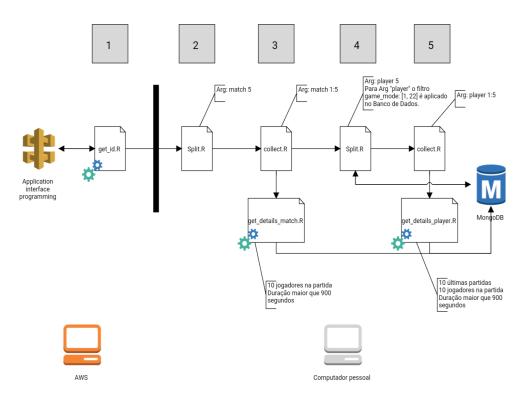


Figura 1 – Fluxo do processo de coleta dos dados

- 1. O programa **get_id.R** faz uma requisição na API e retorna com as partidas que estão acontencendo no momento da requisição, os IDs dessas partidas são processados e armazenados, e repete-se o processo de tempos em tempos (120 segundos). É necessário o armazenamento dos IDs para uma coleta posterior das informações da partida após seu termino, pois é possível coletar as informações da partida se estas ainda não terminaram. Deve-se deixa-lo rodando em uma máquina 24 horas por dia com conexão estável com a internet. Para este trabalho um computador da Amazon⁴ está sendo utilizado.
- 2. O programa **split.R** divide o arquivo gerado pelo programa **get_id.R**, que é um vetor com **n** IDs coletados que são separados em **j** partes, para cada parte uma chave de acesso é necessária, para processar mais rapidamente as informaçõesda partida.
- 3. O programa **collect.**R lê cada um dos arquivos gerados pelo programa **split.**R e cria uma instância do R, para cada um, que irá coletas as informações da partida e armazena-las no MongoDB.
- 4. Após coletar as informações da partida o arquivo **split.R** conecta-se ao banco de dados, seleciona todas as partidas coletadas e pega os IDs de todos os jogadores que compoem a partida. Após isso divide em **j** arquivos .RData, cada qual com informações do ID da partida e o ID do jogador que fez parte dela.

⁴ https://aws.amazon.com/pt/

5. Por último o programa **collect.R** processa cada arquivo com o ID da partida e o ID do jogador varrendo o histórico do jogador coletando as suas últimas 10 partidas e armazenando-as no banco de dados MongoDB.

Como exemplo, tem-se abaixo as informações de uma única partida coletada. O campo players está oculto na primeira representação pois este campo contém as informações de cada jogador na partida, e é demasiadamente longo. Para ilustrar foi selecionado apenas o primeiro jogador que está na próxima página.

Listing 3.1 – match

```
"players": ...,
    "radiant_win": false,
    "duration": 1869,
    "pre_game_duration": 90,
    "start_time": 1549408252,
    "match_id": 4394571998,
    "match_seq_num": 3799941136,
    "tower_status_radiant": 390,
    "tower_status_dire": 1974,
    "barracks_status_radiant": 51,
    "barracks_status_dire": 63,
    "cluster": 184,
    "first_blood_time": 97,
    "lobby_type": 7,
    "human_players": 10,
    "leagueid": 0,
    "positive_votes": 0,
    "negative_votes": 0,
    "game\_mode": 22,
    "flags": 1,
    "engine": 1,
    "radiant_score": 19,
    "dire_score": 36,
    "picks_bans": [
        "is_pick": false,
        "hero_id": 8,
        "team": 0,
        "order": 0
        "is_pick": false,
        "hero_id": 44,
        "team": 0,
        "order": 1
      }
    ]
 }
1
```

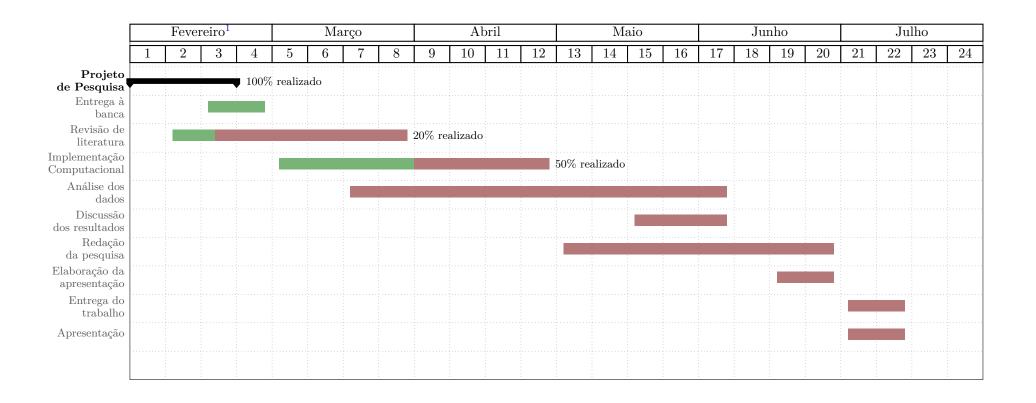
```
[{
"players": [
                                                        {
                                                          "ability": 5252,
  "account_id": 292190898,
                                                          "time": 769,
  "player_slot": 0,
                                                          "level": 6
  "hero_id": 54,
                                                        },
  "item_0": 50,
                                                        {
                                                          "ability": 5249,
  "item_1": 151,
  "item_2": 11,
                                                          "time": 846,
  "item_3": 252,
                                                          "level": 7
  "item_4": 112,
                                                        },
  "item_5": 36,
                                                        {
                                                          "ability": 5250,
  "backpack_0": 0,
  "backpack_1": 0,
                                                          "time": 975,
  "backpack_2": 0,
                                                          "level": 8
  "kills": 2,
                                                        },
  "deaths": 5,
  "assists": 6,
                                                          "ability": 5251,
                                                          "time": 1067,
  "leaver_status": 0,
  "last_hits": 201,
                                                          "level": 9
  "denies": 28,
                                                        },
  "gold\_per\_min": 440,
                                                        {
                                                          "ability": 5906,
  "xp_per_min": 462,
  "level": 18,
                                                          "time": 1158,
                                                          "level": 10
  "hero_damage": 16096,
  "tower_damage": 2198,
                                                        },
  "hero_healing": 1040,
                                                        {
                                                          "ability": 5249,
  "gold": 524,
  "gold_spent": 13110,
                                                          "time": 1202,
  "scaled_hero_damage": 9318,
                                                          "level": 11
  "scaled_tower_damage": 1216,
                                                        },
  "scaled_hero_healing": 603,
  "ability_upgrades":
                                                          "ability": 5253,
                                                          "time": 1333,
    "ability": 5250,
                                                          "level": 12
    "time": 284,
                                                        },
    "level": 1
                                                          "ability": 5249,
  },
                                                          "time": 1438,
    "ability": 5251,
                                                          "level": 13
    "time": 395,
                                                        },
    "level": 2
                                                          "ability": 5251,
                                                          "time": 1515,
    "ability": 5250,
                                                          "level": 14
    "time": 490,
                                                        },
    "level": 3
                                                        {
                                                          "ability": 5939,
                                                          "time": 1718,
    "ability": 5249,
                                                          "level": 15
    "time": 566,
                                                        },
    "level": 4
                                                          "ability": 5251,
  },
                                                          "time": 1860,
    "ability": 5250,
                                                          "level": 16
    "time": 688,
                                                        },
    "level": 5
                                                          "ability": 5252,
  },
                                                          "time": 2169,
                                                          "level": 17
},
]
                                                      }
            Listing 3.2 – "players"
```

Listing 3.3 – "players.ability_upgrades"

3.2 Métodos

Para este trabalho se fará uso tanto de Modelos Lineares Generalizados (NELDER; WEDDERBURN, 1972), em específico o modelo Logito, como modelos Bayesianos e algoritmos de Machine Learning. Parte essencial no processo de modelagem se fará no treinamento e na validação dos modelos ajustados e na criação de características, e para isso diversas técnicas poderão ser utilizadas como validação cruzada, holdount, curva ROC etc...

4 Cronograma de Atividades



REFERÊNCIAS

BERNSTEIN, P. L. Desafio aos deuses: a fascinante história do risco. [S.l.]: Gulf Professional Publishing, 1997. Nenhuma citação no texto.

NELDER, J. A.; WEDDERBURN, R. W. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, Wiley Online Library, v. 135, n. 3, p. 370–384, 1972. Citado na página 9.

NEWZOO, N. [S.l.]: Newzoo, 2018. Nenhuma citação no texto.

R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria, 2018. Disponível em: https://www.R-project.org/. Citado na página 5.

SILVA, V. d. N.; CHAIMOWICZ, L. Moba: a new arena for game ai. arXiv preprint arXiv:1705.10443, 2017. Citado na página 3.