

Raport

Przewidywanie rezultatu meczu w CS2

Wojciech Kosmalski

June 2024

Contents

1	Wstęp	2
1.1	Cechy	2
2	Zbieranie danych	2
2.1	Narzędzia	2
2.2	Schemat zbierania danych	2
2.3	Problemy	3
3	Preprocessing i analiza danych	3
4	Tworzenie modeli	3
5	Wnioski	4

1 Wstęp

Celem projektu jest stworzenie modelu klasyfikującego do przewidywania rezultatu meczu (która drużyna wygra) na podstawie danych dostępnych na stronie [HLTV](#).

1.1 Cechy

- Miejsca drużyn w rankingu HLTV
- Mapa
- Liczba fragów (fragi strzałem w głowę), asyst, śmierci
- Średnie obrażenia na rundę
- KAST - procent rund, w których zawodnik zdobył fraga lub zaliczył asystę lub przeżył lub został ztrade'owany
- Rating HLTV 2.0
- FK diff - *first kill difference* - różnica pierwszych fragów/śmierci w rundzie

2 Zbieranie danych

Zbieranie danych polegało na scrapowaniu strony [HLTV](#) i parsowaniu HTMLa. Zebrałem dane o meczach które odbyły się zarówno Online jak i na Lanie.

2.1 Narzędzia

Do scrapowania danych użyłem podstawowych bibliotek takich jak **requests**, **bs4**, **json**.

2.2 Schemat zbierania danych

Schemat zbierania danych był następujący:

1. Zescrapowanie danych z witryny <https://www.hltv.org/stats/matches> - linki do pojedynczych meczów
2. Zescrapowanie danych z witryny postaci:
<https://www.hltv.org/stats/matches/mapstatsid/idMeczu/druzynaA-vs-druzynaB>
Przykładowy URL:
<https://www.hltv.org/stats/matches/mapstatsid/177653/3dmax-vs-9ine>

3. Zscrapowanie rankingu drużyn po przekierowaniu na kolejną stronę - z poprzedniej strony przechodziłem na kolejną, aby wyciągnąć ranking drużyn.

Przykładowy URL:

<https://www.hltv.org/matches/2373004/3dmax-vs-9ine-regional-clash-arena-europe-2024>

2.3 Problemy

W trakcie scrapowania stron pojawiły się pewne problemy, mianowicie nie dostawałem poprawnej odpowiedzi od serwera, mimo że ustawiłem odpowiednio duży delay (zgodnie z zasadami "friendly scrapingu") pomiędzy kolejnymi zapytaniami. Rozwiązaniem okazało się podmienianie User-agenta w nagłówkach zapytań.

3 Preprocessing i analiza danych

Uznałem, że ranking drużyn jest istotną cechą, więc mecze, w których przynajmniej jedna z drużyn nie miała rankingu, nie brałem pod uwagę.

Preprocessing i analizę danych można zobaczyć pod tym linkiem [Notebook](#).

4 Tworzenie modeli

Postanowiłem, że przetestuję, jak poradzi sobie model, w zależności od tego na jakich cechach go wytrenuję. Zestawienia cech na jakie się zdecydowałem:

- A score, B score
- A rank, B rank
- A rank, B rank, A tier, B tier
- A AVG K, A AVG D, B AVG K, B AVG D
- A AVG ADR, B AVG ADR
- A AVG KAST, B AVG KAST
- A AVG FK Diff, B AVG FK Diff
- A AVG Rating 2.0, B AVG Rating 2.0
- map, A tier, B tier

Dokładny przebieg tworzenia i trenowania modeli jest pod tym linkiem [Notebook](#).

Features	Precision
wszystkie poza zdobytymi rundami	0.925256
A score, B score	1.000000
A rank, B rank,	0.599974
A rank, B rank, A tier, B tier	0.592492
A AVG K, A AVG D, B AVG K, B AVG D	0.928686
A AVG ADR, B AVG ADR,	0.918427
A AVG KAST, B AVG KAST	0.885356
A AVG FK Diff, B AVG FK Diff	0.710682
A AVG Rating 2.0, B AVG Rating 2.0	0.928894
map, A tier, B tier	0.572067

Table 1: Precyzja modelu w zależności od wybranych cech

5 Wnioski

Z tabeli [1] można wyciągnąć następujące wnioski:

1. Proces tworzenia modelu jest poprawny, bo dla cech A score i B score, które jednoznacznie określają zwycięzcę meczu, nie myli się w żadnym przypadku.
2. Przewidywanie zwycięzcy meczu na podstawie rankingu drużyn nie jest już takie proste i pomimo, że nie wynosi 50/50, to 60-procentowa precyzja nie jest najlepszym wynikiem.
3. Najprostsze K/D ratio z bardzo dobrą dokładnością przewiduje zwycięzcę meczu - nie jest to zaskoczeniem, gdyż zazwyczaj drużyna, która więcej zabija i mniej ginie zwycięża, pomimo faktu iż w CS nie jest to jedyny sposób wygrywania rund. Jednak słowem klucz jest słowo "rund", bo faktycznie pojedyncze rundy można wygrywać, pomimo większych strat w postaci zabitych zawodników (CT/T save'ują broń na następną rundę pomimo przewagi liczebnej), jednak na ogół rundę wygrywa ta drużyna, która wyeliminuje większą liczbę graczy z drużyny przeciwnej.
4. Średnie obrażenia na rundę wypadają podobnie jak K/D ratio, co również jest logiczne, bo jedno jest powiązane z drugim.
5. KAST wypada już trochę gorzej, ale wciąż precyzja jest wysoka.
6. FK diff nie ma już takiego wpływu na wynik meczu, chociaż wciąż jest istotny - im więcej wygranych opening duelów tym większa szansa na wygranie meczu.
7. Próba predykcji zwycięzcy meczu na podstawie tierów drużyn i mapy nie zakończyła się zbyt dobrze. Wychodzi tutaj kwestia tego, że różne drużyny lepiej sobie radzą na różnych mapach i nie jest to zależne od ich rankingu,

a raczej od ich przygotowania taktycznego, charakterystyki zawodników i innych czynników, które nie zostały w tym projekcie uwzględnione.

8. Problem z predykowaniem zwycięzcy meczu na podstawie cech, które wybrałem do analizy, jest taki, że większości z nich nie znamy dopóki mecz się nie zakończy, co znacząco ogranicza użyteczność takiego modelu.

Cały projekt jest dostępny na moim GitHubie https://github.com/chewmanji/msid_raport