WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI
POLITECHNIKA KRAKOWSKA

Metody i narzędzia analizy dużych zbiorów danych

Autorzy:
inż. Jakub Skoczewski
inż. Przemysław Skoczewski
inż. Łukasz Januszewski

Prowadzący: *mgr inż. Jacek Tchórzewski*

Kraków, 2022

Przedmiot: Metody i narzędzia analizy dużych zbiorów danych

Prowadzący: Jacek Tchórzewski, Daniel Grzonka

Zespół projektowy: Łukasz Januszewski, Przemysław Skoczewski, Jakub Skoczewski

Wykorzystane narzędzia: Visual Studio Code, MongoDB Atlas, MongoDB Compass, Jupyter Notebook

Strona wykorzystana w celu pozyskania danych: **https://www.hltv.org/**

Aktualność analizowanych danych: Stan na 16.04.2022

HLTV (poprzednio Half-Life Television) – jest to forum i serwis informacyjny, który obejmuje profesjonalne wiadomości e-sportowe na temat gry Counter-Strike: Global Offensive, turnieje jak i związane z nimi statystyki.

Cel projektu

Tematem naszej pracy było stworzenie webcrawlera, który pobierze dane na temat 30 czołówych zespołów światowej klasy, każdy składający się z 5 zawodników, którzy osiągają najwyższe wyniki w e-sportowych zawodach w grze Counter Strike – Global Offensive, czyli w tzw. "majorach". Następnie dane te zostały zapisane w dokumencie MongoDB o nazwie player_list, a później w kolekcji o takiej samej nazwie. Po uprzednim zapisaniu danych w formacie umożliwiającym lepsze zarządzanie nimi (JSON), dane zostały wykesportowane przy użyciu narzędzia MongoDB Compass do pliku .csv, który został wczytany do Jupyter Notebook, aby tam przeanalizować pozyskane dane. Wyniki prac zostaną przedstawione poniżej.

Dane, które zostały pozyskane ze strony to:

- **team** – Definiuje do jakiej drużyny należy zawodnik. *(Typ danych – string)*
- **nick** – Pseudonim zawodnika. *(Typ danych – string)*
- **name** – Imię zawodnika. *(Typ danych – string)*
- **nationality** – Kraj pochodzenia zawodnika. *(Typ danych – string)*
- **age** – Wiek zawodnika. *(Typ danych – integer)*
- **kdratio (Kill/Death ratio)** – Stosunek zabójstw do śmierci zawodnika. To znaczy, że jeżeli zawodnik uzyskał 10 zabójstw i 5 śmierci to jego stosunek zabójstw do śmierci wynosi 2.0. *(Typ danych – float)*
- **headshots** – Procent strzałów w głowę jaki uzyskał zawodnik. *(Typ danych – float)*
- **majors_won** – Ilość wygranych międzynarodowych turniejów, tzw. "majorów" w karierze zawodnika. *(Typ danych – integer)*
- **majors_played** - Ilość rozegranych międzynarodowych turniejów, tzw. "majorów" w karierze zawodnika. *(Typ danych – integer)*

Dane dodane przez MongoDB:

- **_id** – identyfikator dodawany domyślnie przez MongoDB. *(Typ danych - integer)*

Czym jest webcrawler?

Crawler, nazywany też pająkiem lub robotem to program, który zbiera informacje na temat zawartości i struktury stron internetowych w internecie. Najpopularniejszym tego typu programem są Googleboty.

Web crawler może być wykorzystywany na bardzo różne sposoby, wszystko zależy od tego, jak go zaprogramujemy. Dla przykładu, w SEO najpopularniejszymi crawlerami bedą oczywiście boty indeksujące (przede wszystkim od Google).

Przykłady użycia webcrawlerów:

- monitoring stron internetowych i zmian na nich zachodzących
- dodawanie komentarzy
- analizowanie linków
- analizowanie stron pod kątem SEO (automatyczne audyty SEO)
- tworzenie kontaktowych baz danych

Crawlery są tak zaprojektowane, żeby nie indeksować całości stron za każdym razem. Pamiętajmy, że jedna strona będzie posiadała ogromne zasoby w swoim obrębie, a inna będzie np. prostą stroną wizytówką, której indeksacja jest znacznie prostsza i szybsza. W ten sposób wyróżnia się dwie metody skanowania zasobów w Internecie:

- Deep crawl – jest to dogłębna analiza całości witryny, czyli jej struktury, kodu źródłowego itp.
- Fresh crawl – ten rodzaj skanowania będzie miał zastosowanie w przypadku stron, które często są uaktualniane. Crawler będzie zatem badał tylko obszar, który został zmieniony, nie całość obszernej witryny.

W skrócie, web crawler to inaczej m.in. bot indeksujący. Na podstawie analizowania stron internetowych, ich struktury, kodów źródłowych itp., zbiera informacje, które następnie zestawia w sprecyzowany sposób.

CS:GO – co to jest?

CS:GO to skrót od Counter-Strike: Global Offensive. Jest to tak zwana "strzelanka pierwszoosobowa", czyli gra komputerowa z widokiem z oczu postaci wyposażonej w różnego rodzaju broń. CS:GO daje możliwość rozgrywki w trybie online wielu graczom, dzięki czemu mierzyć mogą się ze sobą ludzie z całego świata.

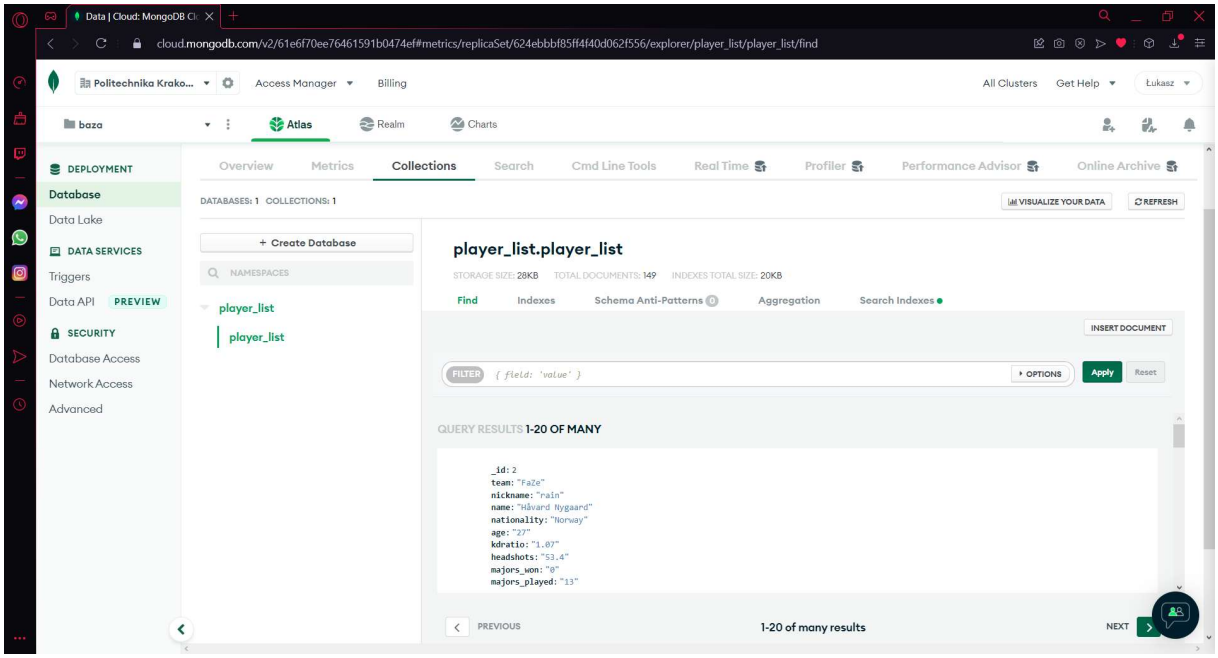
W CS:GO można opowiedzieć się po stronie drużyny terrorystów lub antyterrorystów. Zadaniem każdej z ekip jest wyeliminowanie przeciwników, lub wykonanie określonego zadania. W przypadku terrorystów chodzi o podłożenie bomby lub obronę zakładników. Zadaniem antyterrorystów jest zaś ochrona określonego miejsca przez podłożeniem bomby lub uratowanie zakładników. W grze występuje siedem różnych trybów, z których każdy ma odrębne cechy charakterystyczne. Gracze do dyspozycji mają broń dokładnie odwzorowującą prawdziwe egzemplarze, a rozgrywkę mogą toczyć na jednej z setek map.

Counter-Strike: Global Offensive to nie tylko rozrywka dla wielbicieli gier, ale też profesjonalny sport! Najważniejszymi turniejami dla graczy CS:GO są tzw. "majory" - zawody sponsorowane przez

Valve, wydawcę gry. Zawody w CS:GO są też częścią mistrzostw świata gier komputerowych Electronic Sports World Cup.

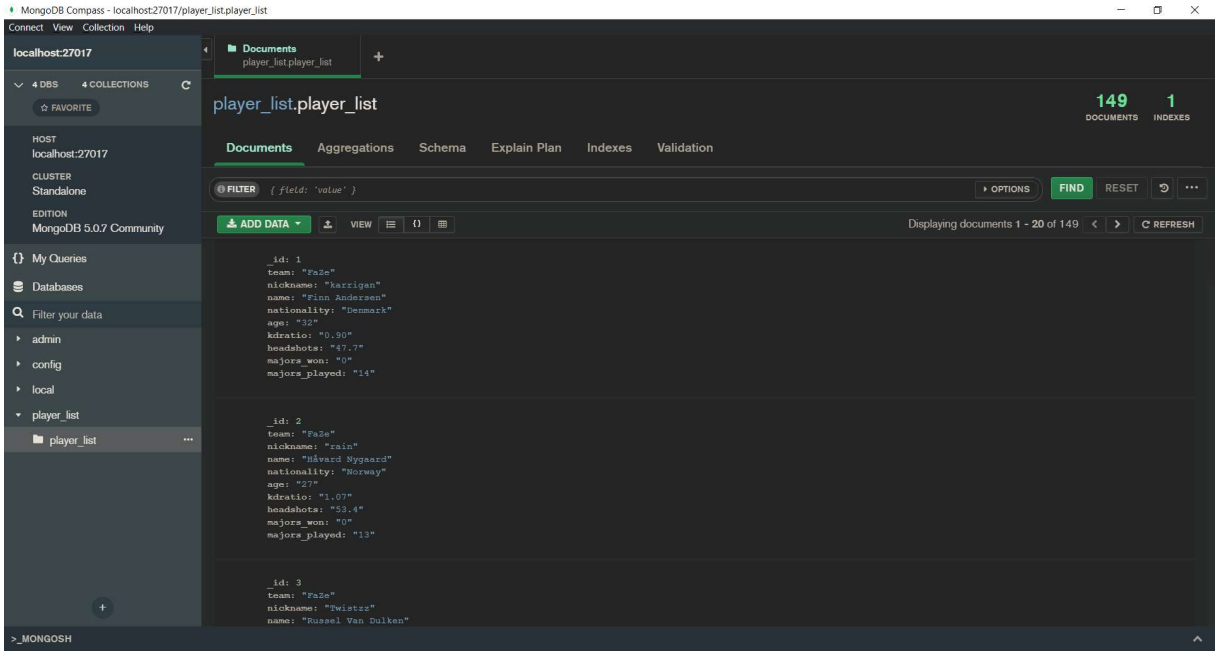
Opis implementacji

Do utworzenia webcrawlera zespół projektowy wybrał bibliotekę BeautifulSoup4 w języku Python. Dane zostają pozyskane na podstawie nazw bloków w języku HTML, które osoba pisząca kod definiuje w programie. W przypadku pobierania danych odnośnie rozgrywek w e-sporcie, zostały wybrane najważniejsze parametry służące do rozróżnienia zawodników, a także porównania wyników ich pracy z innymi zespołami. Tymi parametrami są na przykład stosunek zabójstw w grze do zgonów zawodnika, wiek zawodnika czy ilość rozegranych i wygranych turniejów. Dane pozyskiwane przez webcrawlera są od razu konwertowane w pętli do JSONa tak, aby były one przejrzystsze do odczytu I późniejszej analizy. Po utworzeniu webcrawlera następuje połączenie z bazą danych w chmurze MongoDB, czyli z narzędziem Mongo Atlas. Właśnie tam są kierowane wszystkie dane, które program za nas "wyciąga" z serwisu HLTV. Są one pobieranie z opóźnieniem 0.3s, ponieważ strona ta jest zabezpieczona przed szybszym pobieraniem danych i takie próby są przez nią traktowane jak atak DDoS I serwis blokuje nam do siebie dostęp. Po udanym data crawlingu możemy zobaczyć jakimi danymi dysponujemy.



Jak widać na powyższym zrzucie ekranu, dane zostały zapisane w poprawny sposób w MongoDB Atlas. Dane są przechowywane w dokumencie player_list I w kolekcji o tej samej nazwie.

Kolejnym krokiem było zainstalowanie serwera MongoDB na lokalnej maszynie, a następnie przystosowanie kodu źródłowego aplikacji tak, aby pozyskiwane dane były zapisane w lokalnej wersji bazy danych przechowywanej w pamięci komputera.



Na powyższym zrzucie ekranu widać prawidłowe pobranie danych do lokalnej bazy danych localhost, pracującej na porcie 27017.

Po wykonaniu powyższych czynności zdecydowaliśmy się poddać dane pewnej analizie tak, abyśmy mogli wyciągnąć z nich jakieś wnioski. W tym celu wykesportowaliśmy dane z bazy lokalnej do pliku w formacie arkusza Excel, a następnie wczytaliśmy je przy użyciu Jupyter Notebook.

```
In [10]: player = pd.read_excel('player_list.xlsx')
         player = player[player.columns[1:]]
         player
```

Out[10]:

	age	headshots	kdratio	majors_played	majors_won	name	nationality	nickname	team
0	32	47.7	0.90	14	0	Finn Andersen	Denmark	karrigan	FaZe
1	27	53.4	1.07	13	0	H?vard Nygaard	Norway	rain	FaZe
2	22	59.7	1.12	5	0	Russel Van Dulken	Canada	Twistzz	FaZe
3	22	46.0	1.17	5	0	Robin Kool	Estonia	ropz	FaZe
4	21	30.5	1.20	1	0	Helvijs Saukants	Latvia	broky	FaZe
...	...	...	...	...	...	...	...	...	...
144	25	55.2	1.10	3	0	Leonid Vishnyakov	Russia	chopper	Spirit
145	20	35.5	1.24	0	0	Abdul Gasanov	Russia	degster	Spirit
146	18	53.9	1.14	0	0	Boris Vorobiev	Russia	magixx	Spirit
147	19	44.1	1.02	0	0	Pavel Ogloblin	Russia	s1ren	Spirit
148	18	42.4	1.07	0	0	Robert Isyanov	Russia	Patsi	Spirit

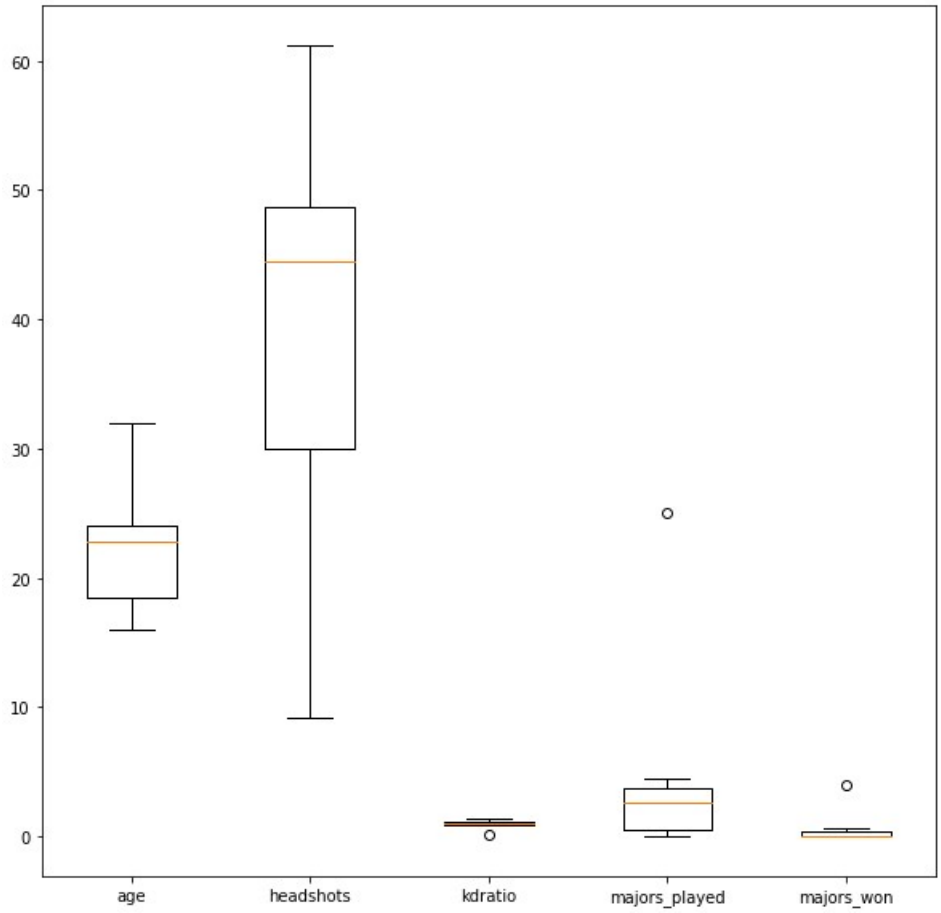
Pozyskane dane prezentują się tak, jak na powyższym zrzucie ekranu.

Kolejno dane pozostały poddane analizie, która przedstawi nam dane tj.: średnia, odchylenie standardowe czy skrajne wartości występujące w każdej z kolumn.

	_id	age	headshots	kdratio	majors_played	majors_won
count	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000
mean	75.000000	22.838926	44.474497	1.068054	2.644295	0.214765
std	43.156691	3.121562	9.243896	0.094395	4.420581	0.693247
min	1.000000	16.000000	21.000000	0.890000	0.000000	0.000000
25%	38.000000	21.000000	39.000000	0.990000	0.000000	0.000000
50%	75.000000	23.000000	46.000000	1.070000	1.000000	0.000000
75%	112.000000	25.000000	51.400000	1.140000	3.000000	0.000000
max	149.000000	32.000000	61.200000	1.340000	25.000000	4.000000

Jak można łatwo zauważyć, topowych graczy zapisanych na stronie HLTV na dzień 22.04.2022 jest 149.

- Średnia ich wieku wynosi 22,83 roku, najmłodszy gracz ma 16 lat, a najstarszy 32 lata.
- Średnia ilość zabójstw przez strzał w głowę na turnieju wynosi 44,47, najmniejsza wartość to 21, a rekordzista ma ich aż 61,2.
- Średni stosunek zabójstw do zgonów gracza wynosi 1.068, najniższa wartość to 0,89, a najwyższy 1,34.
- Średnia ilość rozegranych meczy turniejowych wynosi 2,64 meczu, najniższa wartość wynosi 0,00, a najwyższa wartość wynosi 25,00.
- Średnia ilość wygranych meczy turniejowych przez graczy wynosi 0,21, najniższa wartość wynosi 0,00, a najwyższa wartość wynosi 4,00.



Dane przedstawione w boxplotach prezentują się tak, jak powyżej. Odstające dane sugerują nam, że wśród przenalizowanych graczy mogą znajdować się tacy, którzy grają dużo dłużej niż inni. Z tego powodu mogą mieć więcej sukcesów, niż ich młodsi koledzy, np. 26-letni gracz, który przegrał setki meczy więcej na scenie światowej, od jego 18-letniego kolegi z zespołu ma dużo więcej rozegranych I wygranych meczy od niego.