

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI



---

# Statystyki gracza w zależności od wyniku gry w League of Legends

---

Sprawozdanie z laboratorium

AUTOR

**Michał Kapała**

nr albumu: **260399**

kierunek: **Informatyka**

*14 czerwiec 2022*

### Streszczenie

Praca przedstawia predykcję najkorzystniejszych statystyk potrzebnych do wygrania gry przez danego gracza League of Legends. Dataset został stworzony w wyniku scrapowania strony [leagueofgraphs.com](https://leagueofgraphs.com) za pomocą biblioteki BeautifulSoup. Pobrane dane zostały następnie oczyszczone oraz użyto metody regresji liniowej do predykcji najlepszych statystyk. Dla każdej statystyki otrzymaliśmy dwie predykcje - wartość danej statystyki w grze przegranej oraz w wygranej.

## 1 Wstęp – sformułowanie problemu

Autor chce zwiększyć swoje szanse na wygrywanie. Pozwoli mu to na analizowanie, które przewidywe statystyki dla wygranych i przegranych gier znacząco się różnią (ułatwienie w poszukiwaniu błędów zawodnika w przegranych grach).

## 2 Opis danych

Wielkość datasetu 100 wierszy.

Kolumna "wynik" - wynik rozgrywki. Zbiór wartości: 0, 1. (int)

Kolumna "kills" - liczba zabójstw. (int)

Kolumna "deaths" - liczba śmierci. (int)

Kolumna "assists" - liczba asyst. (int)

Kolumna "cs" - liczba pokonanych stworów. (int)

Kolumna "KP" - procentowe uczestnictwo we wszystkich zabójstwach drużyny. Zbiór wartości: większe od 0, mniejsze lub równe 100. (int)

## 3 Opis rozwiązania

Dane zostaną pobrane ze strony <https://www.leagueofgraphs.com>. Dostęp do danych uzyskano za pomocą biblioteki BeautifulSoup. Baza danych zawiera informacje o 100 rozgrywkach wraz z 6 cechami je określającymi.

Używając metody *regresji liniowej* na danych uzyskano model pozwalający na określenie statystyk typowych dla wygranych jak i przegranych gier danego gracza.

## 4 Rezultaty obliczeń

### 4.1 Plan badań

Zbiór danych zostanie podzielony na dwie części: treningową i testową w stosunku 75:25.

## 4.2 Wyniki obliczeń

Wykres zależności wyniku od ilości **zabójstw**:

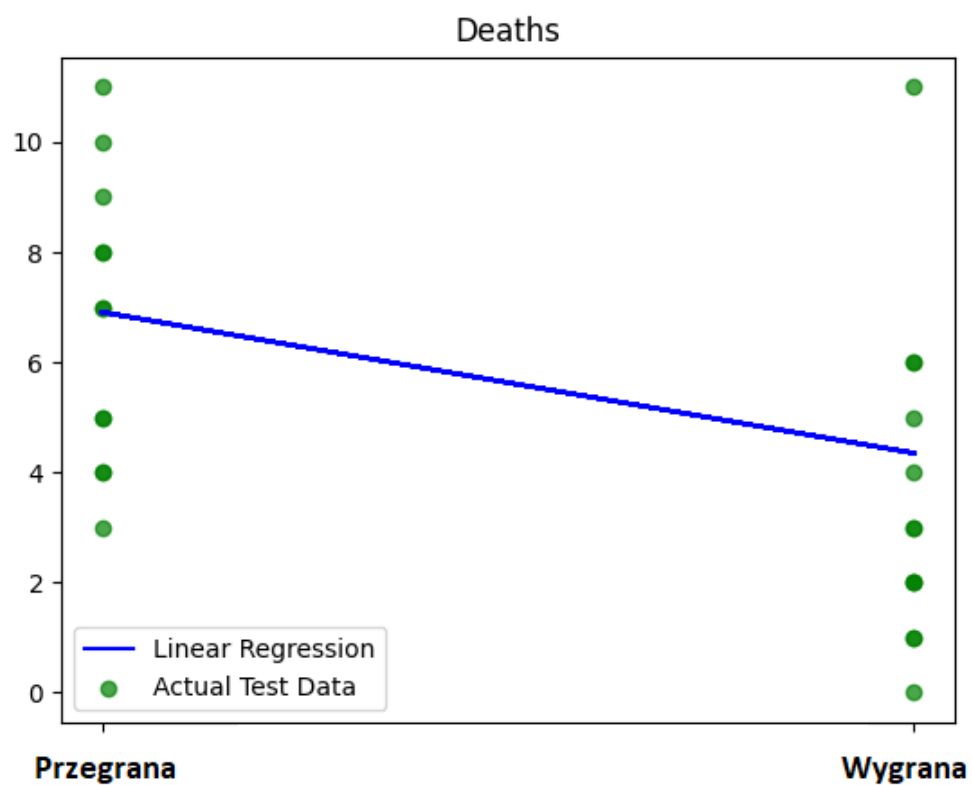


### Wyniki regresji liniowej:

Wartość statystyki obliczona dla wygranej gry: 11.095238095238095

Wartość statystyki obliczona dla przegranej gry: 6.53125

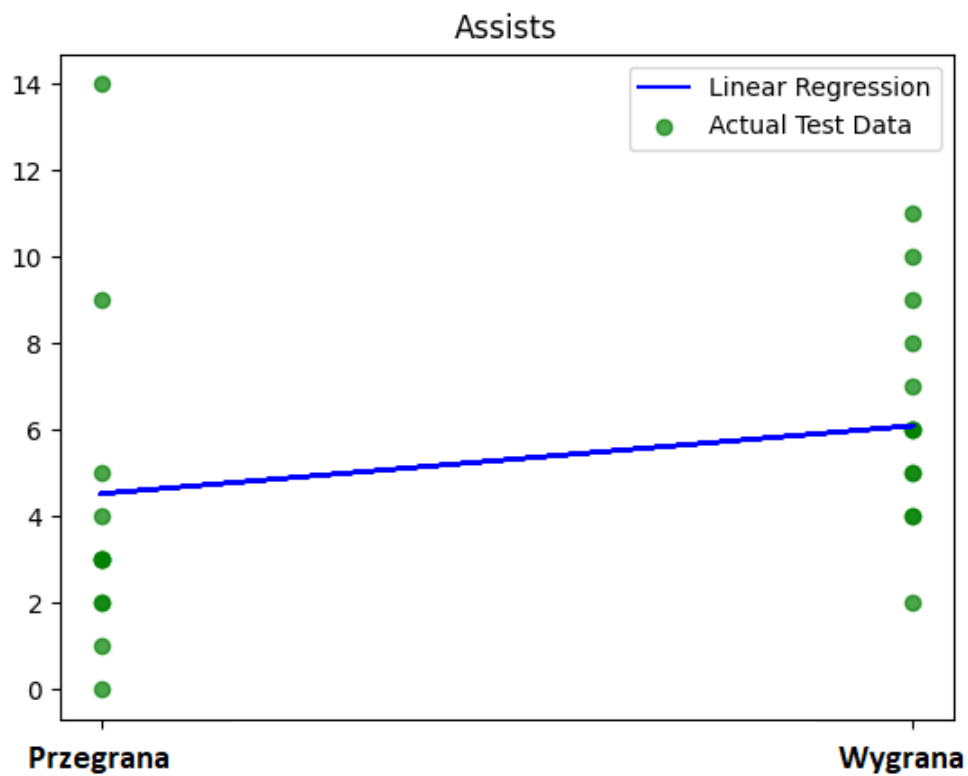
Wniosek: O przegranych grach tego gracza mogła decydować zbyt pasywna gra.

**Wyniki regresji liniowej:**

Wartość statystyki obliczona dla wygranej gry: 4.527777777777778

Wartość statystyki obliczona dla przegranej gry: 6.7894736842105265

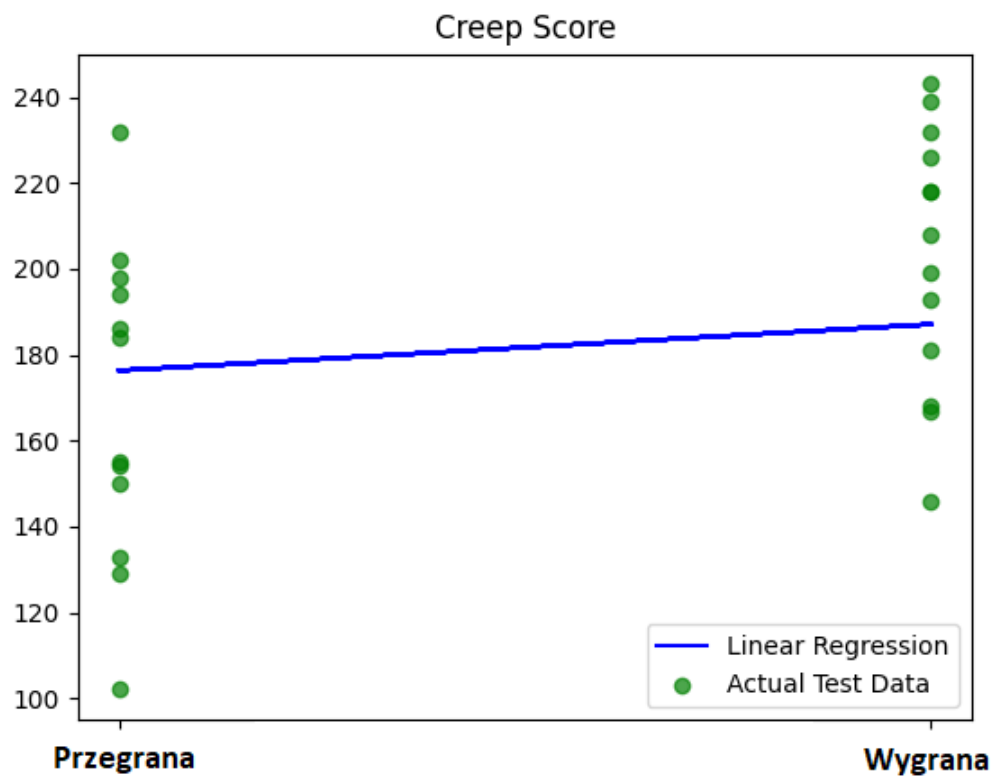
Wniosek: O przegranych grach tego gracza mogły decydować błędy w wybieraniu walk.

**Wyniki regresji liniowej:**

Wartość statystyki obliczona dla wygranej gry: 6.195121951219512

Wartość statystyki obliczona dla przegranej gry: 4.303030303030304

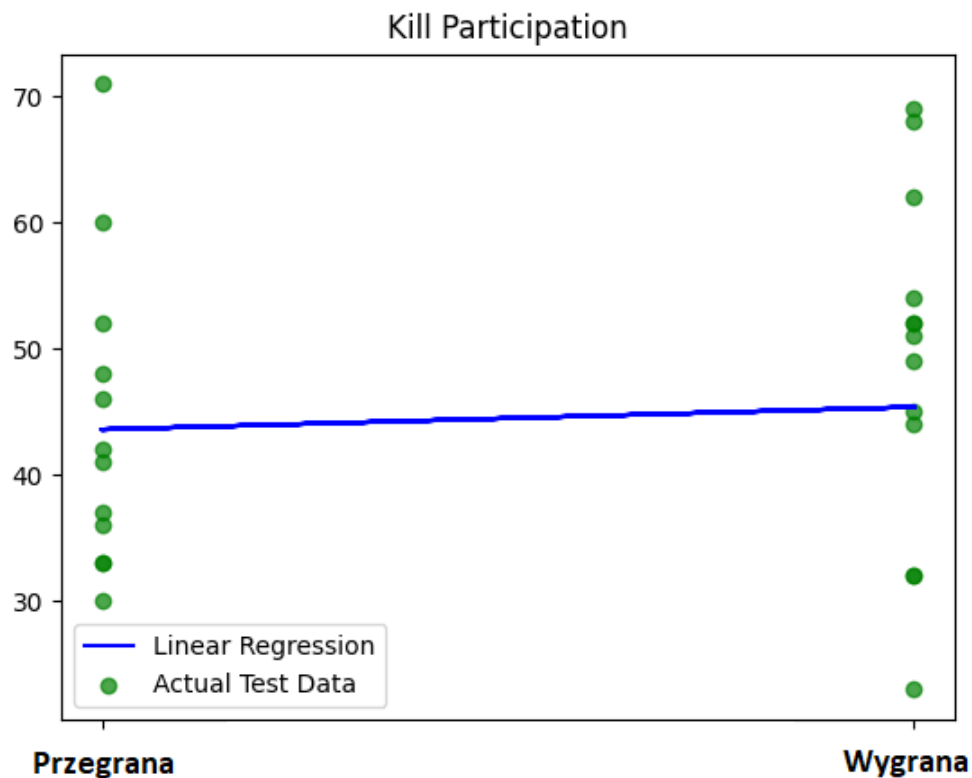
Wniosek: Ta statystyka nieznacznie się różni w zależności od wygranych i przegranych gier, co oznacza, że najprawdopodobniej nie ma w tym przypadku znacznych błędów gracza.

**Wyniki regresji liniowej:**

Wartość statystyki obliczona dla wygranej gry: 191.5

Wartość statystyki obliczona dla przegranej gry: 179.85294117647058

Wniosek: Ta statystyka nieznacznie się różni w zależności od wygranych i przegranych gier, co oznacza, że najprawdopodobniej nie ma w tym przypadku znacznych błędów gracza.



#### Wyniki regresji liniowej:

Wartość statystyki obliczona dla wygranej gry: 46.07692307692307

Wartość statystyki obliczona dla przegranej gry: 45.08571428571428

Wniosek: Ta statystyka nieznacznie się różni w zależności od wygranych i przegranych gier, co oznacza, że najprawdopodobniej nie ma w tym przypadku znacznych błędów gracza.

## 5 Wnioski

Przedstawiona metoda pozwala na przygotowanie modelu wytrzymałego na nowe dane. Uzyskany model daje szczególnie ciekawe dane w przypadku bardziej niestabilnego stylu gry, który bardziej uzależniałby swoje statystyki od gry pozostałych graczy, a więc i od wyniku rozgrywki, pokazując, na które statystyki gracz powinien zwrócić szczególną uwagę.

## A Dodatek

Kody źródłowe umieszczone zostały w repozytorium github:

<https://github.com/MichalKapa/LinearRegressionOnLoLGames>.