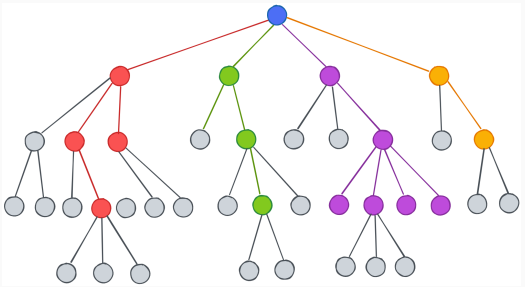


Analiza Danych z branży gamingowej

Drzewa decyzyjne i regresyjne

Stanisław Cabaj, Grzegorz Grodecki, Wojciech Szyszka

25 czerwca 2023

AGH WMS

Opis zbioru danych

Podczas naszej analizy korzystaliśmy ze zbioru **”Video Game Sales with Ratings”** dostępnej na stronie kaggle.com pod następującym linkiem: Video Game Sales.
Nasz zbiór powstał przy wykorzystaniu informacji pochodzących ze strony ”Metacritic” gdzie dostępne są oceny użytkowników gier oraz strony ”vgchartz” gdzie dostępne są informacje dotyczące sprzedaży tych gier.

Zbiór danych podzielony jest na 16 kolumn: Name, Platform, Year of Release, Genre, Publisher, NA Sales, EU Sales, JP Sales, Other Sales, Global Sales, Critic Score, Critic Count, User Score, User Count, Developer oraz Rating gdzie kolumny ”Sales” oznaczają sprzedaż gier w różnych częściach świata podane w milionach kopii. Dane obejmują gry wydawane do 2016 roku.

Cel analizy

Analizę przeprowadziliśmy w celu zbadania upodobań użytkowników gier video. Informacje pozyskane w ten sposób mogłyby być przydatne dla firm zajmujących się tworzeniem oraz sprzedażą gier video. Postanowiliśmy sprawdzić czy na podstawie dostępnych danych można przewidzieć zadowolenie graczy w taki sposób, aby wspomniane firmy mogły skorzystać z naszych potencjalnych wyników w swoich planach biznesowych.
Sprawdziliśmy też powiązania pomiędzy poszczególnymi cechami, z których najciekawsze zaprezentujemy poniżej.
W dalszej części próbowaliśmy również dokonywać predykcji ilości sprzedanych gier, co z punktu widzenia producentów gier wydaje się być najistotniejszym czynnikiem biznesowym.

Przygotowanie zbioru danych

W celu uzyskania rzetelnych wyników musieliśmy wykonać kilka operacji przygotowujących opisaną bazę danych do użycia w zastosowanych metodach predykcyjnych.

Głównym problemem były braki danych dla większości obserwacji, przy czym te braki dotyczyły wielu kolumn jednocześnie. Z tego powodu zdecydowaliśmy się na usunięcie wszystkich obserwacji, w których pojawiały się braki. Ze względu na dużą liczbę danych wyjściowych nie było to aż tak kosztowne.

Ponadto trzeba było zachować ostrożność w kontekście typów zmiennych. Wstępnie wartości z kolumn User Score i Year Of Release były typu tekstowego (chr), wartości User Score zamieniliśmy na typ rzeczywisty (numeric), natomiast Year Of Release na typ całkowity (integer).

Musieliśmy również usunąć znaki specjalne występujące w nazwach twórców i wydawców gier - algorytm **C5.0** nie działał z uwzględnieniem tych znaków.

Używane narzędzia

Naszym celem było dokonanie predykcji zmiennej **User Score** odzwierciedlającej poziom zadowolenia klientów. Ponieważ ocena zazwyczaj nadawana jest w skali od 1 do 10, podeszliśmy do naszego problemu jak do problemu klasyfikacji. Aby taka analiza była możliwa, wartości User Score zamieniliśmy na liczby całkowite (jako ”sufit” z pierwotnych wartości). Ze względu na dużą liczbę zmiennych jakościowych o wielu różnych wartościach, wybraliśmy algorytm **C5.0** do budowania drzew decyzyjnych wraz z boostingiem na poziomie 100 prób, korzystając z metody **AdaBoost**.

W trakcie analizy stwierdziliśmy, że warto również sprawdzić skuteczność metod drzew regresyjnych. Szczególnie zainteresowani byliśmy algorytmem drzew modeli regresji **Cubist**, ponieważ w zbiorze danych występują zauważalne zależności liniowe. Algorytmy drzewowe są również wartościowe dzięki stosunkowo łatwej interpretacji i możliwości wykorzystania ich do stwierdzenia zależności między zmiennymi.

Weryfikacja modeli

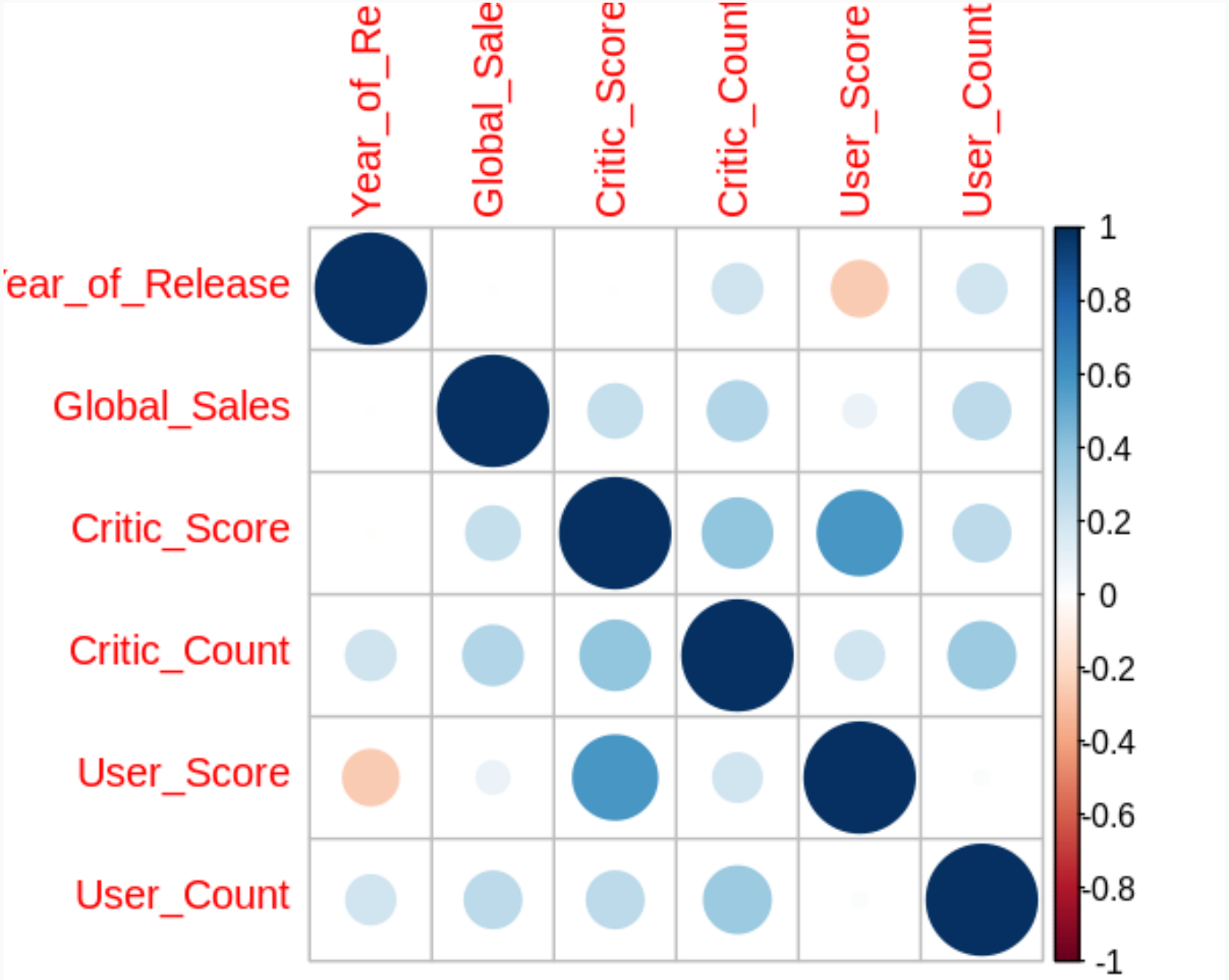
W celu poprawnej weryfikacji naszych wyników zbudowaliśmy pomocnicze funkcje określające średnią wartość błędu bezwzględnego oraz kwadratowego pomiędzy naszymi predykcjami oraz prawdziwymi wartościami. Do policzenia jaki odsetek obserwacji został dobrze przyporządkowany posłużyła nam przygotowana przez nas funkcja accuracy. Kilkakrotnie posłużyliśmy się też tabelką ”CrossTable” z zajęć porównującą wyniki predykcji z wartościami prawdziwymi.

Aby korzystać z powyższych narzędzi dokonaliśmy podziału naszych danych na zbiór uczący oraz zbiór testowy. Zbiór uczący zawierał 6000 z 6825 losowo wybranych obserwacji. Wybraliśmy wielkość zbioru uczącego na poziomie około 88 procent wszystkich obserwacji, aby jak najlepiej nauczyć nasze modele. Reszta obserwacji została przyporządkowana do zbioru testowego i posłużyła nam do sprawdzenia zdolności predykcyjnych modeli.

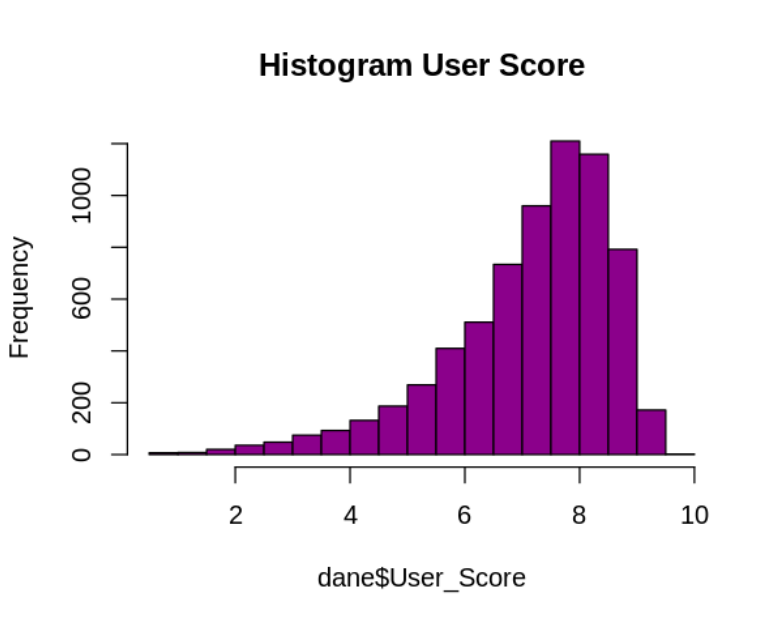
Przebieg analizy 1

Po niezbędnym przygotowaniu danych, pierwszym krokiem naszej analizy było zbadanie zależności pomiędzy poszczególnymi zmiennymi. W używanych algorytmach nigdy nie używaliśmy wszystkich kolumn z danymi dotyczącymi sprzedaży, ponieważ wartość sprzedaży światowej jest równa sumie pozostałych wartości sprzedaży, więc jednej z tych kolumn można nie używać bez straty informacji. Udało nam się również zauważyć dość dużą zależność pomiędzy kolumnami User Score oraz Critic Score wskazującą na powiązania pomiędzy ocenami krytyków oraz użytkowników.

Macierz Korelacji



Histogram User Score i drzewo klasyfikacji



Budowa drzewa klasyfikacji:

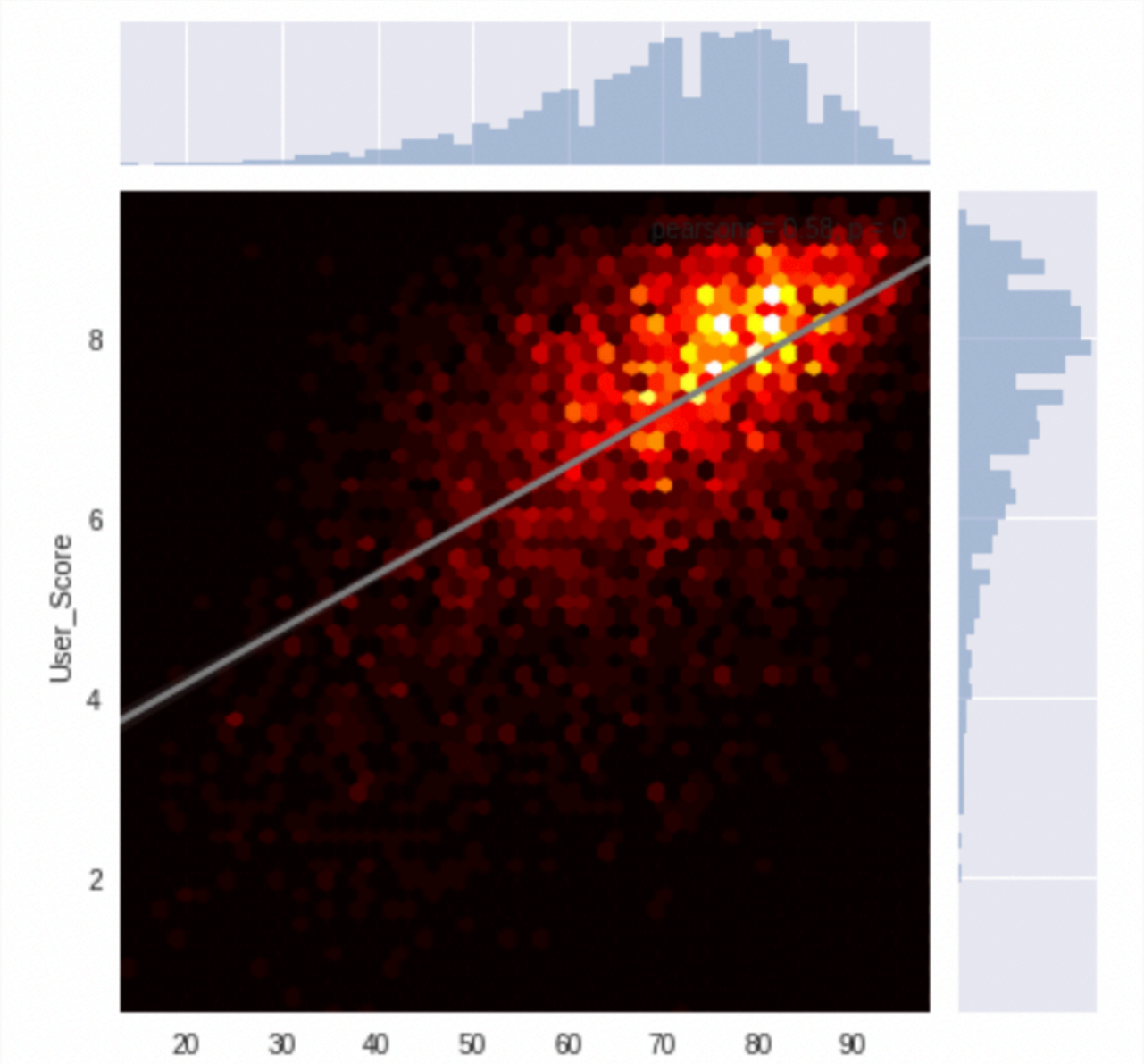
```
65  #install.packages('C50')
66  library("C50")
67  games_model <- C5.0(games_train_cl[, c(1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 15)],
68                      as.factor(games_train_cl$User_Score))
69  games_model
```

Przebieg analizy 2

Kolejnym krokiem naszej analizy było przygotowanie drzewa decyzyjnego klasyfikującego gry, którego zadaniem była predykcja poziomu User Score. Po zastosowaniu boostingu, udało nam się osiągnąć wyniki na poziomie dokładności równej 42 procent skuteczności. Zauważyliśmy też, że uzyskane błędy były na poziomie 0,89 bezwzględny oraz 1,78 kwadratowy. Co pokazuje całkiem dobre, aczkolwiek umiarkowane możliwości predykcji naszego algorytmu.
W powyższej analizie nie wzięliśmy pod uwagę kolumny Developer - ze względu na jej dość dużą różnorodność która powodowała nadmierne dopasowanie modelu do danych. (skuteczność spadała do 38,5 procent)
Zauważyliśmy też, że użycie do predykcji tylko samej kolumny Critic Score może dać wyniki na poziomie 40 procent skuteczności, a użycie dwóch kolumn: Critic Score oraz Year of Release na poziomie 41 procent. Więc w razie potrzeby istnieje możliwość zdecydowanego uproszczenia modelu, przy braku dużej zmiany wyniku.

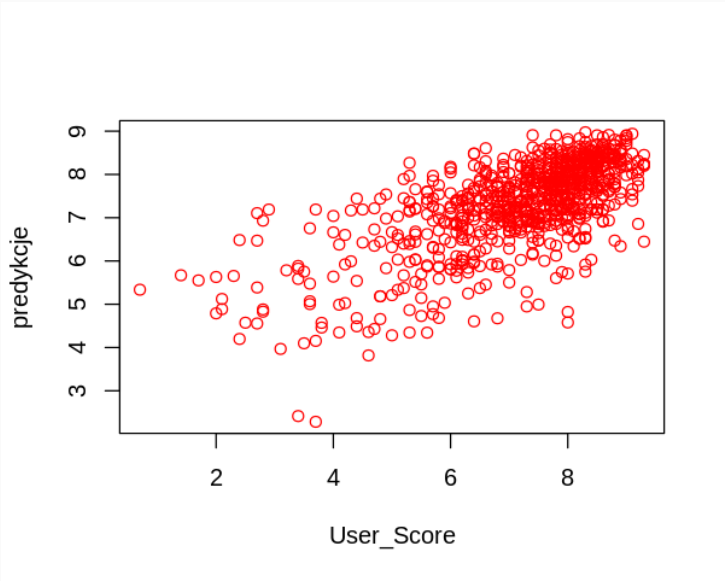
Critic Score vs User Score

Liniowa zależność pomiędzy Critic Score oraz User Score:



Przebieg analizy 3 - Drzewa Regresji

Ze względu na zaobserwowaną dużą liniową zależność pomiędzy User Score oraz Critic Score naszym kolejnym krokiem było rozważenie drzewa modeli regresji. Wyniki pokazują, że ta metoda daje lepsze wyniki predykcyjne. Minimalny średni błąd bezwzględny był na poziomie ok. 0,78 i pozwolił w dość dobry sposób przewidywać wysokość User Score. Jest to również wyraźnie lepszy wynik od uzyskanego z użyciem zwykłego drzewa regresyjnego z pakietu rpart (średni błąd bezględny na poziomie ok. 0,86). Poniżej wykres rozrzutu predykcji z drzewa Cubist i danym ze zbioru testowego.



Przebieg analizy 4- budowa drzewa regresji oraz Global Sales

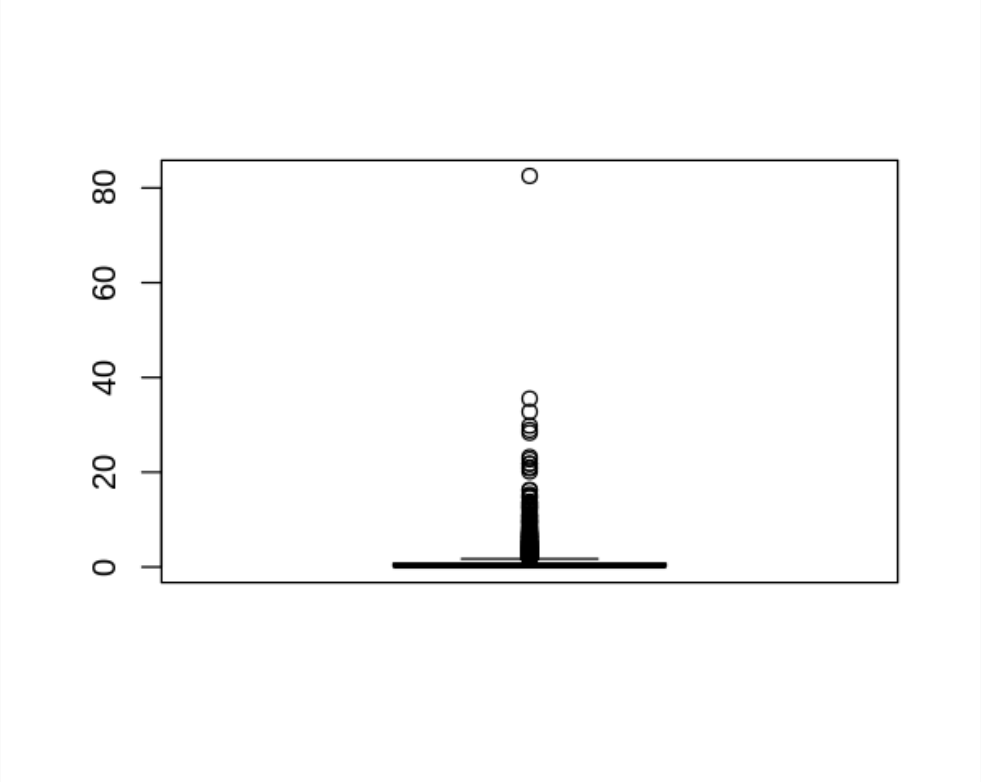
Budowa optymalnego drzewa regresji:

```
136 library(Cubist)
137 games_cubist <- cubist(x = games_train_reg[, c(1, 2, 3, 4, 10, 11, 13)],
138                      y = games_train_reg[, 12])
139 games_cubist
```

Ze względów biznesowych, z perspektywy wydawców bardzo często bardziej interesujący od poziomu zadowolenia klientów może być wynik sprzedaży gier na świecie. Z tego powodu postanowiliśmy w kolejnej części analizy zająć się predykcją poziomu sprzedaży. Ze względu na oczywiste powiązania, w naszej dalszej analizie nie braliśmy pod uwagę kolumn zawierających informacje na temat sprzedaży w poszczególnych miejscach.
Niestety w przypadku zmiennej Global Sales dużą przeszkodą okazała się duża liczba obserwacji odstających nawet przy skali 1:1000000 co pokazuje poniższy rysunek:

Global Sales- obserwacje odstające

Wykres ramka-wąsy dla zmiennej Global Sales:



Z wykresu widać bardzo duże zagęszczenie obserwacji w przedziale 0-1 oraz obecność dużych obserwacji odstających. Należy pamiętać, że skala Global Sales jest w milionach egzemplarzy.

Przebieg analizy 5

```
181 sales_cubist <- cubist(x = sales_train[, c(1, 2, 3, 4, 6, 7, 8, 9, 10, 11)],
182                        y = sales_train[, 5])
183 sales_cubist
```

Ze względu na dość duże skoncentrowanie obserwacji pomiędzy 0 a 1 milionem egzemplarzy nasz błąd bezwzględny w modelu regresji wyniósł 0,5, natomiast kwadratowy był na poziomie 1,75, za co z kolei w dużej mierze odpowiadały obserwacje odstające. Niestety próba zastosowania algorytmu **C5.0** do zmiennej Global Sales nie dała dobrych wyników. Ze względu na skalę wyrażoną w milionach i duże zagęszczenie obserwacji pomiędzy 0 a 1, sensowny test można było przeprowadzić jedynie dla gęstszej podziałki i dawał on bardzo słabe wyniki.

Podsumowanie

Podsumowując naszą analizę - największą skuteczność otrzymaliśmy całkowicie pomijając część zmiennych. W załączonym skrypcie algorytmy są użyte dla tych zbiorów zmiennych, dla których dają najlepsze wyniki z uzyskanych w trakcie analizy pod względem wprowadzonych metryk. Bez większego zaskoczenia, najistotniejszym parametrem do predykcji User Score było Critic Score, czego można było się spodziewać po dużej korelacji między nimi. Co ciekawe, najlepszy znaleziony algorytm dla predykcji User Score nie używa żadnej z kolumn odnoszących się do sprzedaży.
Jeśli chodzi o predykcję liczby sprzedanych gier to rezultaty były trochę trudniejsze w interpretacji ze względu na wcześniej wspomniane komplikacje, lecz wciąż mogą być pomocne w biznesie.
Na koniec dodamy, że jeśli chodzi o korelację między krytykami, a użytkownikami - jest ona na w miarę wysokim poziomie, co dobrze świadczy o wykonywanej pracy przez krytyków :)