System rekomendacyjny

Kamila Ciężabka

Kwiecień 2024

Spis treści

1 Wstęp

Celem niniejszego projektu jest rozwój systemu rekomendacyjnego zdolnego do przewidywania ocen filmów użytkowników na podstawie danych o poprzednio ocenianych przez nich filmach, a także porównanie różnych metod faktoryzacji macierzy oraz optymalizacji, takich jak:

- NMF (Non-negative Matrix Factorization),
- SVD1, SVD2 (Singular Value Decomposition),
- SGD (Stochastic Gradient Descent),

aby zidentyfikować technikę, która najlepiej i najdokładniej radzi sobie z predykcją nieznanych ocen użytkowników.

System rekomendacyjny został opracowany w języku Python. Zastosowana została biblioteka argparse, służąca do zarządzania argumentami przekazywanymi przez wiersz poleceń, co znacząco zwiększa elastyczność użytkowania programu.

2 Opis i przygotowanie danych

Projekt wykorzystuje zbiór danych ratings.csv, który zawiera informacje o ocenach filmów przyznawanych przez użytkowników. Dane składają się z około 100000 ocen dokonanych przez około 600 użytkowników i dotyczą blisko 9000 filmów. Struktura zbioru danych jest reprezentatywna dla realnych systemów rekomendacyjnych, gdzie znaczna część danych jest nieznana, w tym przypadku nieznane wartości stanowią około 98% wszystkich danych. Ten fakt podkreśla wyzwanie związane z estymacją brakujących wartości.

Każda ocena zawiera identyfikator użytkownika, identyfikator filmu oraz przyznaną ocenę w skali od 0 do 5. Zbiór danych został podzielony na zbiór treningowy i testowy. Dla każdego użytkownika losowo wybrano około 90% jego ocen do zbioru treningowego, a pozostałe 10% trafiło do zbioru testowego.

W kontekście projektu, zbiory testowy i treningowy zostały zareprezentowane jako macierze - $\mathbf{Z}$ (zbiór treningowy) i $\mathbf{T}$ (zbiór testowy), o jednakowych wymiarach 600×9000. Po wykonaniu predykcji ocen, jakość działania danego algorytmu będzie obliczana jako pierwiastek błędu średniokwadratowego:

$$RMSE = \sqrt{\frac{1}{|\mathbf{T}|} \sum_{(u,m)\in\mathbf{T}} (\mathbf{Z'}[u,m] - \mathbf{T}[u,m])^2},$$

gdzie $\mathbf{Z'}$ to macierz $\mathbf{Z}$ po zastosowaniu algorytmu.

2.1 Metody wypełniania macierzy

Dla zastosowania algorytmów NMF oraz SVD, konieczne jest, aby macierz wejściowa $\mathbf{Z}$ była kompletna, tj. nie zawierała żadnych wartości brakujących. Należy zatem zastosować odpowiednią metodę imputacji brakujących wartości, która minimalizuje RMSE w kontekście estymowanych ocen.

Zaproponowane zostały następujące strategie imputacji brakujących ocen użytkowników w macierzy $\mathbf{Z}$:

- Wypełnianie zerami - najprostsze podejście, które w przypadku rozważanego systemu okazało się najmniej efektywne.
- Wypełnianie średnią oceną filmu - brakujące oceny wypełniane są po kolumnach, średnią z danej kolumny; skuteczność tej metody była niższa w porównaniu do imputacji opartej na średniej ocen użytkownika, co wskazuje na mniejsze znaczenie ogólnej popularności filmu w kontekście indywidualnych preferencji użytkownika.
- Wypełnianie średnią ocen użytkownika - brakujące wartości są uzupełniane wierszami, średnią ocen jakie dany użytkownik przyznał innym filmom; wyniki wykazały, że ta metoda jest bardziej skuteczna w porównaniu do imputacji średnią oceną filmu, sugerując, że indywidualne preferencje użytkownika mają większy wpływ na skuteczność rekomendacji, niż zdanie ogółu użytkowników na temat filmu.
- Wypełnianie średnią ważoną ocen użytkownika i filmu: ta strategia polega na uzupełnieniu brakujących wartości przy użyciu średniej ważonej; gdzie waga 4 przypisana jest średniej ocenie użytkownika, a waga 1 średniej ocenie filmu. Wagi zostały dobrane doświadczalnie. W przypadku pojawienia się w zbiorze treningowym filmów, które nie zostały ocenione przez żadnego użytkownika, brakujące wartości w danej kolumnie uzupełniane są średnią danego użytkownika.

Ostatecznie, metoda wypełniania średnią ważoną ocen użytkownika i filmu z wagami odpowiednio 4 i 1, w finalnej wersji systemu rekomendacyjnego została zaimplementowana dla każdego z trzech rozważanych algorytmów: NMF, SVD i SVD2. Metoda ta przyniosła najlepsze wyniki w zakresie minimalizacji błędu RMSE, co sugeruje, że kombinacja tych dwóch czynników z odpowiednio dobranymi wagami może skutecznie przewidywać preferencje użytkowników wobec nieznanych filmów.

3 NMF

Pierwszym z zaproponowanych algorytmów, które mogą pomóc skutecznie przewidzieć przyszłą ocenę użytkownika jest algorytm NMF, czyli Non-negative matrix factorization (metoda faktoryzacji macierzy nieujemnej).

3.1 Działanie algorytmu

Ogólne działanie NMF polega na aproksymacji nieujemnej macierzy $\mathbf{Z} \in \mathbb{M}_{n\times d}(\mathbb{R})$ jako iloczynu dwóch nieujemnych macierzy $\mathbf{W} \in \mathbb{M}_{n\times r}(\mathbb{R})$ oraz $\mathbf{H} \in \mathbb{M}_{r\times d}(\mathbb{R})$:

$$\mathbf{Z} \approx \mathbf{WH}.$$

Każda kolumna macierzy $\mathbf{W}$ reprezentuje cechę, która powtarza się wśród d n-wymiarowych punktów macierzy $\mathbf{Z}$. Są to podstawowe elementy za pomocą których

możemy zrekonstruować przybliżenia do wszystkich oryginalnych punktów danych. Natomiast każda kolumna macierzy **H** zawiera informacje na temat wag związanych z macierzą **W**. Dzięki temu każdy punkt danych w kolumnie **Z** może być przybliżony poprzez kombinację liniową nieujemnych wektorów reprezentowanych jako kolumny macierzy **W**.

Celem NMF jest minimalizacja dystansu pomiędzy **Z** i **WH**, czyli znalezienie

$$\arg\min_{\mathbf{W},\mathbf{H}} d(\mathbf{Z}, \mathbf{WH}),$$

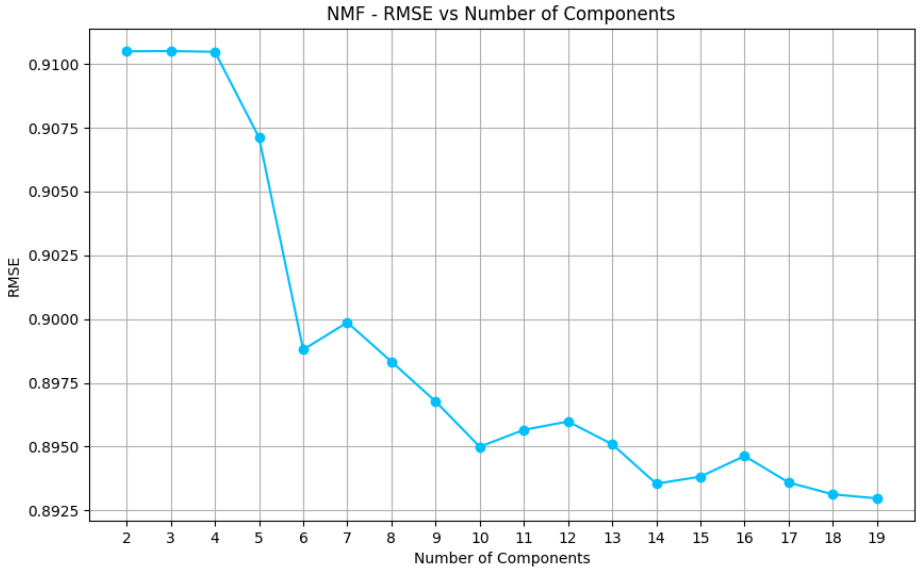
gdzie $r \leq d$ oraz d jest wybraną funkcją odległości pomiędzy macierzami - zazwyczaj wybór tej funkcji zależy od konkretnego zastosowania oraz charakterystyki danych. Celem jest otrzymanie dobrej aproksymacji dla parametru $r \ll \text{rank}(\mathbf{Z})$.

Stosując NMF można otrzymać macierze faktoryzowane o znacznie mniejszych wymiarach niż wymiar iloczynu macierzy.

3.2 Wybór wymiaru macierzy

Jak opisano wyżej, kluczowym aspektem algorytmu NMF jest odpowiednie dobranie wymiaru r dla macierzy **W** i **H**. W kontekście niniejszego projektu wartość r dobrana została doświadczalnie - sprawdzone zostało, jak zmienia się wartość błędu RMSE w zależności od wymiaru r.

Wyniki zostały uśrednione po wykonaniu 10 różnych podziałów na zbiory treningowy i testowy i zaprezentowane zostały na poniższym wykresie:



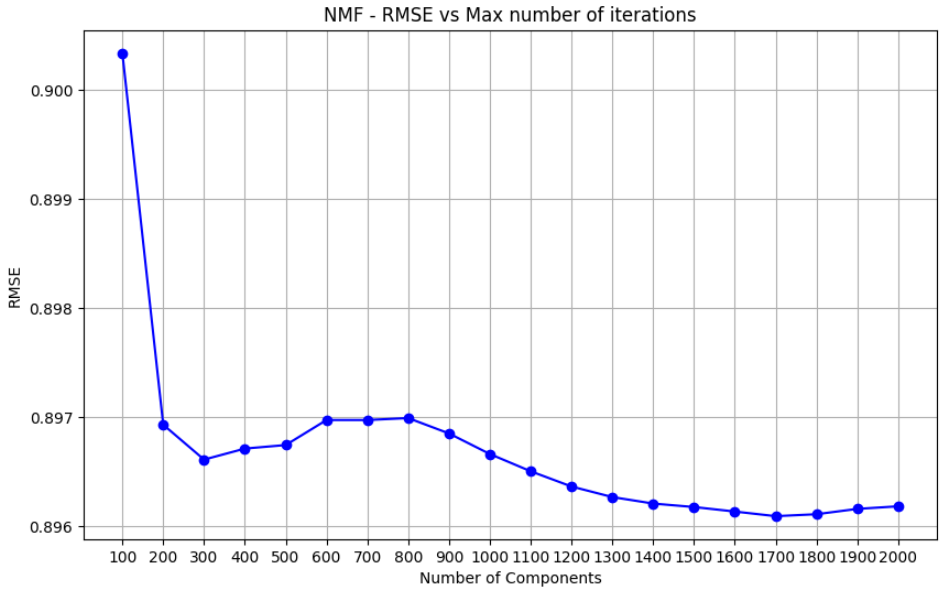
Rysunek 1: RMSE w zależności od r w NMF

Na wykresie widzimy, że przy większej wartości r, wartość RMSE zaczyna się stabilizować się (zmiany dla kolejnych wartości r nie są znaczące). Większa liczba r prowadzi

do zwiększenia się czasu trwania algorytmu, w związku z tym, finalnie jako optymalna wartość do algorytmu została wybrana liczba $r = 14$.

3.3 Liczba iteracji

Jednym z parametrów konfiguracyjnych w algorytmie NMF jest maksymalna liczba iteracji, która domyślnie ustawiona jest na 200. W ramach przeprowadzonych eksperymentów, zbadano wpływ zwiększenia maksymalnej liczby iteracji na jakość modelu. Doświadczenie zostało powtórzone 10 razy, a wyniki przedstawione na poniższym wykresie:



Rysunek 2: RMSE w zależności od liczby iteracji w NMF

Zwiększenie liczby iteracji może prowadzić do pewnego spadku wartości RMSE, poprawiając przy tym jakość aproksymacji. Jednak różnice między RMSE są dość marginalne, a do tego wraz ze wzrostem liczby iteracji, wzrasta także czas wykonania algorytmu. Przykładowo, w jednym z początkowych eksperymentów, dla 1200 iteracji, RMSE osiągnęło wartość 0.9005, natomiast dla domyślnej liczby iteracji równej 200, wynik RMSE wynosił 0.901 - przy czym czas obliczeń zwiększył się z 3.27s do 18.97s.

Biorąc pod uwagę zarówno jakość aproksymacji, jak i wydajność obliczeniową, zdecydowano o zachowaniu domyślnej liczby iteracji równej 200, jako optymalnego ustawienia dla badanego algorytmu.

4 SVD1

Drugim zaproponowanym algorytmem, jest algorytm SVD, czyli Singular Value Decomposition (rozkład według wartości osobliwych), który podobnie jak NMF jest metodą matematyczną służącą do redukcji wymiaru macierzy.

4.1 Działanie algorytmu

Algorytm zakłada, że każda macierz rzeczywista $\mathbf{Z} \in \mathbb{M}_{n\times d}(\mathbb{R})$ może być przedstawiona w postaci rozkładu SVD:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

co oznacza, że chcemy aproksymować macierz $\mathbf{Z}$ za pomocą trzech macierzy. W rozważanym przypadku zapiszemy równanie jako:

$$\mathbf{Z} \approx \mathbf{U}_r\Lambda_r\mathbf{V}_r^T,$$

gdzie:
- $\mathbf{U}_r$ jest macierzą ortogonalna o wymiarach $n \times r$, czyli $\mathbf{U}_r^T\mathbf{U}_r = \mathbf{U}_r\mathbf{U}_r^T = \mathbf{I}$,
- Λ_r jest macierzą diagonalną o wymiarach $r \times r$, zawierającą wartości osobliwe macierzy $\mathbf{Z}$, zazwyczaj uporządkowane nierosnąco.
- $\mathbf{V}_r^T$ jest transpozycją macierzy ortogonalnej o wymiarach $r \times d$.

Indeks dolny r odnosi się do zredukowanej liczby komponentów używanych w aproksymacji SVD i oznacza, że macierz $\mathbf{U}$ odcinamy do r kolumn, macierz $\mathbf{V}$ do r wierszy, a macierz Λ do r kolumn i wierszy.

Aby móc przybliżyć macierz $\mathbf{Z}$ w opisany sposób, musi być ona kompletna, czyli nie może zawierać żadnych brakujących wartości.

Można zdefiniować macierze $\mathbf{W}$ i $\mathbf{H}$ odpowiednio jako $\mathbf{W} = \mathbf{U}_r$ i $\mathbf{H} = \Lambda_r\mathbf{V}_r^T$, co pozwala przedstawić macierz $\mathbf{Z}$ jako produkt dwóch macierzy o mniejszym wymiarze, jednocześnie zachowując kluczowe informacje i strukturę oryginalnych danych.

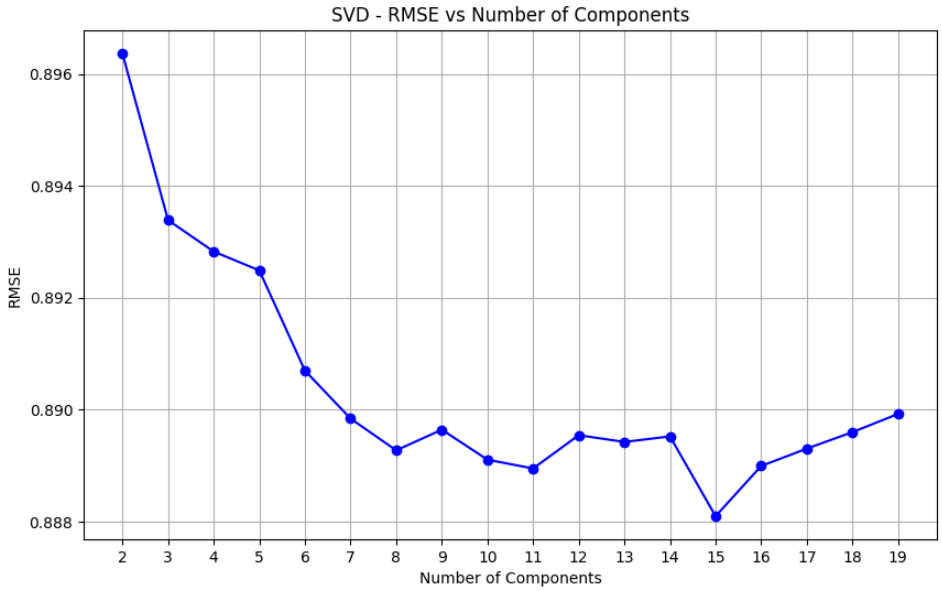
W tej konwencji, w zgodzie z opisanym uprzednio algorytmem NMF, można zapisać:

$$\mathbf{Z} \approx \mathbf{W}\mathbf{H}.$$

4.2 Wybór wymiaru macierzy

Podobnie jak w NMF, kluczową rzeczą jest wybór parametru r, który pozwala zbalansować dokładność aproksymacji.

Doświadczalnie sprawdzone zostało, jak dla różnych wartości r zmienia się wartość błędu RMSE. Rozważone zostały wartości $r \in [2, 19]$. Eksperyment przeprowadzono 15 razy, dla różnych podziałów na zbiór treningowy i testowy, a uśrednione wyniki zostały przedstawione na wykresie.



Rysunek 3: RMSE w zależności od r w SVD1

Na podstawie wykresu można zauważyć, że wraz ze zwiększaniem r, do pewnego momentu średni błąd maleje, co wskazuje na poprawę dokładności modelu. Jednak gdy r jest za duże, RMSE zaczyna wzrastać, co może sugerować przeuczenie. Na podstawie przedstawionej analizy, jako optymalny wymiar dla algorytmu SVD1 wybrany został wymiar $r = 15$.

5 SVD2

Kolejnym zaproponowanym algorytmem, mogącym skuteczniej poradzić sobie z predykcją nieznanych ocen użytkowników, jest SVD2, będący rozwinięciem standardowego algorytmu SVD opisanego powyżej.

5.1 Działanie algorytmu

Algorytm SVD2 wykorzystuje podejście iteracyjne, zwiększając dokładność przybliżenia brakujących danych. W każdej iteracji powtarzany jest algorytm SVD1, z parametrem r takim, jak dobrano powyżej. Algorytm wykonuje następujące kroki:

1. Stosuje SVD do macierzy, gdzie brakujące wartości są zastąpione pewną metodą imputacji.

2. Oblicza przybliżoną macierz ocen $\mathbf{Z}_{\text{approx}}$ z wykorzystaniem dekompozycji otrzymanych z SVD.

3. Aktualizuje tylko te elementy w macierzy $\mathbf{Z}$, które były pierwotnie niezdefiniowane (NaN), wykorzystując nowe przewidywania z $\mathbf{Z}_{\text{approx}}$.

5.2 Kryterium stopu

Algorytm wykorzystuje trzy główne kryteria stopu podczas iteracji, które pomagają określić, kiedy proces powinien zostać zakończony:

- **Zbieżność RMSE**: Iteracje są kontynuowane do momentu, gdy różnica pomiędzy RMSE dwóch kolejnych iteracji jest mniejsza niż zadany próg, w tym przypadku okreslony jako 1e-4. Jeżeli ta różnica jest mniejsza niż próg, uznaje się, że algorytm zbiegł do stabilnego rozwiązania. To kryterium pomaga uniknąć niepotrzebnego wydłużania czasu obliczeń, gdy osiągnięto już odpowiednią dokładność.
- **Wzrost RMSE**: Jeśli RMSE zacznie wzrastać między iteracjami, algorytm jest zatrzymywany przedwcześnie. Jest to sygnał, że dodatkowe iteracje mogą prowadzić do przeuczenia lub destabilizacji wyników.
- **Liczba iteracji**: Jeżeli liczba iteracji przekroczy próg równy 100, algorytm zostanie zakończony, celem zachowania optymalnego czasu wykonywania algorytmu.

W większości prób wartość RMSE osiągała zbieżność lub wzrastała dla liczby iteracji z przedziału [3, 5], jednak w ostatecznej implementacji algorytmu dokładna liczba iteracji nie jest z góry określona i zależy od wymienionych kryteriów stopu.

6 SGD

Metoda SGD, czyli Stochastic Gradient Descent (Stochastyczny Spadek Gradientu), to iteracyjna metoda optymalizacji funkcji celu o odpowiednich własnościach. Zastępuje ona rzeczywisty gradient (obliczony z całego zestawu danych) jego estymacją (obliczoną z losowo wybranego podzbioru danych).

6.1 Działanie algorytmu

Poprzednio zaproponowane metody zakładały, aby przed przybliżaniem macierzy $\mathbf{Z}$ za pomocą dwóch mniejszych macierzy, najpierw dokonać imputacji brakujących wartości. Problem ten może być przeformułowany w następujący sposób: Dla danej macierzy $\mathbf{Z}$ o wymiarach $n \times d$, celem jest znaleźć macierze $\mathbf{W}$ o wymiarach $n \times r$ i $\mathbf{H}$ o wymiarach $r \times d$, która zminimalizuje funkcję straty. Zaimplementowany algorytm skupia się na minimalizacji podstawowej dla SGD funkcji, czyli kwadratu różnicy między oryginalnymi a przybliżonymi wartościami, bez dodatkowego terminu regularyzacji:

$$\arg\min_{\mathbf{W},\mathbf{H}} \sum_{(i,j):z_{ij}\neq ?} (z_{ij} - w_i^T h_j)^2,$$

gdzie w_i^T jest i-tym wierszem z $\mathbf{W}$), h_j jest j-tą kolumną z $\mathbf{H}$, a znak '(?)' oznacza brakujące wartości. Zaimplementowany algorytm korzysta także z parametru α (współczynnik uczenia), kontrolującego wielkość kroku wykonywanego w kierunku przeciwnym do gradientu w procesie optymalizacyjnym. Parametr ten wpływa na to, jak szybko algorytm będzie zbiegać do poszukiwanego minimum.

Ogólnym celem SGD jest więc dostosować macierze **W** i **H** tak, aby wartości w macierzy **Z** były jak najbliżej wartości wyliczonych przez model. Algorytm SGD działa przez wybranie losowego elementu (tutaj para indeksów i, j), a nie przez operowanie na całej macierzy za jednym razem.

Po wybraniu elementu (i, j) obliczany jest błąd jako różnica między rzeczywistą wartością a wartością przewidywaną przez model. Następnie, używając tego błędu, obliczane są gradienty funkcji straty względem w_i i h_j które są używane do aktualizacji tych wektorów w kierunku zmniejszania błędu.

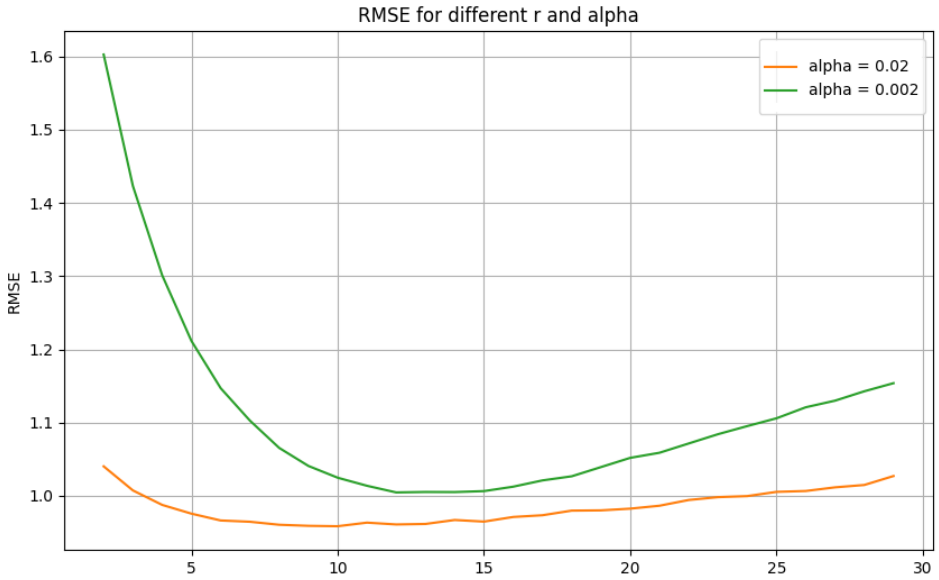
Wykonywane są następujące kroki:

1. Inicjalizacja dwóch macierzy **W** i **H**, o wymiarach odpowiednio $n \times r$ i $r \times d$, z losowymi wartościami.
2. Stworzenie listy indeksów, dla których znane są wartości z_{ij} (nie są NaN).
3. Losowe wybieranie pary indeksów (i, j) z listy indeksów na każdej iteracji.
4. Obliczenie błędu jako różnicy między rzeczywistą wartością z_{ij} a wartością obliczoną jako iloczyn skalarny i-tego wiersza **W** i j-tej kolumny **H**.
5. Obliczenie gradientów funkcji błędu względem wierszy **W** i kolumn **H**.
6. Aktualizacja wierszy **W** i kolumn **H** poprzez odejmowanie gradientów pomnożonych przez współczynnik uczenia α.

6.2 Wybór wymiaru r i współczynnika uczenia

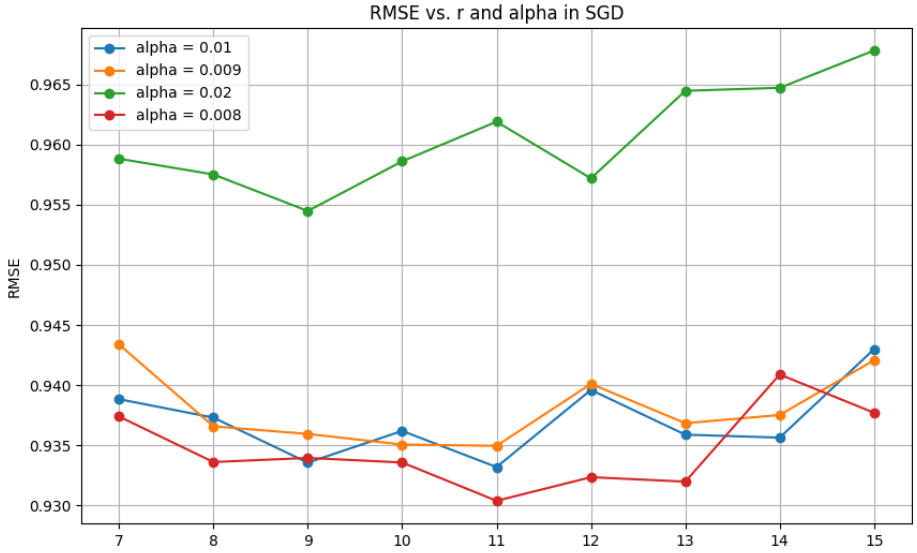
W przypadku algorytmu SGD, na dobre przewidywanie ocen przez algorytm, wpływ ma odpowiedni wybór wymiaru r, współczynnika uczenia α oraz liczby iteracji.

Poniższy wykres został stworzony dla jednej próby, przedstawia wartości RMSE w zależności od wymiaru $r \in [2, 29]$ oraz dwóch wybranych wartości $\alpha \in \{0.002, 0.02\}$.



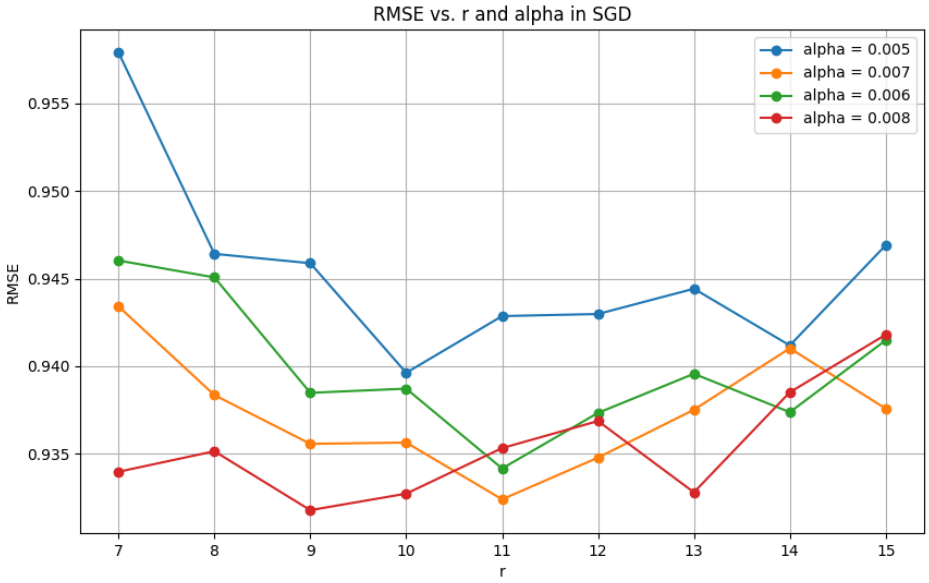
Rysunek 4: RMSE w zależności od r i α w SGD

Najniższe RMSE było osiągane przy większym z rozważanych współczynników α oraz dla wartości r w okolicach przedziału $[7, 15]$. W związku z tym stworzony został wykres z zawężeniem do r do tego przedziału i rozważeniem większej ilości współczynników α:



Rysunek 5: RMSE w zależności od r i α w SGD

Jako, że najmniejsze RMSE uzyskiwane było przy najmniejszej rozważanej wartości $\alpha = 0.008$, ostatni raz doświadczenie zostało przeprowadzone dla $\alpha \in \{0.005, 0.006, 0.007, 0.008\}$, wyniki zostały uśrednione po 10-krotnym powtórzeniu dla różnych podziałów na zbiory testowe:



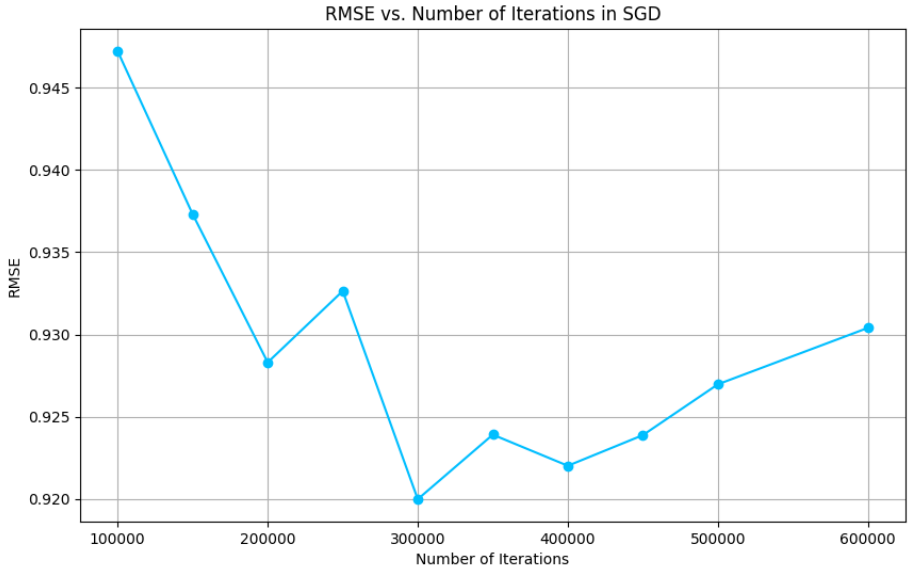
Rysunek 6: RMSE w zależności od r i α w SGD

Co wpłynęło na finalny wybór wartości $r = 13$ oraz $\alpha = 0.008$.

6.3 Liczba iteracji

Na wpływ skuteczności algorytmu SGD w kontekście rozważanego systemu rekomendacyjnego, wpływ ma także odpowiednia liczba iteracji.

Poniższy wykres pokazuje zmiany wartości RMSE w zależności od liczby iteracji, dla wybranego wcześniej $r = 13$ oraz $\alpha = 0.008$, wyniki są uśrednione po 10-krotnym powtórzeniu doświadczenia dla różnych podziałów na zbiory testowe i treningowe:



Rysunek 7: RMSE w zależności od liczby iteracji w SGD

Początkowo, wraz ze wzrostem liczby iteracji RMSE maleje, jednak przy zbyt dużej liczbie iteracji, wartość RMSE wzrasta. Finalnym wyborem była liczba 300000 iteracji.

6.4 Potencjalne ulepszenia

Prezentowany algorytm skupiał się na minimalizacji podstawowej funkcji straty, jednak możliwe jest rozszerzenie powyższego modelu, poprzez dodanie terminu regularyzacji, który pomaga w zapobieganiu nadmiernemu dopasowaniu modelu do danych. Potencjalne ulepszenia algorytmu SGD mogłyby uwzględnić wykorzystanie momentum. Momentum pomaga algorytmowi optymalizacyjnemu SGD zmierzać w odpowiednim kierunku, gromadząc poprzednie aktualizacje gradientów i dodając ułamek z nich do bieżącej aktualizacji. To sprawia, że algorytm jest mniej wrażliwy na lokalne minima i przyspiesza konwergencję. Innym pomysłem byłoby zastosowanie techniki decay (również Learning Rate Decay lub Weight Decay), gdzie głównym celem jest zredukowanie współczynnika uczenia w miarę zbliżania się do minimum funkcji kosztu, co pozwala na dokładniejsze dopasowanie do danych.

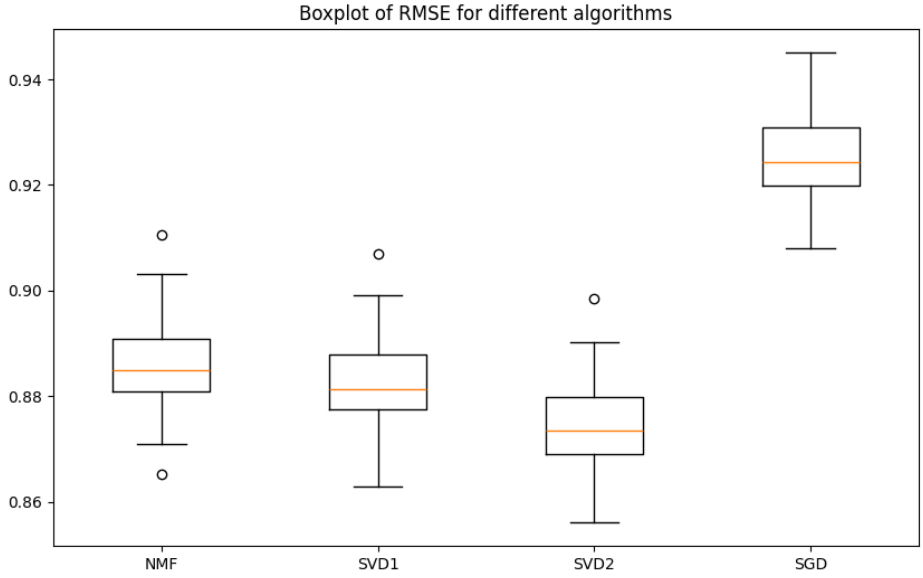
7 Wyniki i podsumowanie

Algorytm po przeszkoleniu na zbiorze **Z** tworzy **Z'**, macierz zawierającą elementy **Z'**$[u, m]$ dla $(u, m) \in \mathbf{T}$. Wtedy jakość działania algorytmu jest obliczana jako pierwiastek błędu średniokwadratowego:

$$RMSE = \sqrt{\frac{1}{|\mathbf{T}|} \sum_{(u,m)\in\mathbf{T}} (\mathbf{Z'}[u, m] - \mathbf{T}[u, m])^2}.$$

Wartości RMSE dla każdego z rozważanych algorytmów zostały obliczone dla 100 różnych podziałów na zbiory treningowy i testowy, a wyniki są przedstawione poniżej:

	NMF	**SVD1**	**SVD2**	**SGD**
min RMSE	0.865	0.862	0.855	0.906
max RMSE	0.911	0.906	0.897	0.944
mean RMSE	0.886	0.882	0.874	0.921
czas działania (s)	12.31	7.88	9.23	24.48



Rysunek 8: Tabela i wykres pudełkowy dla 100 powtórzeń

Analiza otrzymanych wyników wykazała, że spośród zaimplementowanych metod najlepsze rezultaty w przewidywaniu nieznanych ocen użytkowników zapewnia SVD2, co zostało potwierdzone najniższą średnią wartością RMSE na poziomie 0.874. Najniższa wartość błędu przy 100 powtórzeniach z wykorzystaniem SVD2 wyniosła 0.855. Ponadto, algorytm ten wykazuje optymalność również pod względem efektywności czasowej. Algorytm SGD, który osiągnął najwyższy średni RMSE na poziomie 0.921, wymagał znacznie więcej czasu na wykonanie niż pozostałe rozważone algorytmy, co może wynikać z natury tej metody, wymagającej wielu iteracji aby osiągnąć konwergencję.