
Model rekomendacji aparatów fotograficznych

Błażej Balczun
Błażej Stawarz
Mikołaj Kaźmierczak

1.02.2024

Spis treści

Symbole	2
1 Opis problemu	2
2 Rozwiązanie	2
2.1 Schemat blokowy	2
2.2 Algorytm	5
3 Proof of Concept	6
4 Alternatywne podejście do problemu	8
Podsumowanie	9
Załączniki	9

Symbole

Każde wystąpienie symbolu x oznacza brak wartości (*null*), a w kontekście tego dokumentu dokładniej "brak opinii".

1 Opis problemu

Rozpatrywanym przez nas problemem jest zaproponowanie modelu rekomendacji w sklepie zajmującym się sprzedażą aparatów fotograficznych. W tym przypadku rozważamy system, gdzie rekomendacje bazowane są na recenzjach wystawianych przez użytkowników, które następnie poddawane są analizie tekstowej z użyciem technologii NLP [ang. *Natural Language Processing*] aby poznać stosunek użytkowników do poszczególnych "komponentów" aparatów (takich jak jakość zdjęć, dodatkowe funkcje, itd.). Wartości dla komponentów wyznaczane są w zależności od emocji przejawianych przez wystawiającego opinię, na podstawie czego agregowane są oceny "całościowe" poszczególnych aparatów, a na ich bazie późniejsze rekomendacje.

Możliwe oceny komponentów zostały zawarte w zbiorze: $\{x, -1, 0, 1\}$, gdzie opisują one odpowiednio: brak opinii, opinię negatywną, neutralną i pozytywną. Oceny całościowe są agregowane z użyciem wag ocen poszczególnych komponentów wszystkich aparatów dla konkretnego użytkownika, ustalonych na podstawie tego jak często dany użytkownik wspominał o danym komponencie w swoich recenzjach.

Wagi komponentów mogą także być poddane dodatkowej funkcji promującej, która zwiększa je lub zmniejsza w zależności od ich znaczenia dla użytkownika. Dzięki temu możemy zaproponować takiej osobie aparat o cechach, na których prawdopodobnie najbardziej zależy danemu użytkownikowi.

2 Rozwiązanie

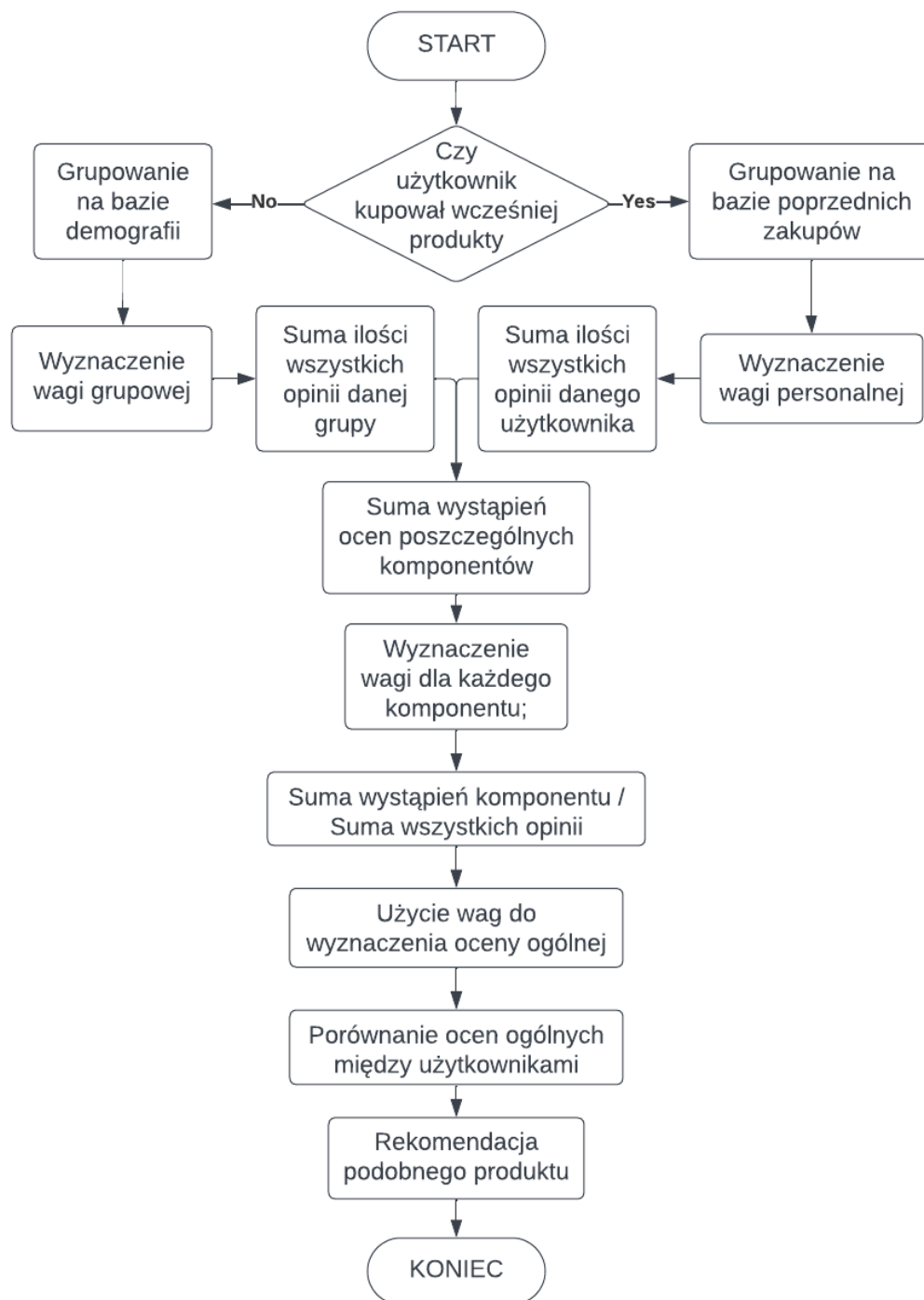
2.1 Schemat blokowy

Na rysunku 1 przedstawiony został schemat opisujący działanie systemu rekomendacji. Rysunek 2 opisuje proponowany proces działania w przypadku dodawania nowego użytkownika, tzw. cold start.

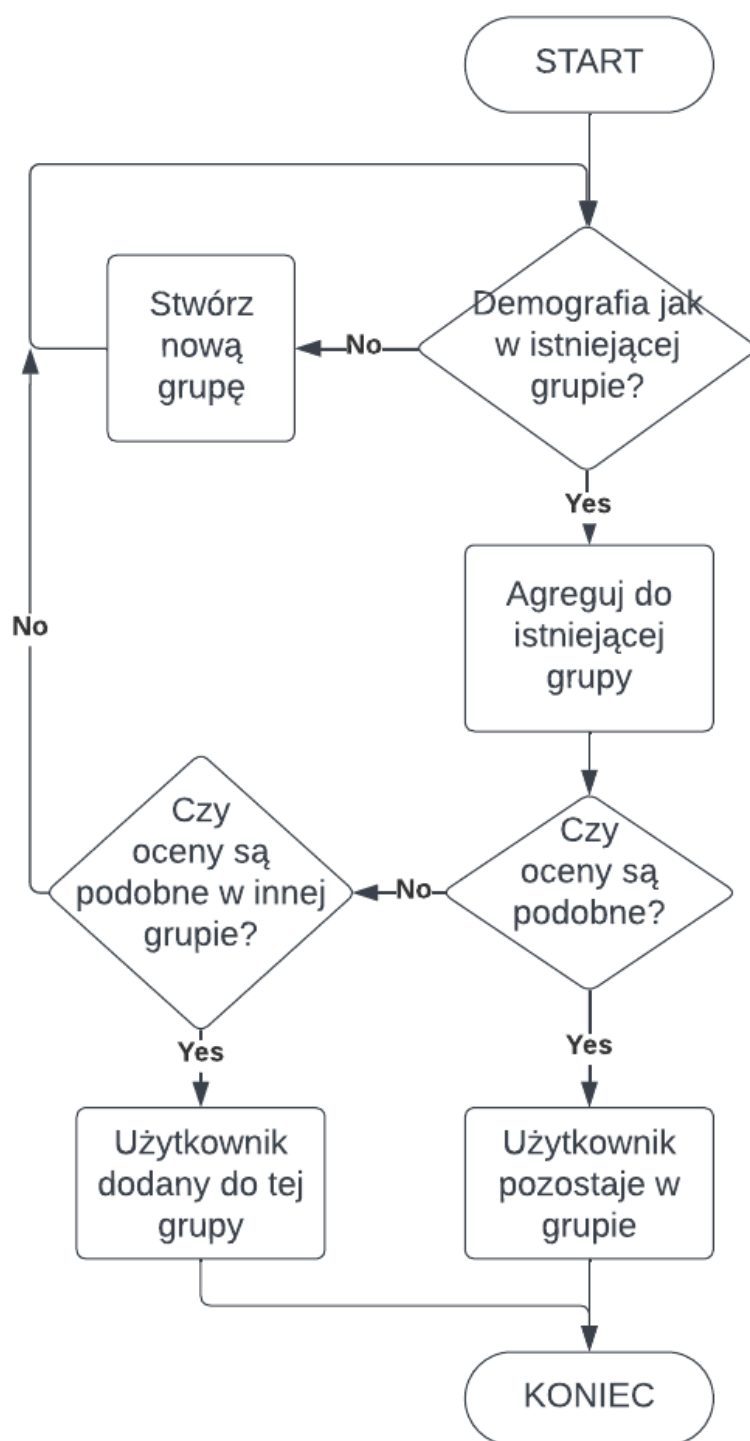
Przy tworzeniu rekomendacji produktu pierwszym krokiem jest sprawdzenie, czy dany użytkownik wystawił wcześniej opinię na temat jednego z produktów. Na bazie tej informacji, system ten tworzy rekomendacje na podstawie jednej z dwóch możliwości:

- poprzednich ocen tego użytkownika — **wystawiał** już opinie,
- grupy demograficznej tego użytkownika — **nie wystawił** jeszcze żadnej opinii.

W obu przypadkach wyznaczane są wagi znaczenia poszczególnych komponentów aparatu dla danego użytkownika, dzięki czemu można zagregować ocenę całościową dla danego produktu.



Rysunek 1: Schemat blokowy systemu rekomendacji.



Rysunek 2: Schemat blokowy dla cold start.

2.2 Algorytm

Niech:

- n będzie ilością użytkowników,
- m ilością aparatów,
- v ilością komponentów w każdym aparacie (ilość komponentów jest taka sama dla wszystkich aparatów, ale przyjmować wartość symbolu x , jeśli aparat go nie posiada — dobrym przykładem może być wymienna bateria),
- $U = \{u_1, \dots, u_i, \dots, u_n\}$ będzie zbiorem użytkowników,
- $A = \{a_1, \dots, a_j, \dots, a_m\}$ będzie zbiorem aparatów,
- $K = \{\vec{k}_{11}, \vec{k}_{12}, \dots, \vec{k}_{ij}, \dots, \vec{k}_{nm}\}$ będzie zbiorem wektorów ocen komponentów określonego aparatu a_j wyznaczonych przez określonego użytkownika u_i ,
- $\vec{k}_{ij} = [k_{ij1}, \dots, k_{ijp}, \dots, k_{ijv}]$, gdzie $k_{ijp} \in \{x, -1, 0, 1\}$, będzie wektorem wartości ocen komponentów określonego aparatu a_j wyznaczonych przez określonego użytkownika u_i .

Dla tak oznaczonych zmiennych można przedstawić tabelę 1.

	a_1	\dots	a_m
u_1	\vec{k}_{11}	\dots	\vec{k}_{1m}
\vdots	\vdots	\ddots	\vdots
u_n	\vec{k}_{n1}	\dots	\vec{k}_{nm}

Tabela 1: Tabela wektorów ocen komponentów (K) dla aparatów (A) przez użytkowników (U).

Warto zaznaczyć, że wektor \vec{k}_{ij} dla aparatu dla którego dany użytkownik nie wystawił opinii będzie wypełniony wartościami x (null). Oznacza to, że ilość komponentów dla każdego aparatu jest stała, nawet jeśli dany komponent rzeczowo nie istnieje w danym aparacie. W związku z tym, każdy komponent każdego aparatu musi mieć przypisaną ocenę, ale domyślnie przyjmuje ona wartość x , co jednak nie ma wpływu na wynik rekomendacji, ze względu na sposób agregacji ocen całościowych.

Przyjmijmy teraz arbitralnego użytkownika u_i , który ocenił już wcześniej dowolną ilość aparatów. Aby wyznaczyć wagi komponentów dla tego użytkownika, wpieryw należy sprawdzić, jak często oceniany był dany komponent (tj. przyjął wartości ze zbioru $\{-1, 0, 1\}$) w stosunku do ilości wszystkich aparatów ($|A|$).

Niech zbiór Ω zawiera wektory wag komponentów dla każdego użytkownika u_i .

$$\Omega = \vec{\omega}_1, \dots, \vec{\omega}_i, \dots, \vec{\omega}_n \quad (1)$$

Wtedy:

$$\vec{\omega}_i = [\omega_{i1}, \dots, \omega_{ip}, \dots, \omega_{iv}] \quad (2)$$

Gdzie ω_{ip} oznacza wagę danego komponentu w ramach określonego użytkownika:

$$\omega_{ip} = \frac{|\{k_{i1p}, \dots, k_{ijp}, \dots, k_{imp}\}|}{m} \quad (3)$$

Następnie wartości te są poddawane normalizacji (której opis został tu pominięty ze względu na jej trywialność) oraz funkcji promującej komponenty o wyższych wagach. Przykładowa funkcja promująca może przyjąć nawet tak prostą formę jak:

$$P(\omega_{ip}) = \omega_{ip}^2 \quad (4)$$

Na skutek działania funkcji promującej różnice pomiędzy wagami stają się większe. Tak wyznaczone wagi komponentów są używane w celu wyznaczenia oceny całościowej aparatu. Oznaczamy zbiór zagregowanych ocen całościowych jako C , wtedy:

$$C = \{c_{11}, c_{12}, \dots, c_{ij}, \dots, c_{nm}\}, \quad \text{gdzie } c_{ij} \in \mathbb{R} \quad (5)$$

Czyli ocena c_{ij} , wynika ze wzoru:

$$c_{ij} = k_{ij1}\omega_{i1} + \dots + k_{ijp}\omega_{ip} + \dots + k_{ijv}\omega_{iv} \quad (6)$$

Dzięki temu uzyskujemy ocenę całościową dla danego aparatu według określonego użytkownika. Teraz, znając całościowe oceny aparatów, można przejść do porównywania użytkowników za pomocą podobieństwa Cosinusowego. Należy wpierw wyznaczyć wektory ocen całościowych wszystkich produktów dla dwóch użytkowników, oznaczone będą jako a i b . Zakładając, że dla a : $i = 1$, a dla b : $i = 2$:

$$\vec{a} = [c_{11}, \dots, c_{1,j}, \dots, c_{1,m}] \quad (7)$$

$$\vec{b} = [c_{21}, \dots, c_{2,j}, \dots, c_{2,m}] \quad (8)$$

$$d(a,b) = \frac{a \cdot b}{\|a\| \|b\|} \quad (9)$$

Należy powtórzyć ten krok dla wszystkich i względem wybranego użytkownika, a z tak zebranego wektora wyznaczyć, z którym użytkownikiem podobieństwo było największe, na podstawie tego, która zwrotna wartość funkcji podobieństwa Cosinusowego była najwyższa. Następnie z aparatów tamtego użytkownika dobrać aparat na podstawie oceny, która była u niego najwyższa.

W skutek tych działań otrzymamy rekomendację aparatu dla danego użytkownika.

3 Proof of Concept

Prototypowa implementacja modelu jest oparta na zaproponowanym projekcie algorytmu. Kod implementacji można sprawdzić pod linkiem:

github.com/mikolajkazmierczak/camera-recommendation-model.

Przykładowa, prototypowa implementacja przygotowana została na ręcznie spreparowanych przykładowych danych (tabela 2), opartych na 7 użytkownikach, 4 aparatach i 10 komponentach dla każdego z nich. Ostatecznie podana jest rekomendacja aparatu dla pierwszego użytkownika (u_0), a także, na jakim innym użytkowniku została zaproponowana.

k	a_0	a_1
u_0	$[-1, 0, 1, 1, x, x, x, 1, -1, 1]$	$[1, 1, x, x, 1, 1, x, -1, x, 1]$
u_1	$[x, 1, 1, x, 1, x, x, x, -1, 1]$	$[x, -1, 1, 1, 1, x, x, x, 1, 1]$
u_2	$[-1, 0, x, 1, x, x, x, 1, x, 1]$	$[0, 1, x, 1, 0, 1, x, -1, x, x]$
u_3	$[x, 0, 1, 1, 1, x, x, 1, -1, 1]$	$[-1, -1, x, x, 1, 1, x, -1, x, -1]$
u_4	$[x, 0, -1, x, 1, x, 1, 1, x, 1]$	$[x, -1, x, -1, x, 1, x, -1, x, -1]$
u_5	$[-1, x, x, 1, x, x, x, 1, 0, 1]$	$[1, -1, 1, x, x, 0, x, -1, 0, 0]$
u_6	$[x, x, 1, 1, x, x, x, 1, x, 1]$	$[x, 0, 1, x, x, -1, x, -1, 0, 0]$

k	a_2	a_3
u_0	$[0, 0, 1, 1, 1, -1, -1, x, x, 0]$	$[x, x, -1, -1, 0, -1, x, x, x, -1]$
u_1	$[x, -1, -1, x, x, x, x, x, 0, -1]$	$[1, -1, x, x, x, x, x, x, 1, 1]$
u_2	$[1, -1, 0, x, 1, -1, -1, x, 0, -1]$	$[1, 1, x, -1, 0, x, x, 1, x, -1]$
u_3	$[0, 0, x, 1, x, -1, -1, x, x, -1]$	$[1, x, -1, x, 0, -1, x, x, 0, -1]$
u_4	$[1, x, x, 1, x, x, -1, x, 0, -1]$	$[0, x, -1, x, -1, -1, -1, x, x, -1]$
u_5	$[1, x, x, 1, x, 1, 1, x, -1, 1]$	$[1, x, x, x, 0, -1, x, -1, 0, -1]$
u_6	$[0, x, 0, 1, x, 1, 1, x, -1, 1]$	$[1, 1, x, x, x, -1, x, -1, 0, x]$

Tabela 2: Tabela wektorów ocen komponentów (k) dla aparatów (a) przez użytkowników (u)

	a_0	a_1	a_2	a_3
u_0	0.32	0.42	0.26	0
u_1	0.33	0.43	0	0.24
u_2	0.34	0.37	0	0.29
u_3	1.00	0	0	0
u_4	0.52	0.16	0.32	0
u_5	0.31	0.17	0.51	0
u_6	0.40	0	0.49	0.11

Tabela 3: Zagregowane oceny komponentów jeśli ich wagi **nie były** poddane funkcji promującej

	a_0	a_1	a_2	a_3
u_0	0.30	0.50	0.20	0
u_1	0.30	0.53	0	0.16
u_2	0.35	0.40	0	0.25
u_3	1.00	0	0	0
u_4	0.67	0.07	0.26	0
u_5	0.25	0.07	0.67	0
u_6	0.39	0	0.58	0.03

Tabela 4: Zagregowane oceny komponentów jeśli ich wagi **były** poddane funkcji promującej

Tabela 3 pokazuje wynikowe oceny produktów bez poddawania wag komponentów funkcji promującej. Tabela 4 pokazuje wynikowe oceny produktów, jeśli wagi komponentów zostały poddane funkcji promującej. Różnice pomiędzy ocenami poszczególnych aparatów rosną po zastosowaniu funkcji promującej, dzięki czemu system rekomendacji jest bardziej dopasowany do najważniejszych potrzeb określonych użytkowników.

Jeśli wagi nie są poddane funkcji promującej rekomendacja systemu dla użytkownika u_0 to aparat a_0 na bazie użytkownika u_4 . Jeśli natomiast zastosuje się funkcję promującą rekomendacja zmienia się na aparat a_1 na bazie użytkownika u_1 . Pokazuje to jak duży wpływ ma funkcja promująca na rekomendację modelu.

4 Alternatywne podejście do problemu

Problem rekomendacji został podzielony na dwie części:

- element klasteryzacji użytkowników na podstawie cech produktów, które zostały przez nich wymienione w opiniach.
- rekomendowanie danych produktów bazując na przynależności użytkownika do danej grupy

Użytkownik u_n uwzględnił w swoich opiniach cechy produktów f_n . Analiza tekstu opinii przypisała do każdej z wymienionych cech wartość od 0.0 do 1.0, które reprezentuje jej wagę w stosunku do całej opinii. W przypadku kiedy dana cecha nie została wymieniona jej wartość wynosi 0.0

	f_1	f_2	f_3	f_4	f_m
u_1	0	0.5	0.7	0	
u_2	0	0.7	0.7	1	
u_3	0.6	0.6	1	0	
u_n					

Tabela 5: Uwzględnienie cechy aparatów w opiniach użytkowników.

Wyznaczamy macierz podobieństwa użytkowników wykorzystując do tego odległość Cosinusową wyrażoną jako:

$$d(u_n, u_m) = \frac{u_n \cdot u_m}{\|u_n\| \|u_m\|} \quad (10)$$

Wszystkie w tabeli 5 zawsze są dodatnie i nie ma potrzeby ich normalizowania.

$d(u_n, u_m)$	u_1	u_2	u_m
u_1	0	0.5	
u_2	0	0.7	
u_n			

Tabela 6: Macierz podobieństwa użytkowników.

Na macierzy podobieństwa (tabela 6) szukamy grup użytkowników wykorzystując algorytm k-średnich, który w swoich założeniach minimalizuje wartość dewiacji, która jest odległością eukleidesową do . Zakładamy pewną k liczbę grup, których szukamy.

$$D_k = \sum_{d \in S_k} |d - \mu_k|^2 \quad (11)$$

gdzie:

$$\mu_k = \frac{1}{|S_k|} \sum_{d \in S_k} d \quad (12)$$

Dla każdej grupy S_k wyznaczamy wagi cech produktu. Każda z pojedynczych wag jest średnią arytmetyczną znaczenia danej cechy w wybranej grupie.

$$W_{k,i} = \frac{1}{n} \cdot \sum_{f_n \in S_k} f_n \quad (13)$$

Opinie użytkowników przedstawione są w postaci trój wymiarowej macierzy (tabela 7), w której dla każdego produktu oraz użytkownika przypisujemy zbiór ocen cząstkowych poszczególnych cech produktu, które należą do zbioru $\langle -1, 1 \rangle \cap \{0\} \cup \{x\}$. Zbiory ocen są tej wielkości zależnej od ilości cech (tabela 5) W tym przypadku zakładamy, że zarówno dla opinii neutralnej, jak i braku opinii przypisujemy wartość x . Neutralna opinia pewnej cechy produktu nie wnosi wartości co do jej sumarycznej oceny, a jedynie wskazuje, że użytkownik przywiązuje do danej cechy uwagę.

O_{u_n, c_m}	c_1	c_2	c_m
u_1	$[-1, 0.3, 0, 0.7]$	$[-1, 0, 0.3, 0]$	
u_2	$[x, x, x, x]$	$[-0.5, 0, 0.3, 0]$	
u_n			

Tabela 7: Opinie produktów wyrażone przez użytkowników

W momencie, kiedy użytkownik został już przypisany do pewnej grupy użytkowników S_k , jesteśmy w stanie obliczyć sumaryczną ocenę każdego z produktów na podstawie wag W_k przypisanych do wybranej grupy.

$$R_n = \frac{1}{i} \cdot \sum_{O_n} (O_n \cdot W_k) \quad (14)$$

Podsumowanie

Realizacja tego projektu pozwoliła nam lepiej zrozumieć zagadnienie profilowania użytkowników w systemach rekomendacji, jak i mnogość problemów podczas projektowania takiego systemu. Jako że opracowany system operował na ocenach komponentów wyznaczonych z recenzji, musieliśmy zaadaptować go w celu tworzenia agregowania ocen składowych do jednej "całościowej" wartości liczbowej. Dzięki temu jednak model rekomenduje produkt o cechach, które najbardziej ceni zainteresowany użytkownik.

System można dodatkowo uzupełnić o wiele funkcji, które wynikają z problemów z obsługą wyjątków. Jednym z nich, jest system klasteryzacji, który przypisuje użytkownika do grup na bazie jego poprzednich ocen, pozwalając na łatwiejszą rekomendację produktów, które mogłyby spełniać jego wymagania. W podobny sposób można obsłużyć problem cold start. Jest to problem, w którym musimy obsłużyć użytkownika, który dopiero zarejestrował się w serwisie zakupowym, a więc nie wystawił jeszcze żadnych recenzji. W takim wypadku osobę taką można przypisać do grupy na bazie demografii. Może to być przykładowo wiek, płeć czy miejsce zamieszkania. Na podstawie przypisania do takiej grupy, możemy zaproponować użytkownikowi produkty, które kupowały osoby o przybliżonych cechach demograficznych.

Załączniki

github.com/mikolajkazmierczak/camera-recommendation-model